

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра комп'ютерної інженерії



**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему Розробка комп'ютерної гри в жанрі «аркада»
(назва кваліфікаційної роботи згідно наказу ОНТУ)

Здобувача Діденчука В. О.
(прізвище, ініціали)
4 курсу КІ-543 групи

Керівники: ст. викл. Сіренко О. І.
(посада, прізвище та ініціали)
проф. Артеменко С. В.
(посада, прізвище та ініціали)

Консультанти: проф. Басюркіна Н. Й.
(посада, прізвище та ініціали)
доцент Нєнов О. Л.
(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 10.06 2023 р., протокол № 8
Завідувач кафедри комп. інженерії _____ Сергій АРТЕМЕНКО
(назва кафедри) (підпис) (Ім'я ПРІЗВИЩЕ)

Одеса - 2023 рік

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет	<u>комп'ютерної інженерії, програмування та кіберзахисту</u>
Кафедра	<u>комп'ютерної інженерії</u>
Ступінь вищої освіти	<u>бакалавр</u>
Спеціальність	<u>123 «Комп'ютерна інженерія»</u>
Освітня програма	<u>Розробка ігор та інтерактивних медіа у віртуальній реальності</u>

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної інженерії

Сергій АРТЕМЕНКО

« 1 » вересня 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Діденчука Владислава Олександровича

1. Тема роботи Розробка комп'ютерної гри в жанрі «аркада»

Затверджена наказом університету від « 10 » серпня 2022 р., наказ № 440-03

2 Термін здачі здобувачем закінченої роботи 5 червня 2023 р.

3. Вихідні дані роботи

1. Жанр комп'ютерної гри.

2. Документація на середовище розробки Unity.

4. Перелік питань, які потрібно розробити

1. Вступ. 2. Передпроектний аналіз. 3. Постановка завдання. Концепт-документ.

4. Проектування гри. Дизайн-документ. 5. Створення контенту і програмна реалізація.

6. Економічні розрахунки. 7. Охорона праці. 8. Загальні висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Слайд 1. Характеристика кваліфікаційної роботи. Слайд 2. Огляд подібних ігор на ринку. Слайд 3. Вимоги до програмного проекту. Слайд 4. Архітектура проекту.

Слайд 6. Програмні алгоритми. Слайд 7. Елементи інтерфейсу користувача.

Слайд 8. Техніко-економічні показники. Слайд 9. Загальні висновки.

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Економіка</i>	<i>Басюркіна Н. Й., д. е. н., проф.</i>		
<i>Охорона праці</i>	<i>Нєнов О. Л., к. т. н., доц.</i>		
<i>Нормоконтроль</i>	<i>Сіренко О. І., ст. викл.</i>		

7. Дата видачі завдання 31.08.2022

Керівники _____ *Олександр СІРЕНКО*

_____ *Сергій АРТЕМЕНКО*

Завдання прийняв до виконання _____ *Владислав ДІДЕНЧУК*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Передпроектний аналіз предметної області.</i>	<i>01.10.2022</i>	
2.	<i>Постановка завдання на розробку.</i>	<i>01.12.2022</i>	
3.	<i>Проектування архітектури проекту гри.</i>	<i>01.02.2023</i>	
4.	<i>Проектування компонентів проекту гри.</i>	<i>31.03.2023</i>	
5.	<i>Проектування алгоритмів гри.</i>	<i>07.04.2023</i>	
6.	<i>Програмна реалізація гри.</i>	<i>14.04.2023</i>	
7.	<i>Тестування гри.</i>	<i>28.04.2023</i>	
8.	<i>Техніко-економічний аналіз проекту.</i>	<i>05.05.2023</i>	
9.	<i>Опрацювання питань охорони праці.</i>	<i>12.05.2023</i>	
10.	<i>Оформлення пояснювальної записки і презентації.</i>	<i>31.05.2023</i>	

Керівники роботи _____ *Олександр СІРЕНКО*

_____ *Сергій АРТЕМЕНКО*

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач - дипломник _____ *Владислав ДІДЕНЧУК*

АНОТАЦІЯ

Дана робота присвячена розробці комп'ютерної гри – ремейку класичної аркадної гри «Asteroids» для платформи Windows. В роботі виконаний передпроектний аналіз, в тому числі – аналоги, розроблено технічне завдання, виконано проектування застосунку. Здійснено обґрунтований вибір інструментальних засобів розробки. Виконана програмна реалізація його першого варіанту. Досліджено питання економічного обґрунтування проекту, охорони праці та пожежної безпеки.

Обсяг пояснювальної записки без додатків – 93 листа, додатків – 11 листів.

Ключові слова: *комп'ютерна гра, аркада, ігрова фізика, Unity, C#.*

ABSTRACT

This work is devoted to the development of a computer game - a remake of the classic arcade game "Asteroids" for the Windows platform. In the work, a pre-project analysis was performed, including analogs, a technical task was developed, and the application was designed. A well-founded selection of development tools was made. The software implementation of its first version has been completed. The issues of economic justification of the project, labor protection and fire safety were studied.

The volume of the explanatory note without appendices is 93 sheets, appendices – 11 sheets.

Keywords: computer game, arcade, game physics, Unity, C#.

ЗМІСТ

	стор.
ВСТУП.....	6
РОЗДІЛ 1 ПЕРЕДПРОЕКТНИЙ АНАЛІЗ І ПІДГОТУВАННЯ	
РОБОЧОГО МІСЦЯ	8
1.1 Загальна характеристика жанру комп'ютерних ігор «аркада»	8
1.2 Гра Asteroids як представник аркадних ігор	12
1.3 Огляд подібних ігор на ринку.....	16
1.3.1 Asteroids від Free Video Games Online.....	16
1.3.2 Asteroids від AARP	17
1.3.3 Asteroids від Novel Games	17
1.3.4 Asteroids: Space Defense	18
1.4 Узагальнена постановка завдання	19
1.5 Підготування робочого місця	20
Висновки першого розділу.....	26
РОЗДІЛ 2 ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ АРКАДНОЇ ГРИ	
2.1 Технічне завдання на розробку комп'ютерної логічної гри	27
2.1.1 Загальна інформація про проект.....	27
2.1.2 Функціонал гри.....	27
2.1.3 Нефункціональні вимоги до гри.....	28
2.1.4 Організаційні оцінки	28
2.2 Вибір інструментів розробки	29
2.3 Структура проекту Unity.....	30
2.4 Керування кораблем в грі	35
Висновки другого розділу	37
РОЗДІЛ 3 РОЗРОБКА ЗАСТОСУНКУ В UNITY	
3.1 Створення проекту і налаштування сцени	38
3.2 Створення ігрового персонажа – корабля.....	40

					<i>КРБ.КІ.1.440-03.3.12</i>		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>		<i>Владислав ДІДЕНЧУК</i>			<i>Літ.</i>	<i>Арк.</i>	<i>Архувів</i>
<i>Перевір.</i>		<i>Олександр СІРЕНКО</i>			4	104	
<i>Рецензент</i>		<i>Євгеній ДАНЬКО</i>			<i>гр. КІ-543, ОНТУ</i>		
<i>Н. контр.</i>		<i>Олександр СІРЕНКО</i>					
<i>Затверд.</i>		<i>Сергій АРТЕМЕНКО</i>					

Розробка комп'ютерної гри в жанрі «аркада»

3.3	Створення ігрових куль.....	45
3.4	Створення астероїдів.....	50
3.5	Обробка зіткнення астероїдів з кулями.....	55
3.6	Обробка зіткнення корабля з астероїдами	56
3.7	Реалізація ефекту вибуху при руйнуванні астероїда і корабля	59
3.8	Підрахунок очок гри	62
3.9	Завершення гри.....	63
	Висновки третього розділу.....	64
РОЗДІЛ 4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЕКТУ		65
4.1	Організаційно-економічна характеристика роботи.....	65
4.2	Розрахунок науково-технічної ефективності проекту.....	68
4.3	Проведення оцінки науково-технічного рівня розробки.....	71
4.4	Розрахунок економічної ефективності проекту.....	73
4.5	Розрахунок ціни продукту	74
4.6	Витрати на матеріали	75
4.7	Розрахунок заробітної плати.....	75
4.8	Калькуляція собівартості продукту	77
4.9	Розрахунок капітальних витрат	78
4.10	Розрахунок поточних експлуатаційних витрат.....	79
4.11	Розрахунок показників економічної ефективності проекту.....	81
	Висновки четвертого розділу	83
РОЗДІЛ 5 ОХОРОНА ПРАЦІ НА РОБОЧОМУ МІСЦІ		84
5.1	Мікроклімат виробничого середовища	84
5.2	Пожежна профілактика	86
	Висновки п'ятого розділу.....	89
ЗАГАЛЬНІ ВИСНОВКИ.....		90
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		91
ДОДАТКИ		93
	Додаток А Вибрані слайди презентації	93
	Додаток Б Сирцевий код застосунку	98

ВСТУП

Сьогодні індустрія комп'ютерних ігор є дуже важливою частиною ІТ-галузі. Вона швидко зростає, привертаючи мільйони гравців з усього світу та забезпечуючи зацікавленим особам мільярди доларів доходів щороку. Багато компаній по всьому світу створюють ігри для різних платформ: комп'ютерів, консолей, мобільних пристроїв тощо. Крім того, індустрія постійно розвивається та вдосконалюється, використовуючи нові технології, покращену графіку та інші функції, що забезпечуються новими апаратними і програмними засобами.

Крім того, комп'ютерні ігри мають значний вплив на суспільство та культуру, що підкреслює їхню важливість у сучасному світі. Є ігри, які допомагають людям розвивати певні навички та здібності, інші пропонують віртуальні світи для експериментів та відпочинку, а деякі використовуються в бізнесі і навчальному процесі. Таким чином, індустрія комп'ютерних ігор не просто залишається актуальною: вона розширює свої межі та вплив на різні аспекти нашого життя. Це показує важливість і актуальність цієї дипломної роботи.

Окремий тип комп'ютерних ігор – так звані аркадні ігри – були дуже популярні у 80-х та 90-х роках ХХ століття. Ці ігри зазвичай були прості у використанні, мали яскраву графіку та веселу музику. Вони часто включали елементи азартних ігор та були орієнтовані на швидку взаємодію між гравцем та грою. У наші дні, незважаючи на появу нових технологій та нових жанрів комп'ютерних ігор, аркадні ігри залишаються актуальними. Вони можуть запропонувати гравцям простий та швидкий спосіб отримання задоволення, часто у поєднанні з ностальгічними спогадами про минуле. Крім того, сучасні аркадні ігри іноді включають нові технології, такі як віртуальна реальність або сучасні засоби інтерактивної взаємодії, що дозволяє їм залишатися актуальними і привабливими для нового покоління гравців.

					КРБ.КІ.1.440-03.3.12	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктом дослідження в даній роботі є узагальнений процес розробки комп'ютерних аркадних ігор.

Предметом дослідження є складові розробки проекту комп'ютерної аркадної гри.

Основною прикладною метою даної роботи є розробка проекту і реалізація початкової версії комп'ютерної аркадної гри.

Для здійснення поставленої мети вирішується низка завдань, серед яких:

- узагальнена постановка завдання;
- аналітичний огляд ринку комп'ютерних аркадних ігор;
- розробка концепт-документу аркадної гри;
- створення проекту комп'ютерної аркадної гри у відповідності до розробленого концепту;
- реалізація проекту у тестовий прототип застосунку;
- техніко-економічне обґрунтування розробки;
- дослідження суміжних питань охорони праці;
- оформлення пояснювальної записки та підготування презентації проекту.

					<i>КРБ.КІ.1.440-03.3.12</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		7

РОЗДІЛ 1

ПЕРЕДПРОЕКТНИЙ АНАЛІЗ І ПІДГОТУВАННЯ РОБОЧОГО МІСЦЯ

1.1 Загальна характеристика жанру комп'ютерних ігор «аркада»

Аркада (від англ. *Arcade* – пасаж, крита галерея) – жанр комп'ютерних ігор, що відрізняється коротким, але інтенсивним ігровим процесом. Історично аркадними іграми називалися ігри для так званих аркадних ігрових автоматів, які розташовувалися у критих галереях магазинів. «Золотий час» аркадних ігор припав на період з кінця 1970-х до середини 1980-х. Пізніше аркадами стали називати ігри, які були портовані з аркадних автоматів, а також нові ігри схожої жанрової тематики.

Computer Space – одна з перших аркадних ігор – була створена у 1971 році (рис. 1.1). Зараз вважається, що вона є першою промисловою електронною грою цього жанру, хоча її назва почала асоціюватися з аркадними іграми тільки в кінці 1970-х років.

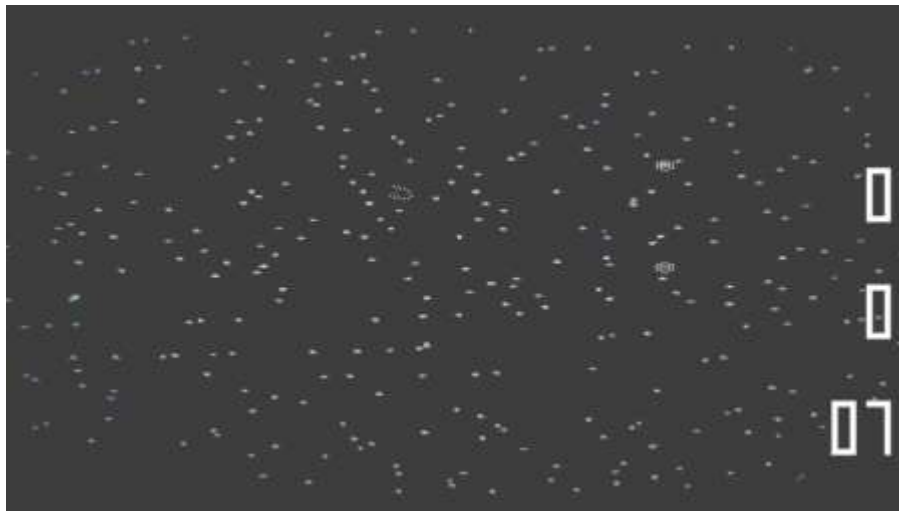


Рис. 1.1 – Гра Computer Space (1971)

Перші аркадні ігри, такі як *Asteroids*, *Battlezone* (рис. 1.2) і *Star Castle*, використовували векторну графіку для створення зображень з ліній.

					КРБ.КІ.1.440-03.3.12	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

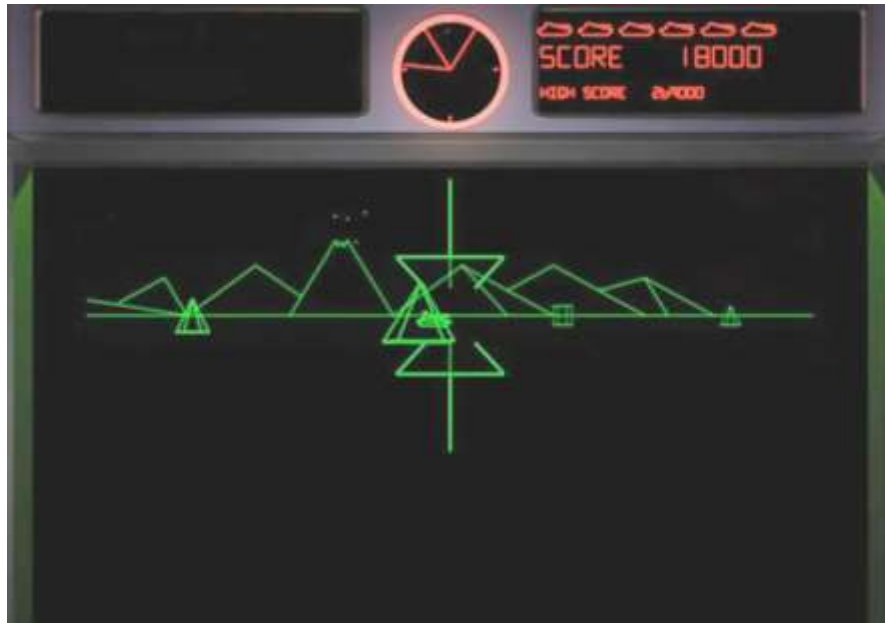


Рис. 1.2 – Гра Atari Battlezone (1980)

З появою кольорової растрової графіки з'явилися мультиплікаційні відеоігри та їх культові персонажі, такі як *Pac-Man* (рис. 1.3) і *Donkey Kong*, які стали характерними проявом тогочасної електронної культури.



Рис. 1.3 – Гра Pac Man від Namco (1980)

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

У середині 1980-х років аркадні ігри стали більш поширеними та різноманітними стосовно інтерфейсів управління, тематики, графіки та жанрів. Були створені більш реалістичні контролери, такі як двомісні космічні кораблі *Tail Gunner* і управління автомобілем *Ferrari* в грі *Out Run*. З часом домашні ігрові системи почали витісняти аркадні автомати, оскільки їх графіка почала перевершувати більшість аркадних ігор. До кінця 1990-х років аркадні автомати були майже повністю виведені з індустрії.

У 2000-х році, коли ігрові центри стали надавати послуги погодинної оренди комп'ютерних та консольних ігор, аркадний жанр почав використовуватися в соціальних та мережевих іграх. Типовими відмінними рисами цього використання стали обстановка та переваги інтернет-кафе або відеозалу. Деякі компанії розробляли ігри, намагаючись поєднати аркадний досвід гравців та переваги атракціонного підходу. Наприклад, *Disney* та *Sally Corporation* у 2008 році у грі *Toy Story Midway Mania!* використовували гігантські відеоекрани для чотирьох гравців, карнавальний стиль та візуальні ефекти, створені за допомогою спеціального спрею.

Нижче наведені характерні властивості класичних аркад.

1. Одноекранний геймплей – у класичних аркадних іграх усі дії відбувалися на одному екрані. Історично це було обумовлено, головним чином, технічними обмеженнями, але в той же час, здійснило значний вплив на гейм дизайн аркад взагалі. При одно екранному геймплеї гравці можуть в будь-який час бачити весь ігровий світ і приймати рішення, ґрунтуючись на повній інформації про свій поточний стан. Деякі ігри цього жанру включали в себе більше одного екрану, які виконували функції ігрових рівнів. Прикладами таких ігор є *Joust*, *Rac-Man*, *Mario Bros*.
2. Нескінченна гра – ще одна з особливостей аркадних ігор. Гравець міг безмежно довго грати аркадну гри, при цьому не маючи можливості завершити гру виграшем. Це заставляло гравців кидати виклик самим собі: наскільки довго вони можуть протриматися. Ігри проектувалися

										Арк.
										10
Змн.	Арк.	№ докум.	Підпис	Дата						

КРБ.КІ.1.440-03.3.12

таким чином, щоб з часом гравцеві ставало все складніше грати: нескінченна гра пропонувала, таким чином, нескінченну складність. З появою ринку домашніх комп'ютерів дана ситуація змінилася, так як видавці були зацікавлені в тому, щоб гравці проходили гру і після цього хотіли купити нову.

3. Обмежений набір спроб. Класичні аркади часто пропонують гравцеві кілька «життів», щоб він міг вивчити ігрові механізми і продовжувати грати, навіть якщо завершить гру внаслідок помилки. Цей підхід був привабливим для новачків, які могли таким чином навчитися грати і повертатися до гри знову і знову. Крім того, гравці, які продемонстрували успіх у грі, могли отримати додаткові «життя» в якості нагороди за свої досягнення.
4. Ігровий рахунок і таблиця рекордів. У класичних аркадах зазвичай є ігровий рахунок, який гравець заробляє за виконання різних завдань та цілей у грі. Це допомагає гравцеві зрозуміти, наскільки добре він грав, навіть якщо не зміг (або не міг) виграти. Також є таблиця рекордів, де гравець може ввести свої ініціали поруч із результатом, порівняти свій результат з іншими гравцями та змагатися з ними. У типовій аркадній грі тривалість ігрового сеансу середньостатистичного гравця становить близько двох хвилин, а досвідченого – може досягати десятків хвилин.
5. Простий геймплей. Класичні аркади відрізняються швидким навчанням та нескладним ігровим процесом, який тим не менш не дозволяє стати майстром у грі через її високу складність. Якщо гравець гине, це зазвичай відбувається з його вини, оскільки в аркадах немає складних комбінацій клавіш, які потрібно вивчити для виконання особливих дій. У таких іграх не застосовуються очки здоров'я, щити або пігулки сили, оскільки розробникам потрібно було залучити якнайбільше гравців, включаючи новачків. Простота ігрового процесу не означає, що

					<i>КРБ.КІ.1.440-03.3.12</i>	<i>Арк.</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		11

він поганий або обмежений – навпаки, він може бути елегантним та дуже детально пропрацьованим.

6. Класичні аркади не мають сюжету або історії. Ця тенденція зберігається і у сучасних аркадах. Мета ігор цього жанру – зрозумілість та швидке входження в гру, чи то є наукова фантастика, війна, спорт або щось ще. Деякі аркади створюють свої унікальні світи, такі як *Rac-Man* та *Q*Bert*. Однак геймдизайнери більшості класичних аркад не вважали за необхідне заповнювати свої світи сюжетом або пояснювати гравцям, чому їм потрібно стріляти в ті чи інші цілі.

Існують класичні аркади, які природним чином порушують деякі з правил, описаних раніше, але все ж таки розглядаються як представники жанру. Наприклад, ігри *Sinistar* та *Defender* мали скролінгові світи, проте гравці могли бачити всі аспекти гри. Іншим прикладом була гра *Battlezone*, яка принципово відрізнялася від інших ігор свого часу. На відміну від цих трьох ігор, такі ігри як *Space Invaders* та *Asteroids* повністю відповідають класичним правилам побудови аркадних ігор.

Штучний інтелект в аркадних іграх зазвичай представлений простими патернами та не реагує на дії гравця. Хоча штучний інтелект простий, його налаштування є нетривіальним завданням, оскільки розробнику необхідно збалансувати два фактори: залучення гравця та монетизацію. Для опису неігрових персонажів із простим штучним інтелектом використовується термін «аркадні вороги».

1.2 Гра Asteroids як представник аркадних ігор

Одним із перших великих хітів золотого віку аркадних ігор була гра *Asteroids*. Це багатоспрямована аркадна відеогра-шутер на космічну тематику, розроблена Лайлом Рейнсом та Едом Логгом і випущена в листопаді 1979 компанією Atari, Inc. Гравець управляє одним космічним

									Арк.
									12
Змн.	Арк.	№ докум.	Підпис	Дата					

КРБ.КІ.1.440-03.3.12

кораблем у полі астероїдів, яке періодично перетинають літаючі тарілки. Мета гри полягає в тому, щоб стріляти і знищувати астероїди та тарілки, не стикаючись з жодним з них і не потрапляючи під вогонь у відповідь тарілок. Гра ускладнюється поступовим збільшенням кількості астероїдів.

Asteroids була заснована на незакінченій грі Cosmos, а її фізична модель, схема управління та елементи ігрового процесу були частково запозичені з ігор Spacewar!, Computer Space та Space Invaders і вдосконалені методом проб та помилок. Гра візуалізується на векторному дисплеї у двовимірному вигляді, що охоплює обидві осі екрана (рис. 1.4).



Рис. 1.4 – Гра Asteroids від Atari (1980)

Гра виявилася дуже популярною серед гравців і вплинула на розробників. Було продано 47840 вертикальних шаф і 8725 коктейльних шаф з аркадними автоматами Asteroids. У 1980-х роках її було перенесено на домашні системи Atari, і версію Atari VCS було продано тиражем понад три

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

мільйони копій. Гру широко копіювали, і вона безпосередньо вплинула на Defender, Gravitar та багато інших відеоігор.

Ігровий процес гри містить трикутний космічний корабель, оточений астероїдами та літаючими тарілками. Мета гри – знищувати астероїди та тарілки. Гравець керує кораблем, який може обертатися ліворуч і праворуч, стріляти прямо вперед і рухатися вперед. Як тільки корабель починає рухатися в якомусь напрямку, він буде продовжувати рухатися в цьому напрямку деякий час без втручання гравця, якщо гравець не застосує тягу в іншому напрямку. Корабель зрештою зупиняється, коли його нічого не штовхає. Гравець також може відправити корабель у гіперпростір, внаслідок чого він зникне і знову з'явиться у випадковому місці на екрані, ризикуючи самознищитися внаслідок зіткнення з астероїдом.

Кожен рівень починається з кількох великих астероїдів, які дрейфують по екрану в різних напрямках. Об'єкти обтікають краї екрана – наприклад, астероїд, який зміщується за верхній край екрана, знову з'являється внизу і продовжує рухатися у тому ж напрямі. Коли гравець стріляє в астероїди, вони розбиваються на дрібніші астероїди, які рухаються швидше і в них важче влучити. Менші астероїди приносять більше очок.

На екрані періодично з'являються дві літаючі тарілки: велика тарілка стріляє у довільних напрямках, а маленька тарілка часто стріляє по кораблю. У оригінальному дизайні гри передбачалося, що тарілки почнуть стріляти, щойно вони з'являться, але згодом це було змінено [5]. Після досягнення 40 000 очок з'являється лише маленька тарілка. У міру того, як рахунок гравця збільшується, діапазон кутів пострілу з маленької тарілки зменшується, поки тарілка не стріляє надзвичайно точно. Як тільки екран буде очищений від усіх астероїдів та літаючих тарілок, з'явиться новий набір великих астероїдів: таким чином розпочнеться наступний рівень. Гра ускладнюється зі збільшенням кількості астероїдів, поки рахунок не досягне діапазону від 40 000 до 60 000. Гравець починає з 3–5 життів на початку гри і отримує додаткове життя за кожні 10 000 очок. Гра продовжується до

					<i>КРБ.КІ.1.440-03.3.12</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

останнього втраченого корабля, що завершує гру. Максимально високий бал, якого можна досягти, – 99 990, після чого лічильник рахунку переповнюється.

Гра Asteroids відразу ж стала успішною після випуску. Вона стала найпопулярнішою аркадною грою Atari за весь час. Atari заробила на продажі гри близько 150 мільйонів доларів, а оператори гральних автоматів заробили ще 500 мільйонів доларів на роздачі монет. «Астероїди» були настільки популярними, що деяким операторам гральних автоматів доводилося встановлювати великі ящики для зберігання монет, витрачених гравцями. У квітні 1980 року вона замінила Space Invaders на вершині чартів ігрових автоматів RePlay і стала найприбутковішою аркадною відеогрою 1980 року в Сполучених Штатах, хоча Space Invaders залишалася найкращою грою у вуличних локаціях. У 1980 році було продано 70 000 ігрових автоматів по всьому світу, з них більше 60 000 – в США в тому ж році.

Гра «Астероїди» отримала позитивні відгуки критиків відеоігор. Вільям Кессіді у огляді гри в журналі GameSpy «Classic Gaming» відзначив її інноваційність і органічність використання векторної графіки у футуристичній космічній темі. У 1995 році журнал Flux поставив аркадну версію гри на 11 місце у своєму рейтингу «100 найкращих відеоігор». Asteroids посіла четверте місце у списку Retro Gamer «25 найкращих аркадних ігор»; співробітники Retro Gamer відмічали її простоту та відзначали відсутність фіналу як можливість повернутися до гри. У 2012 році Asteroids була включена до списку 100 найбільших відеоігор за всю історію журналу Time. Entertainment Weekly назвав Asteroids однією з десяти найкращих ігор для Atari 2600 у 2013 році. Вона була додана до колекції відеоігор Музею сучасного мистецтва [16].

					КРБ.КІ.1.440-03.3.12	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Огляд подібних ігор на ринку

Конкурентами гри, що розробляється в дані роботі, є численні клони та ремейки (ігри за мотивами) оригінальної гри «Asteroids». Перші намагаються якомога точніше відтворити оригінальну гру. Ремейки використовують ту ж концепцію геймплею, але доповнюють її деякими новими функціями, іноді – з суттєвими змінами механіки, а також зміненою графікою, музикою і звуками тощо.

1.3.1 Asteroids від Free Video Games Online

Ця гра є одним з доволі точних клонів оригінальної «Asteroids» (рис. 1.5) і доступна за посиланням <https://freeasteroids.org/>. Управління здійснюється з клавіатури (пояснень щодо призначення клавіш управління немає – їх доводиться знаходити самому). Звукові ефекти та фоновий аудіо супровід також дуже схожі на оригінальні. По закінченні гри пропонується ввести ім'я користувача, але після введення імені виводиться повідомлення про неможливість запису результату.

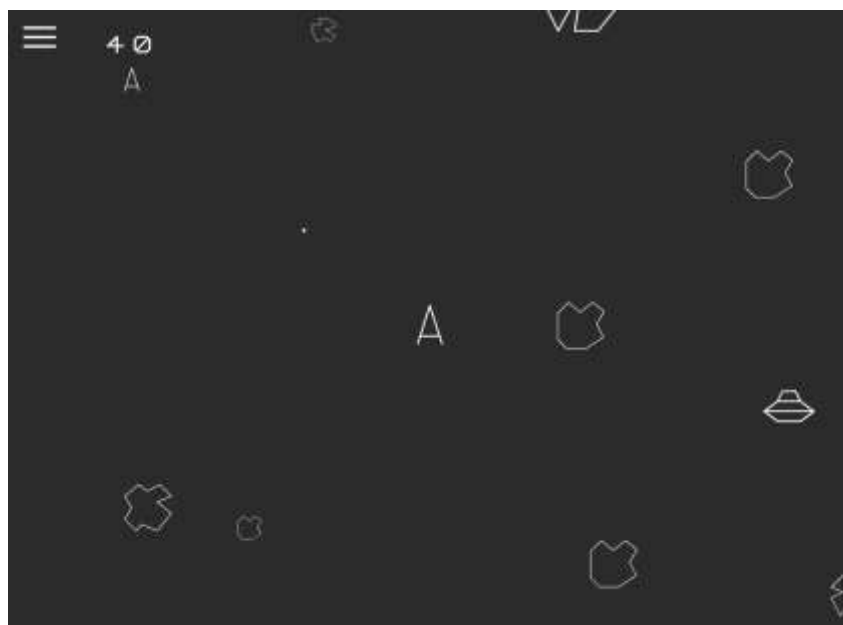


Рис. 1.5 – Браузерна гра Asteroids від Free Video Games Online

					КРБ.КІ.1.440-03.3.12	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3.2 Asteroids від AARP

Ця гра – ще один варіант гри «Asteroids», доволі схожий на оригінал, за виключенням того, що кнопки управління розташовані безпосередньо на ігровому екрані (рис. 1.6). Проте, підтримується і управління з клавіатури. Звукові ефекти та фоновий аудіо супровід також як в оригінальній грі. Підтримується реєстрація користувача і таблиця рекордів. Гра надається організацією «American Association of Retired Persons» і доступна за посиланням <https://games.aarp.org/games/atari-asteroids>.



Рис. 1.6 – Браузерна гра Atari® Asteroids від AARP

1.3.3 Asteroids від Novel Games

На відміну від попередніх, ця гра є ремейком оригінальної гри «Asteroids» від компанії Novel Games. Гра також доступна онлайн за адресою <https://www.novelgames.com/en/asteroids/>. Гра управляється стандартними клавішами клавіатури, проте має суттєво видозмінену графіку, фонову музику і звуки (рис. 1.6). Взагалі, гра здається схожою на дитячу. Можна налаштовувати гучність звуків і музики, а також перемика-тися на повноекранний режим.

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17



Рис. 1.6 – Asteroids від Novel Games

У даній грі одразу помітні певні недоліки. По-перше, на гру дається лише одне «життя», тому кожного разу при загибелі корабля гру пропонується почати заново. По-друге, часто показується реклама, яку неможливо одразу відключити – треба чекати 5 секунд.

1.3.4 Asteroids: Space Defense

Дана гра пропонується компанією MM Retro Games під платформу Android версії 5 і вище і була завантажена з Google Play більше 10 тисяч разів. Вона доволі схожа на оригінальну гру «Asteroids», хоча має змінений вигляд корабля (рис. 1.7) і інші звукові ефекти, доволі приємні. Гра управляється за допомогою сенсорного екрану і цим суттєво відрізняється від оригіналу. Автори позиціонують гру як «олдскульну аркаду з високоякісним піксельним арт-стилем, плавним та регульованим управлінням, а також складними хвилями (ворогів)».

					КРБ.КІ.1.440-03.3.12	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 1.7 – Asteroids: Space Defense від MM Retro Games

У гру можна грати безкоштовно, але після кожної загибелі корабля показується реклама, а по закінченню рівня пропонується подивитися рекламне відео.

1.4 Узагальнена постановка завдання

Розроблювана гра (умовна назва «Астероїди») розрахована на платформу Windows і у базовому варіанті реалізуватиме основу оригінального концепту гри «Asteroids» з деякими змінами:

- гра повинна мати більш реалістичний вигляд астероїдів і корабля;
- корабель повинен мати можливість рухатися не тільки вперед, а й назад.

Складність проходження гри має бути збалансованою і нарощуватися у міру проходження гри і накопичення очок. Мета гри – накопичити щонайбільше очок.

					КРБ.КІ.1.440-03.3.12	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

1.5 Підготування робочого місця

Для створення проекту гри було прийнято рішення використовувати середовище ігрового рушія Unity.

Unity – це програмний рушій для створення, управління та розширення мобільних, настільних комп’ютерних ігор, а також ігор для ігрових консолей.

Для встановлення *Unity* були виконані певні кроки.

На веб-сторінці за посиланням <https://unity3d.com/ru/get-unity/download> натиснули кнопку «Завантажити Unity Hub» (рис. 1.8). Вибрали місце для завантаження файлу та зберегли інсталятор.

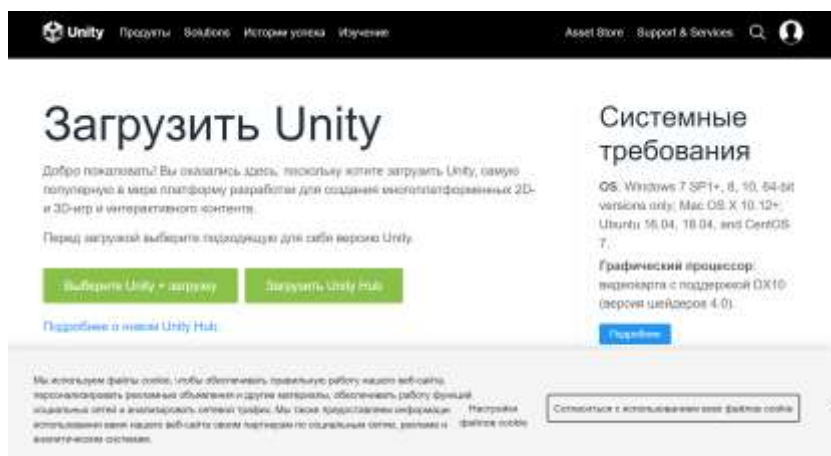


Рис. 1.8 – Підготовка до завантаження Unity Hub

Далі запустили інсталятор, прочитали та погодились з умовами обслуговування (рис. 1.9).

Пройшли далі майстер встановлення (рис. 1.10).

Запустили встановлену програму Unity Hub і дозволили доступ до Інтернету (рис. 1.11).

					КРБ.КІ.1.440-03.3.12	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 1.9 – Умови обслуговування Unity

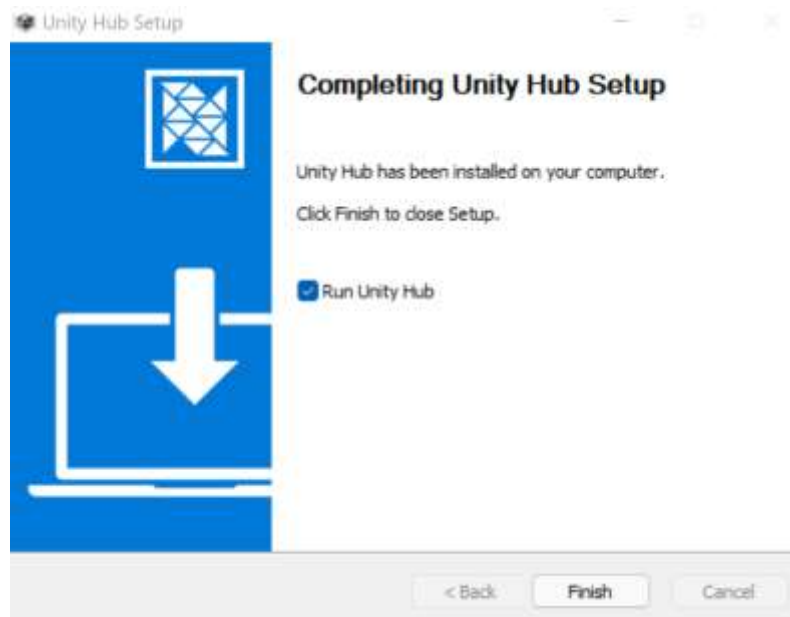


Рис. 1.10 – Майстер інсталяції Unity Hub

Далі необхідно було зареєструватися (створити акаунт) та отримати ліцензію на використання Unity.

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

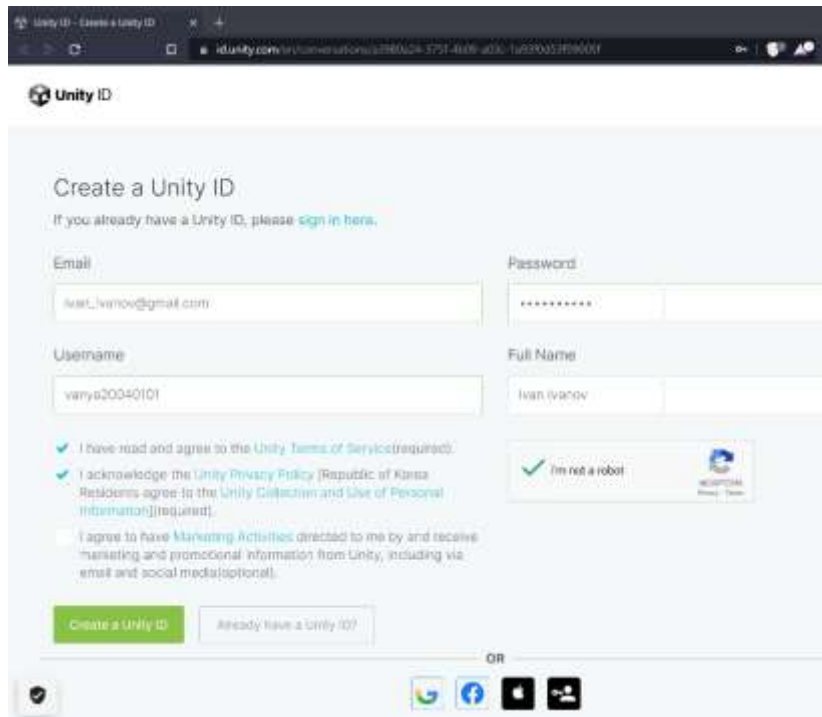


Рис. 1.12 – Реєстрація в Unity

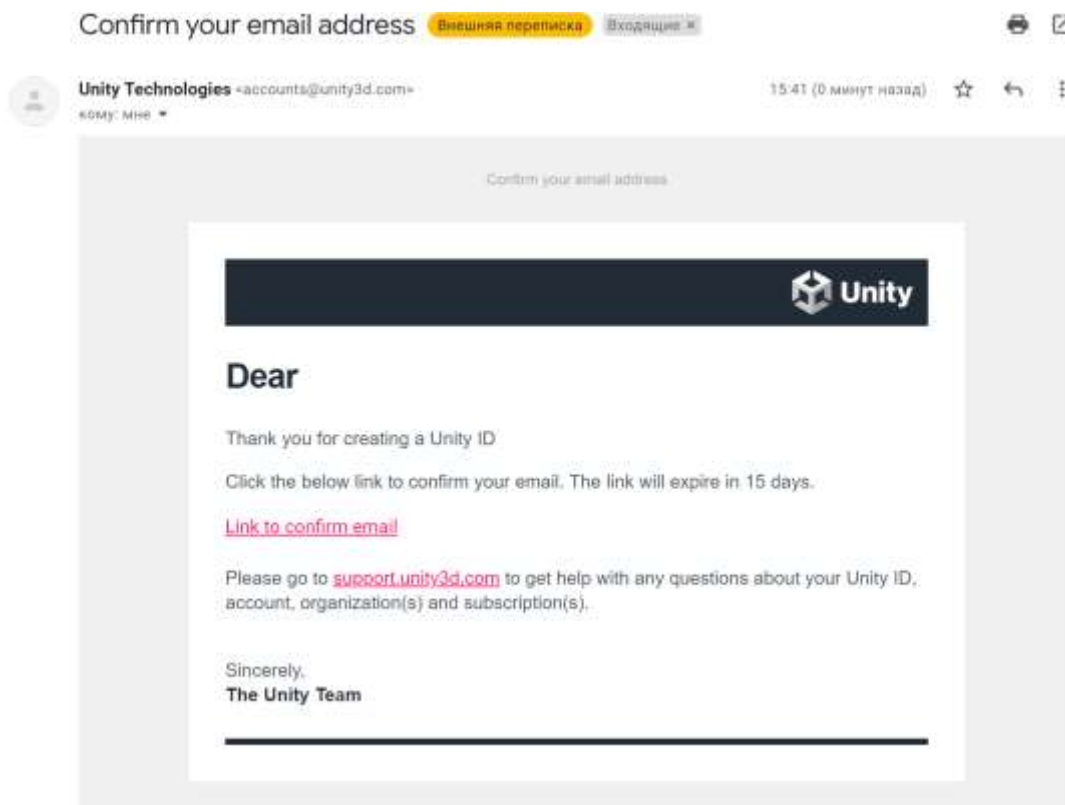


Рис. 1.13 – Підтвердження поштової адреси

Далі у вікні, яке просить відкрити Unity Hub, було натиснуто кнопку *Open Unity Hub* (рис. 1.14). На цьому реєстрацію в Unity було завершено.

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

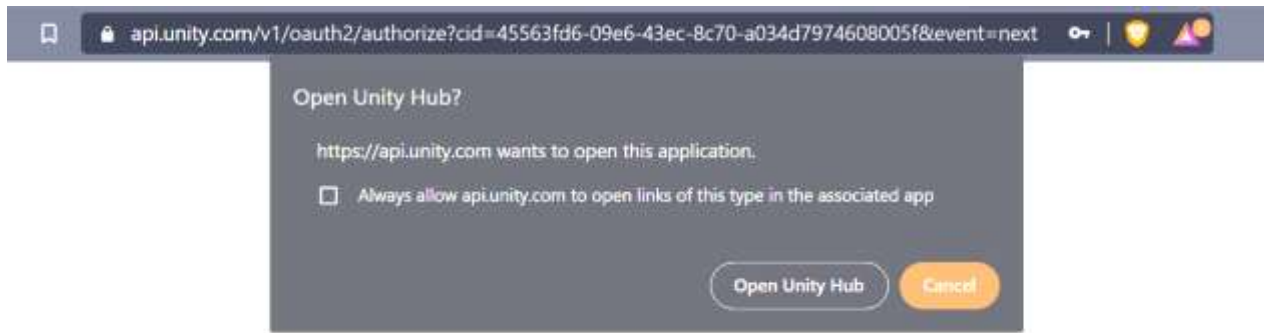


Рис. 1.14 – Завершення реєстрації в Unity

8. Для створення проектів в Unity необхідна активна ліцензія. Для управління ліцензіями у вікні Unity Hub ми натиснули на значок облікового запису (у верхньому лівому куті), а потім вибрали пункт «*Manage licenses*» (рис. 1.15). Далі вікні налаштувань натиснули на кнопку «*Add*» (можна також натискати на кнопку «*Add license*»).

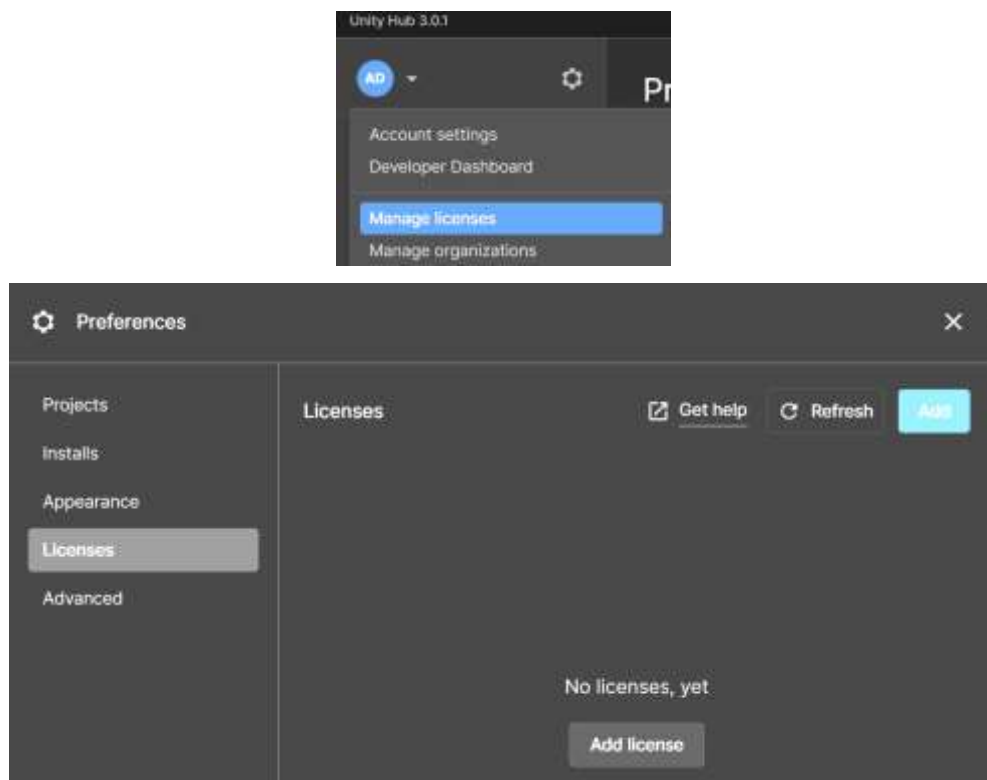


Рис. 1.15 – Додавання ліцензії Unity

Unity надає розробникам безкоштовну ліцензію *Personal Edition* на таких умовах:

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

- компанія не заробляє понад 100 тисяч доларів річного валового доходу, незалежно від того, чи використовується Unity Personal у комерційних цілях, для внутрішнього проекту чи прототипування;
- компанія не залучає коштів понад 100 тисяч доларів;
- в даний час не використовується тарифний план *Unity Plus* або *Pro*.

Для даної дипломної роботи такі умови цілком прийнятні.

Для запиту персональної безкоштовної ліцензії ми натиснули на кнопку «*Get a free personal license*», а у наступному вікні – на кнопку «*Agree and get personal edition license*» (рис. 1.16).

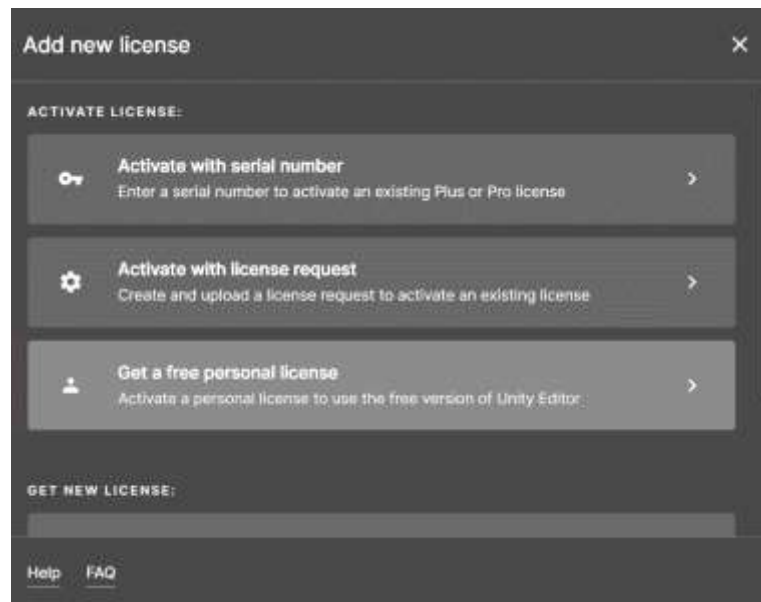


Рис. 1.16 – Отримання безкоштовної персональної ліцензії Unity

Наступний етап – встановлення редактора. Для цього ми перейшли в розділ для налаштувань «*Installs*» і натиснули на кнопку «*Install Editor*» (рис. 1.17).

Далі натиснули кнопку «*Install*» праворуч від рекомендованої версії, зняли прапорець установки Microsoft Visual Studio Community (звідси встановилася б застаріла і неповна версія) і, нарешті, натиснули кнопку «*Install*». Процес встановлення зайняв близько 5 хвилин.

										Арк.
										25
Змн.	Арк.	№ докум.	Підпис	Дата						

КРБ.КІ.1.440-03.3.12

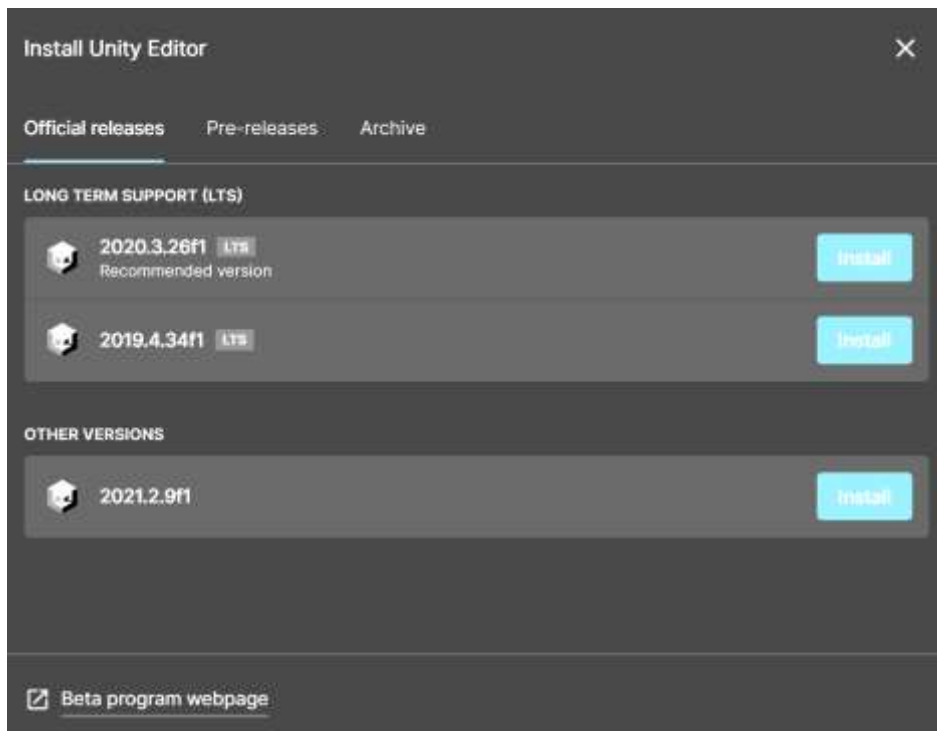


Рис. 1.17 – Встановлення редактора Unity

Тепер можна створювати новий проект гри – для цього слід перейти до розділу *Projects* і натиснути кнопку «*New project*».

Висновки першого розділу

Дослідження, які було проведено в розділі, обґрунтовують актуальність теми і доцільність подальшої роботи. Було також здійснено загальну постановку завдання на дипломну роботу.

Отримані результати і загальна постановка завдання дозволяють приступити до проектування аркадної гри обраного жанру.

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

РОЗДІЛ 2

ПРОЕКТУВАННЯ КОМП'ЮТЕРНОЇ АРКАДНОЇ ГРИ

2.1 Технічне завдання на розробку комп'ютерної логічної гри

2.1.1 Загальна інформація про проект

Комп'ютерна гра – римейк класичної аркадної гри «Asteroids».

Робоча назва – «Астероїди».

Цільова платформа – Windows (версій 7 і вище).

2.1.2 Функціонал гри

Гравець керує космічним кораблем, розташованим початково в середині сцени. Корабель можна повертати вліво і вправо, рухати вперед і назад, а також стріляти кулями.

Після початку гри на екрані поступово з'являються астероїди і рухаються приблизно у напрямку корабля. Мета гри – знищувати астероїди і уникати зіткнення з ними, накопичуючи максимальну кількість очок.

Астероїди діляться на три категорії за розміром. Великий астероїд, після того, як у нього потрапляє постріл, розпадається на два астероїди середніх розмірів. Кожний з цих середніх астероїдів, у свою чергу, розбивається пострілом на два маленькі астероїди.

За знищення об'єктів нараховуються очки: 100 за маленький астероїд, 50 за середній, 20 за великий. Кількість куль не обмежена, але за витрачені кулі бали віднімаються: –3 очка за кожен кулю.

По мірі продовження гри вона ускладнюється: кількість астероїдів збільшується.

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Якщо корабель стикається з астероїдом, він гине. Початково дається 4 «життя» (корабля). Якщо після загибелі корабля кількість життів залишається не нульовою, корабель відновлюється у тій самій сцені і тимчасово (на 3 секунди) набуває здатності не стикатися з астероїдами. Набравши 10000 очок, гравець отримує додаткове життя. Після втрати всіх життів гра завершується, і гравець може зафіксувати своє досягнення в таблиці.

2.1.3 Нефункціональні вимоги до гри

1. Підтримка усіх версій Windows, починаючи з 7.
2. Красивий вигляд астероїдів.
3. Ефект вибуху.
4. Зручне і збалансоване керування кораблем.
5. Невеликі апаратні вимоги: 1 Гбайт оперативної пам'яті, 100 Мбайт вільного місця на зовнішньому носії.
6. Стабільність в роботі: не більше одного нештатного завершення програми за тиждень експлуатації тисячею гравців.

2.1.4 Організаційні оцінки

Проект планується розвивати за інкрементно-спіральною стратегією, яка передбачає генерацію все більш повноцінної робочої версії застосунку на кожному витку спіралі розвитку. На перший виток спіралі планується витратити загалом 2 місяці, конкретно – приблизно 300 годин роботи спеціалістів з системного інжинірингу, програмування та дизайну користувацького інтерфейсу.

Проект може бути монетизований шляхом розміщення реклами в застосунку. Більш докладний маркетинговий аналіз наведений в розділі 4.

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

2.2 Вибір інструментів розробки

В якості основного інструменту розробки гри обраний ігровий рушій Unity. Unity є однією з найпопулярніших та потужних платформ для розробки ігор, існуючих на сьогоднішній день. До переваг Unity належать:

1. Кросплатформенність. Unity надає можливість розробляти гри, які можуть працювати на різних платформах, включаючи ПК, мобільні пристрої, консолі, веб-браузери тощо. Це дозволяє досягти більшої аудиторії та забезпечити максимальну доступність гри для користувачів.
2. Зручність використання. Unity має інтуїтивно зрозумілий інтерфейс, що полегшує розробку гри навіть для початківців. Інтегрована розробка на основі компонентів (component-based development) дозволяє легко керувати об'єктами та функціями гри.
3. Велика спільнота розробників: Unity має широку спільноту розробників, що дозволяє отримувати підтримку, поради та допомогу від інших професіоналів. Існує велика кількість онлайн-ресурсів, документації та уроків, які полегшують процес розробки.
4. Розширюваність: Unity надає можливість використовувати різноманітні розширення, плагіни та активи, які значно розширюють функціональність і можливості платформи. Це дозволяє швидко впроваджувати нові функції та використовувати спеціалізовані рішення для розробки конкретних ефектів, штучного інтелекту, фізики та інших аспектів гри.
5. Підтримка 2D- та 3D-графіки: Unity забезпечує потужні інструменти для розробки як 2D-, так і 3D-графіки. Це дозволяє створювати різноманітні та вражаючі візуальні ефекти, анімацію та деталізовані ігрові світи.
6. Можливості розробки мультиплеєра: Unity надає рішення для розробки мультиплеєрних ігор з можливістю взаємодії гравців через різні

										Арк.
										29
Змн.	Арк.	№ докум.	Підпис	Дата						

КРБ.КІ.1.440-03.3.12

протоколи і мережеві технології. Це дозволяє створювати гри з он-лайн-режимом, спільною грою та багатокористувацькими функціями.

Далі наведена порівняльна характеристика Unity з деякими іншими популярними ігровими рушіями.

Unreal Engine – має високу якість графіки та візуальних ефектів, потужний інструментарій для розробки фотореалістичних ігор, вбудована система фізики та штучного інтелекту.

CryEngine – також характеризується реалістичною графікою та освітленням, має доволі потужні інструменти для взаємодії з ігровим середовищем і деталізовану роботу з текстурами.

Godot Engine відрізняється легкістю використання, вбудованою системою скриптування та візуального редагування, малим розміром виконуваних файлів.

GameMaker Studio – ще один популярний рушій, досить простий у використанні для початківців. Він забезпечує швидкість розробки простих ігор, гнучку систему ресурсів.

В той же час, Unity має більш широкий набір розширень та плагінів, кращу підтримку мобільних платформ, інтегрований магазин активів та ресурсів. Таким чином, завдяки своїм перевагам Unity є досить привабливим вибором для розробки гри в даній роботі.

2.3 Структура проекту Unity

Структура Unity проекту гри MyAsteroids включає ресурси (assets) таких типів:

- сценарії на мові програмування C#, які описують класи програмних об'єктів і інтегруються з існуючою системою класів і компонентів бібліотеки Unity;
- ігрові об'єкти, які взаємодіють з програмними об'єктами;

									Арк.
									30
Змн.	Арк.	№ докум.	Підпис	Дата					

КРБ.КІ.1.440-03.3.12

- $spawnRate = 3$ с – період у секундах між породженнями астероїдів;
- $amountPerSpawn = 1$ – кількість астероїдів, що породжуються одно-моментно;
- $trajectoryVariance = \pm 15^\circ$ – кут розходження можливих траєкторій астероїда.

Методи класу *AsteroidSpawner*:

- *Start()* – ініціалізація породжувача і запуск процесу генерації астероїдів із заданою частотою;
- *Spawn()* – породження $amountPerSpawn$ астероїдів і їхній запуск у певному напрямі в межах кута траєкторій (рис. 2.1).

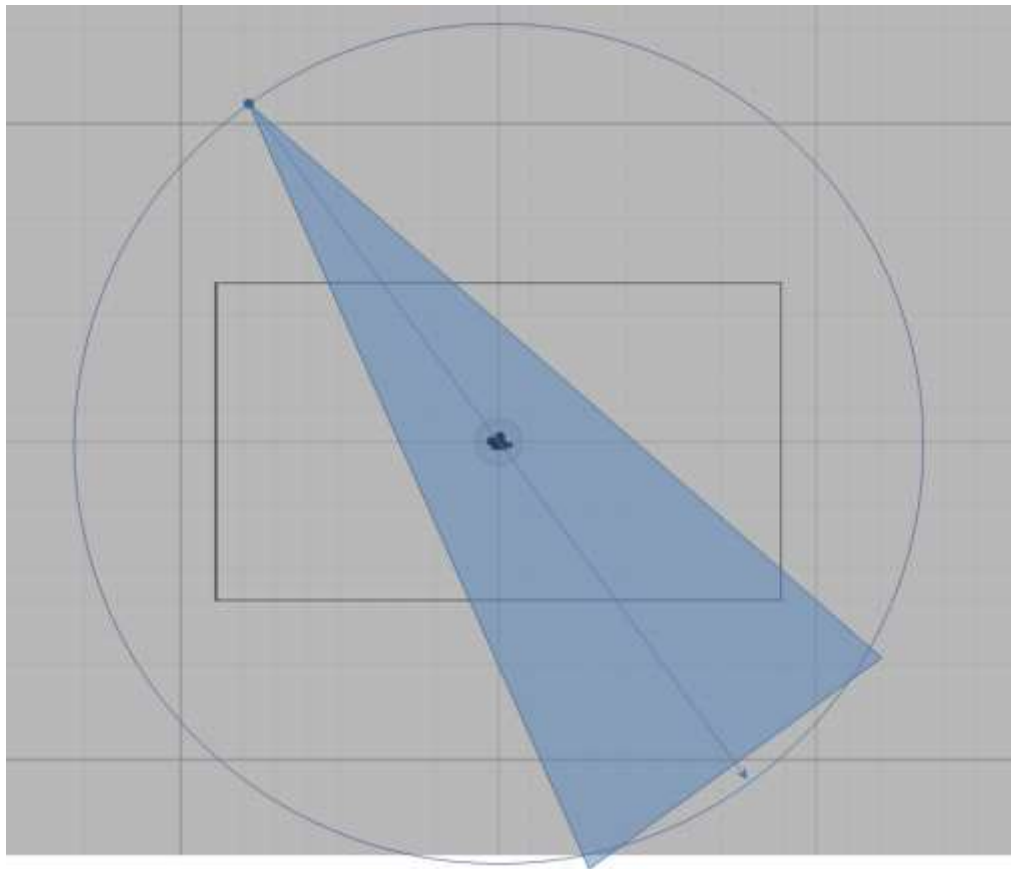


Рис. 2.1 — Точка появи астероїда і зона можливих траєкторій його руху

Bullet – клас куль, якими стріляє корабель гравця. Він описує такі параметри:

- $speed = 500$ – швидкість руху кулі;

									Арк.
									33
Змн.	Арк.	№ докум.	Підпис	Дата	КРБ.КІ.1.440-03.3.12				

– *maxLifetime* = 10 с – максимальний час існування кулі.

Методи класу *Bullet*:

- *Project(direction)* – спрямування кулі у заданому напрямі;
- *OnCollisionEnter2D()* – обробка зіткнення кулі з іншими об'єктами.

GameManager – клас ігрового менеджера, який відповідає за облік життів гравця, загибель корабля і його відродження, а також за закінчення гри.

Поля класу і їх значення:

- *respawnTime* = 3 с – час між загибеллю корабля і його подальшим відродженням;
- *respawnInvulnerabilityTime* = 3 с – час тимчасової невразливості корабля після його відродження;
- *lives* = 3 – початкова кількість життів;
- *score* = 0 – ігровий лічильник очок;
- *explosion* – об'єкт вибуху.

Методи класу *GameManager*:

- *AsteroidDestroyed(asteroid)* – обробка події знищення астероїда;
- *PlayerDied()* – обробка події загибелі корабля;
- *Respawn()* – відродження корабля після його загибелі;
- *TurnOnCollisions()* – увімкнення механізму зіткнень після закінчення терміну тимчасової невразливості корабля;
- *GameOver()* – завершення гри після закінчення життів.

UML-діаграма класів показана на рис. 2.2.

					КРБ.КІ.1.440-03.3.12	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.2 – Структура класів проекту MyAsteroids

2.4 Керування кораблем в грі

Керування кораблем в даній грі складається з таких команд і способів їх віддати (в дужках вказані управляючі змінні з відповідними значеннями) (рис. 2.3):

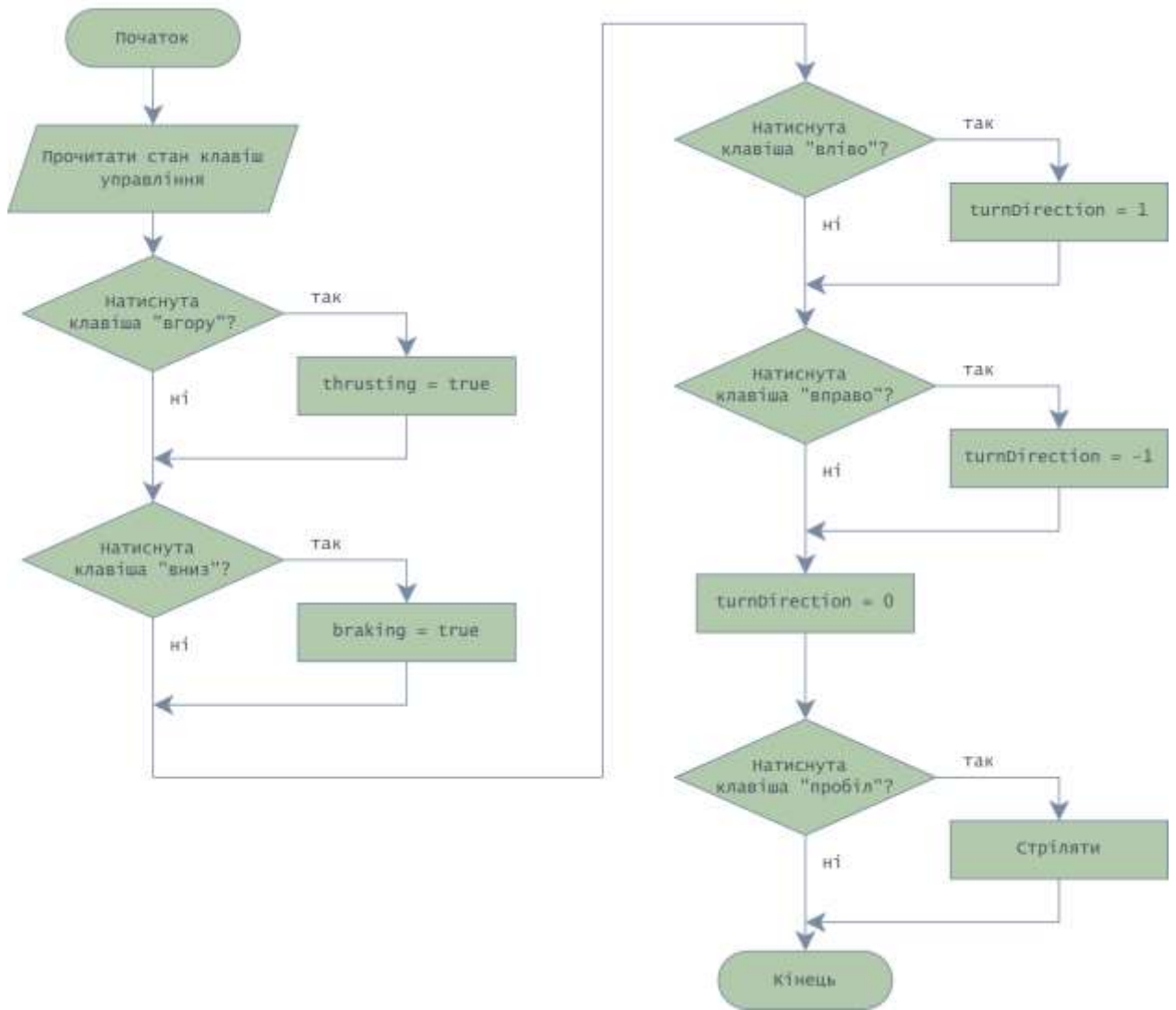


Рис. 2.4 – Алгоритм керування кораблем гравця з клавіатури

Висновки другого розділу

В результаті виконання даного етапу роботи були розроблені технічне завдання, вибрані засоби реалізації проекту аркадної гри «Астероїди». Також в даному розділі представлені проектні рішення, які стосуються структурних і поведінкових аспектів ігрового застосунку. Вони дають змогу перейти до практичної реалізації гри з використанням відповідних інструментальних засобів в обраному середовищі розробки Unity.

									Арк.
									37
Змн.	Арк.	№ докум.	Підпис	Дата					

РОЗДІЛ 3 РОЗРОБКА ЗАСТОСУНКУ В UNITY

3.1 Створення проекту і налаштування сцени

Для початку створюємо проект в Unity Hub. Оскільки гра в нас буде плоска, обираємо шаблон 2D Core (рис. 3.1).

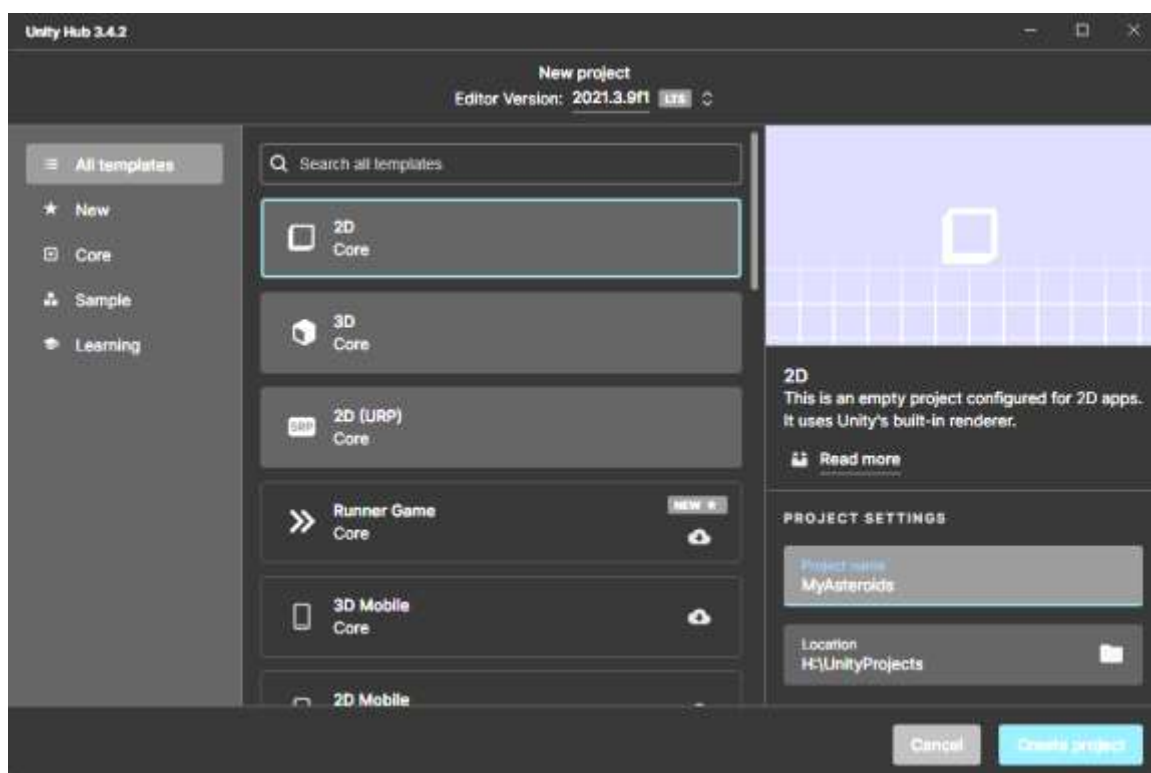


Рис. 3.1 — Створення проекту в Unity Hub

Змінимо стандартну назву сцени на *Asteroids* (рис. 3.2).

Змінимо колір фону камери на чорний (рис. 3.3).

Далі нам треба створити границі нашого ігрового екрану. Для цього створимо порожній ігровий об'єкт *Boundary*, додамо до нього компонент *Box Collider 2D* і налаштуємо його положення й розміри так, щоб він став верхньою границею нашого екрану.

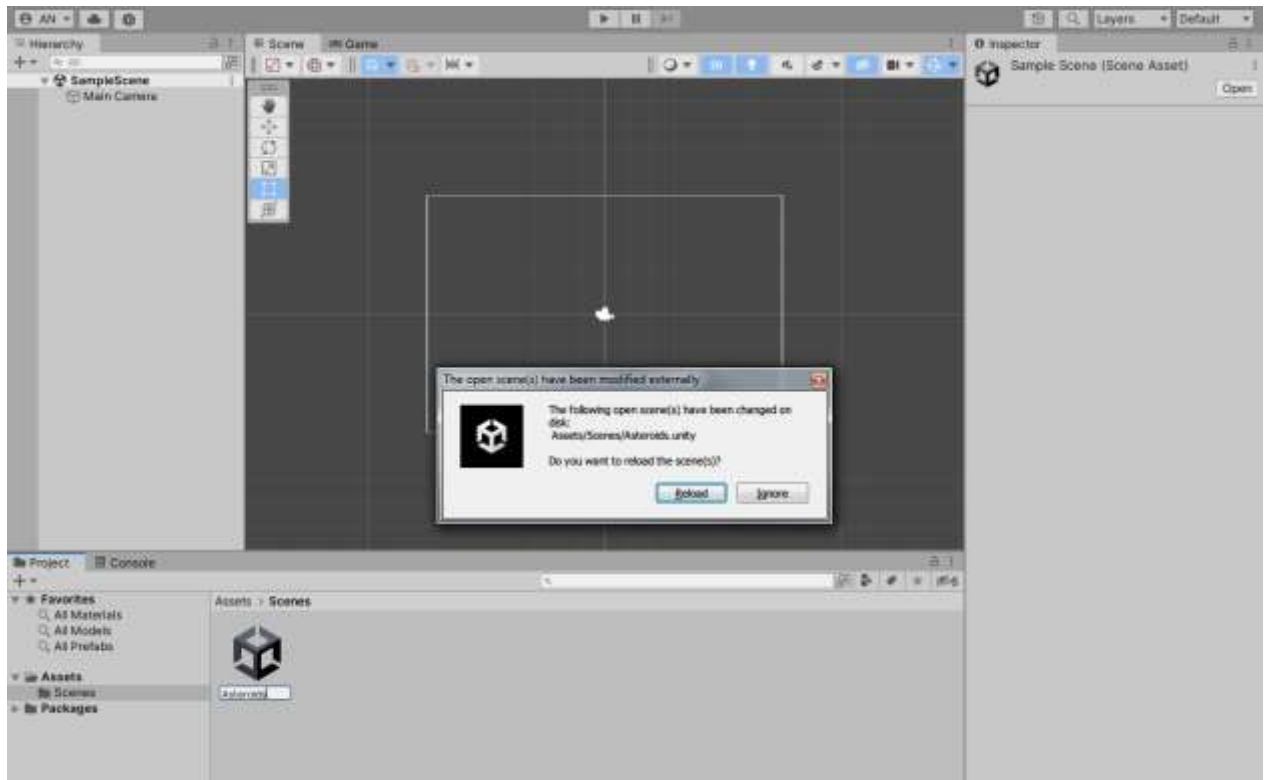


Рис. 3.2 – Перейменування сцени

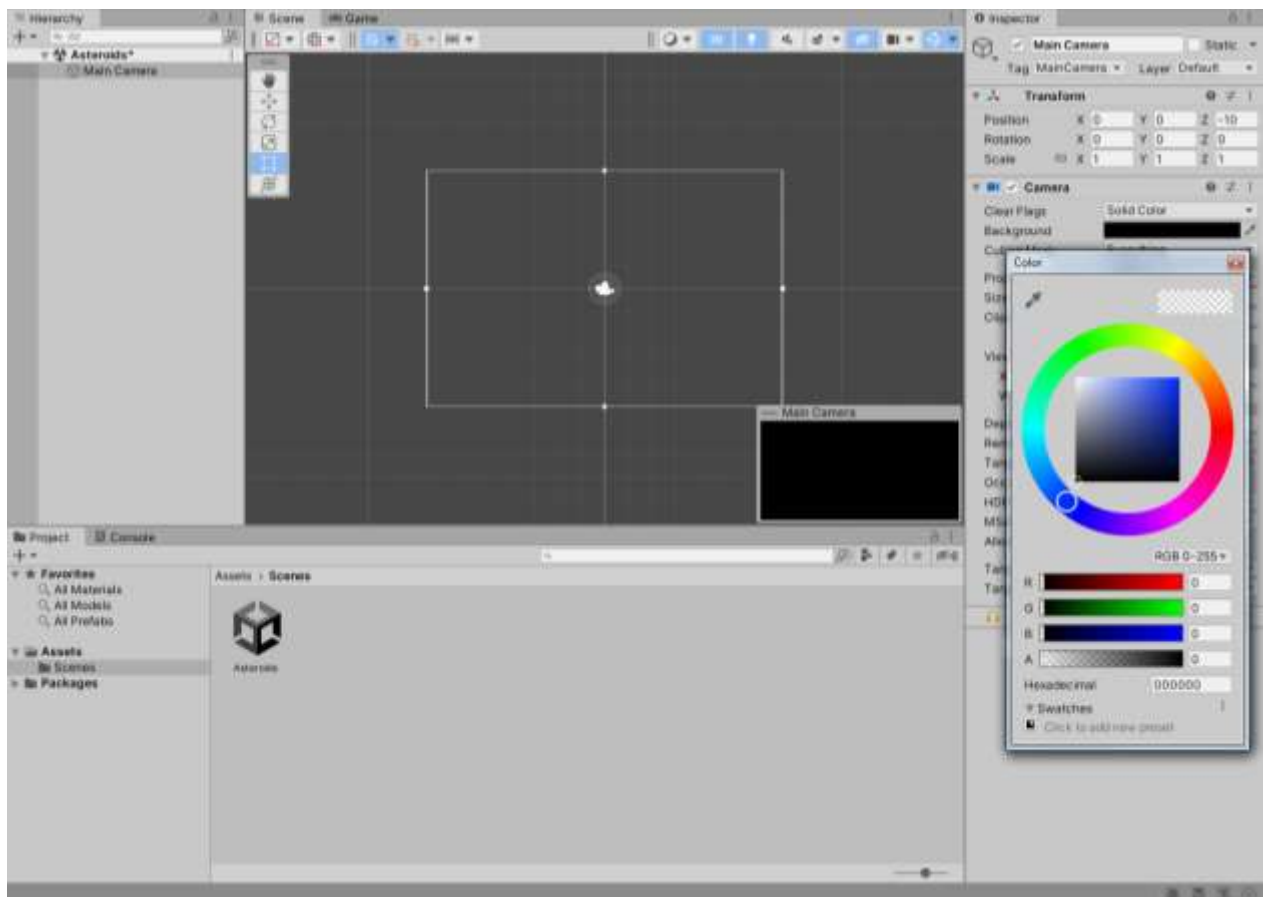


Рис. 3.3 – Налаштування фонового кольору камери

Потім дублюємо об'єкт *Boundary* і цю копію переміщаємо в нижню частину екрану. Далі створюємо ще дві копії, орієнтуємо вертикально і розміщуємо ліворуч та праворуч від границь поля зору нашої камери. Ми отримали чотири свого роду стіни, які обмежують ігрове поле (рис. 3.4).

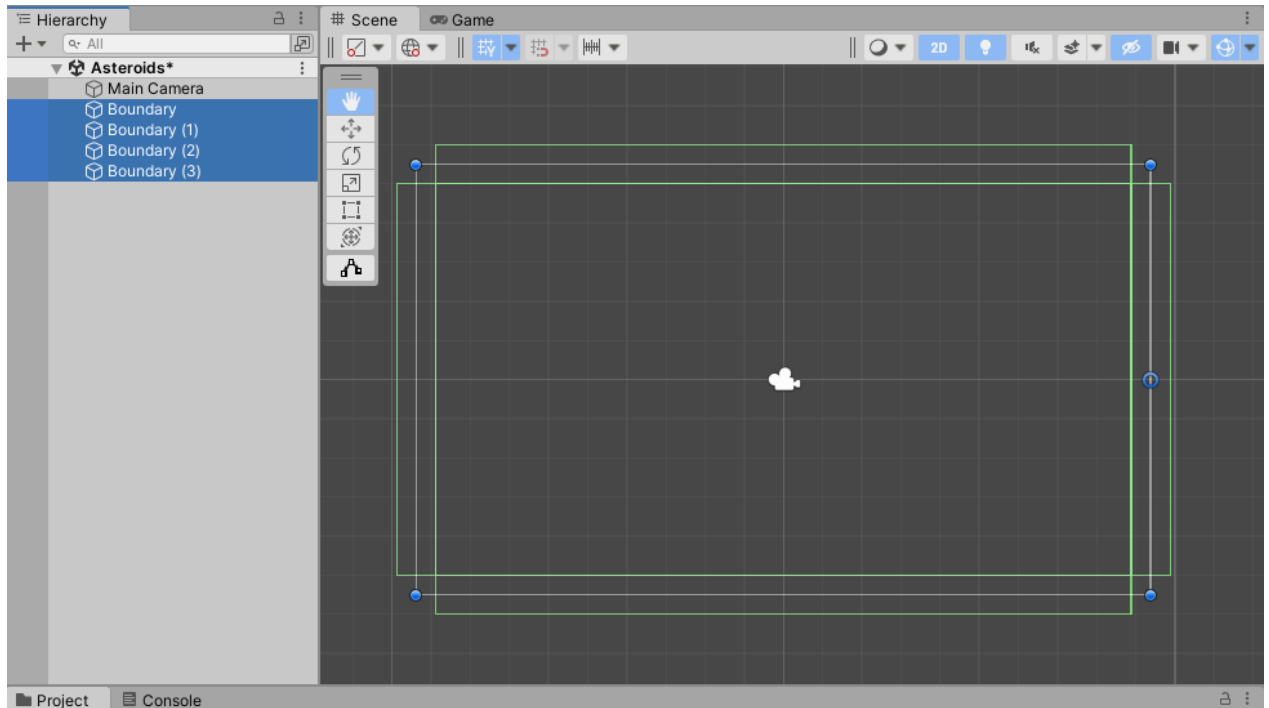


Рис. 3.4 — Створення границь ігрового простору

3.2 Створення ігрового персонажа – корабля

Далі нам треба створити головного персонажа гри. Для цього знову створюємо порожній ігровий об'єкт під назвою *Player* і додаємо до нього компонент *Sprite Renderer*. Цей компонент дозволить нам відображати зображення у вигляді спрайтів, щоб використовувати їх в нашій 2D-сцені. В якості спрайта для зовнішнього вигляду нашого космічного корабля використаємо готове зображення з сайту <https://www.freepng.ru/> (рис. 3.5).

					КРБ.КІ.1.440-03.3.12	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

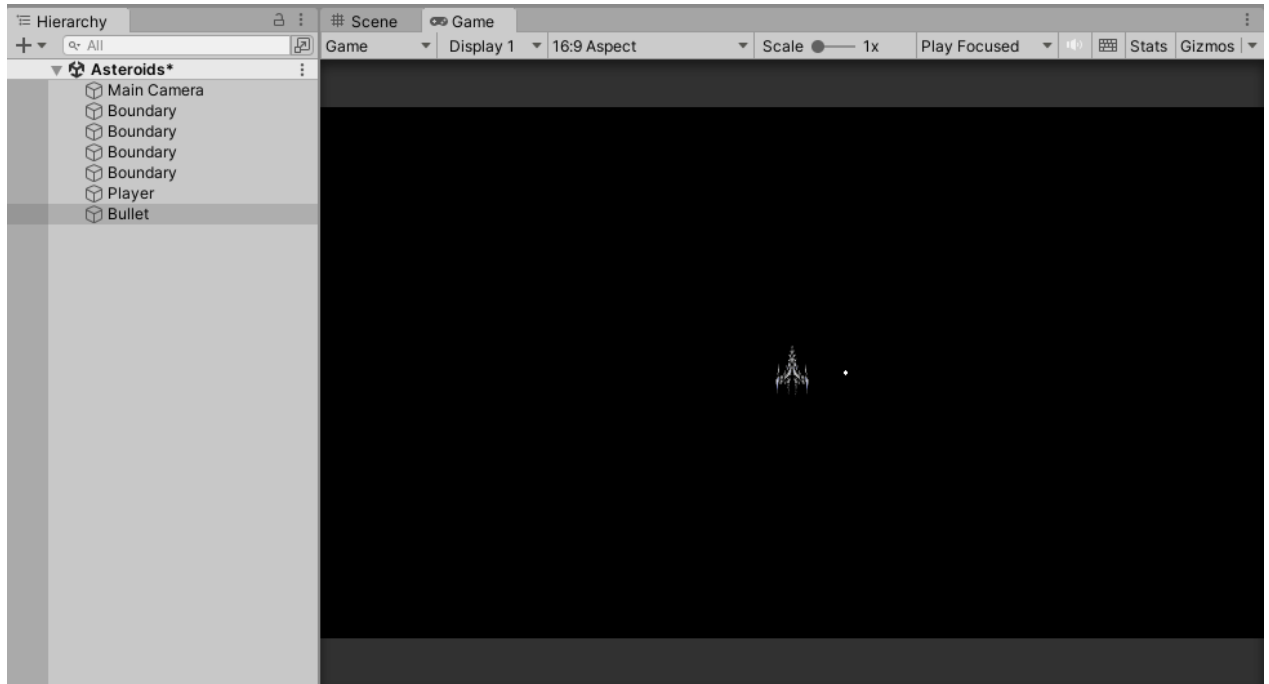


Рис. 3.11 — Створення об'єкту-кулі

Оскільки куль в нас буде багато, і вони мають поводитися однаково, доцільно на основі нашого об'єкту створити так званий префаб – користувачський шаблон для створення множини подібних об'єктів. Тому створимо ще одну папку в проекті – *Prefabs* – і перетягнемо туди наш об'єкт *Bullet*. Це призводить до створення префабу кулі – *Bullet.prefabs* (рис. 3.12).

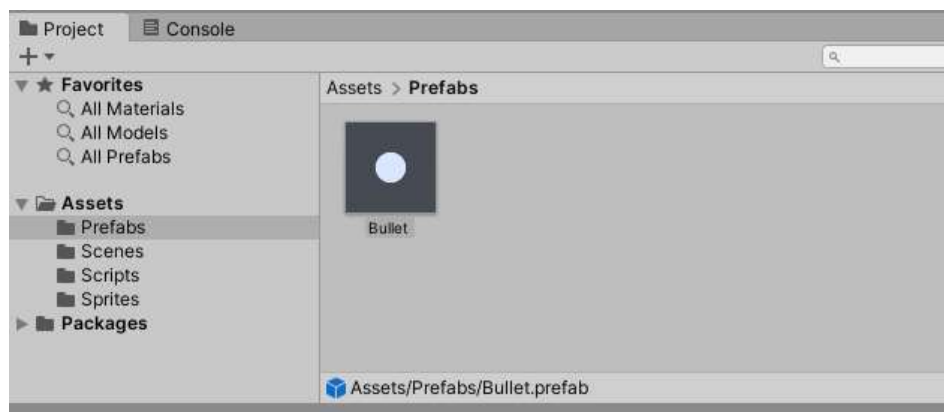


Рис. 3.12 — Створення префабу кулі

Ми зможемо користуватися новим префабом, створюючи інші кулі, зі сценаріїв та іншими способами. Тому, власне, нам вже не потрібен окремий об'єкт *Bullet*, і ми можемо видалити його зі сцени.

Тепер запрограмуємо поведінку кулі. Для цього створимо новий скрипт *Bullet.cs* і приєднаємо його до префабу *Bullet*. Після цього можна відкрити його у Visual Studio для редагування.

Нам треба спрямувати кулю у певному напрямку, ґрунтуючись на поточному положенні корабля. Оформимо ці дії у вигляді методу *Project()*. Нам знадобиться посилання на об'єкт *Rigidbody2D* і змінна для швидкості кулі. Після запуску необхідно буде знищити об'єкт кулі через деякий час (не менший, ніж час польоту кулі по ігровому екрану) – для цього введемо змінну *maxLifetime* і задамо їй значення 10 с.

```
using UnityEngine;

public class Bullet : MonoBehaviour
{

    public float speed = 500.0f;
    public float maxLifetime = 10.0f;
    private Rigidbody2D _rigidbody;

    private void Awake()
    {
        _rigidbody = GetComponent<Rigidbody2D>();
    }

    public void Project(Vector2 direction)
    {
        _rigidbody.AddForce(direction * this.speed); // надаємо імпульс кулі
        Destroy(this.gameObject, this.maxLifetime); // знищуємо її через 10 с
    }
}
```

У сценарії *Player.cs* нам знадобиться посилання на префаб кулі і функція пострілу – *Shoot()*. В цій функції ми будемо створювати екземпляр кулі, після чого запускати її у напрямку корабля викликом функції *Project()*: надавати кулі імпульс і потім знищувати через 10 секунд.

Після збереження скриптів зв'яжемо префаб кулі з ігровим об'єктом *Player* (рис. 3.13).

									Арк.
									47
Змн.	Арк.	№ докум.	Підпис	Дата					

Далі треба налаштувати взаємодію шарів у налаштування проекту (Project Settings), в категорії Physics 2D, а саме матрицю Layer Collision Matrix, яка визначає, які шари стикаються між собою (рис. 3.16).

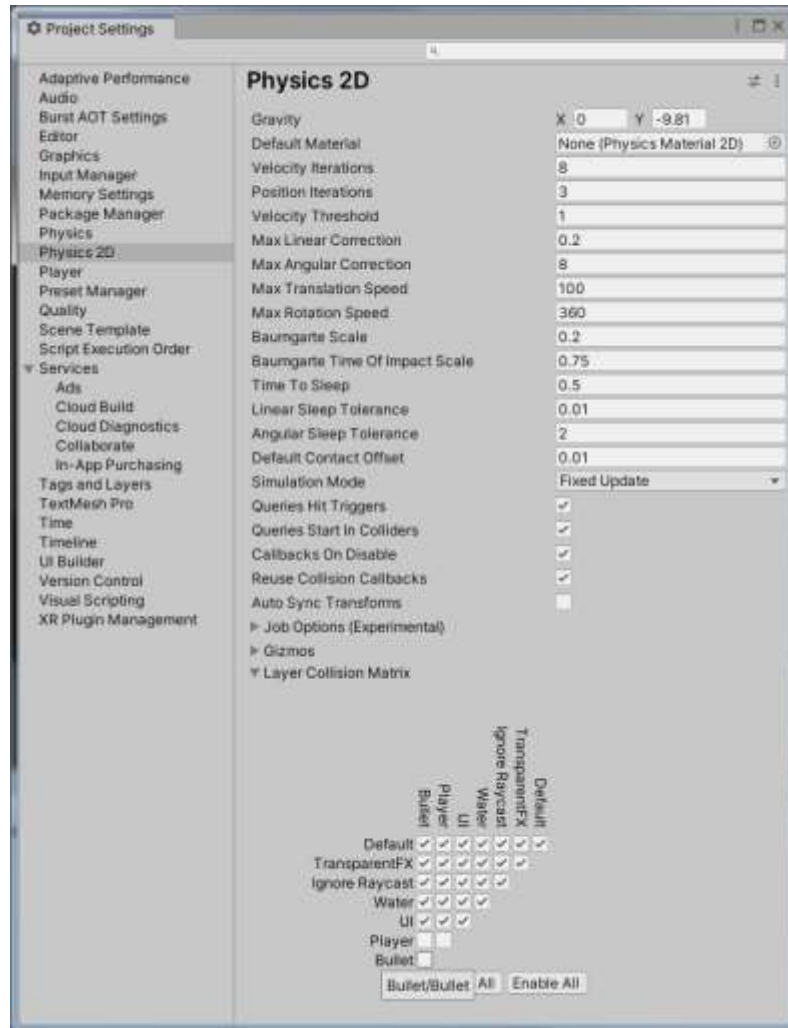


Рис. 3.16 – Обрання шарів для об'єкту *Player* і префабу *Bullet*

Нам не треба, щоб корабель гравця взаємодіяв з кулями і щоб кулі взаємодіяли між собою, тому знімаємо відповідні прапорці на перетині назв шарів.

Другий момент, який потребує виправлення, це зіткнення кулі з границями екрану. При зіткненні кулі з будь-чим вона має знищуватися (і, можливо, знищувати об'єкт, з яким вона зіткнулася, але не стіни нашого екрану). Для знищення кулі при зіткненні додамо до класу *Bullet* метод *OnCollisionEnter2D()*:

					КРБ.КІ.1.440-03.3.12	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

```
private void OnCollisionEnter2D(Collision2D collision)
{
    Destroy(this.gameObject);
}
```

3.4 Створення астероїдів

Наступний етап – створення астероїдів. Вільні для використання зображення астероїдів на прозорому фоні було знайдено на сайті <https://www.pngall.com/>. Після завантаження ці зображення було оброблено в графічному редакторі PhotoFiltre: зменшено розмір, щоб картинки не займали багато місця серед ресурсів гри, а також, щоб ширина картинки дорівнювала висоті і була кратною 4 (120×120 пікселів), переведено в режим градацій сірого, повернуто так, щоб освітлення було з одного боку, а також налаштовано гаму, щоб вони усі були подібні за насиченістю (рис. 3.17).



Рис. 3.17 — Зображення астероїдів

Після обробки додамо збережені зображення в папку *Sprites* проекту.

Далі створимо новий порожній об'єкт *Asteroid* і додамо до нього компоненти *Sprite Renderer*, *Rigidbody 2D* і *Box Collider 2D*. Попередньо налаштуємо масштабування об'єкту (параметр *Scale* = 0,5 для осей *x* та *y*) і розміри колайдера (параметр *Size* = 1 також для *X* та *Y*). З компонентом зв'яжемо одне з зображень астероїдів (рис. 3.18).

Далі створимо на основі створеного об'єкту префаб і видалимо об'єкт, оскільки він більше не потрібен. Тепер створимо скрипт для управління астероїдами – *Asteroid.cs* і прив'яжемо його до створеного префабу.

										Арк.
										50
Змн.	Арк.	№ докум.	Підпис	Дата						

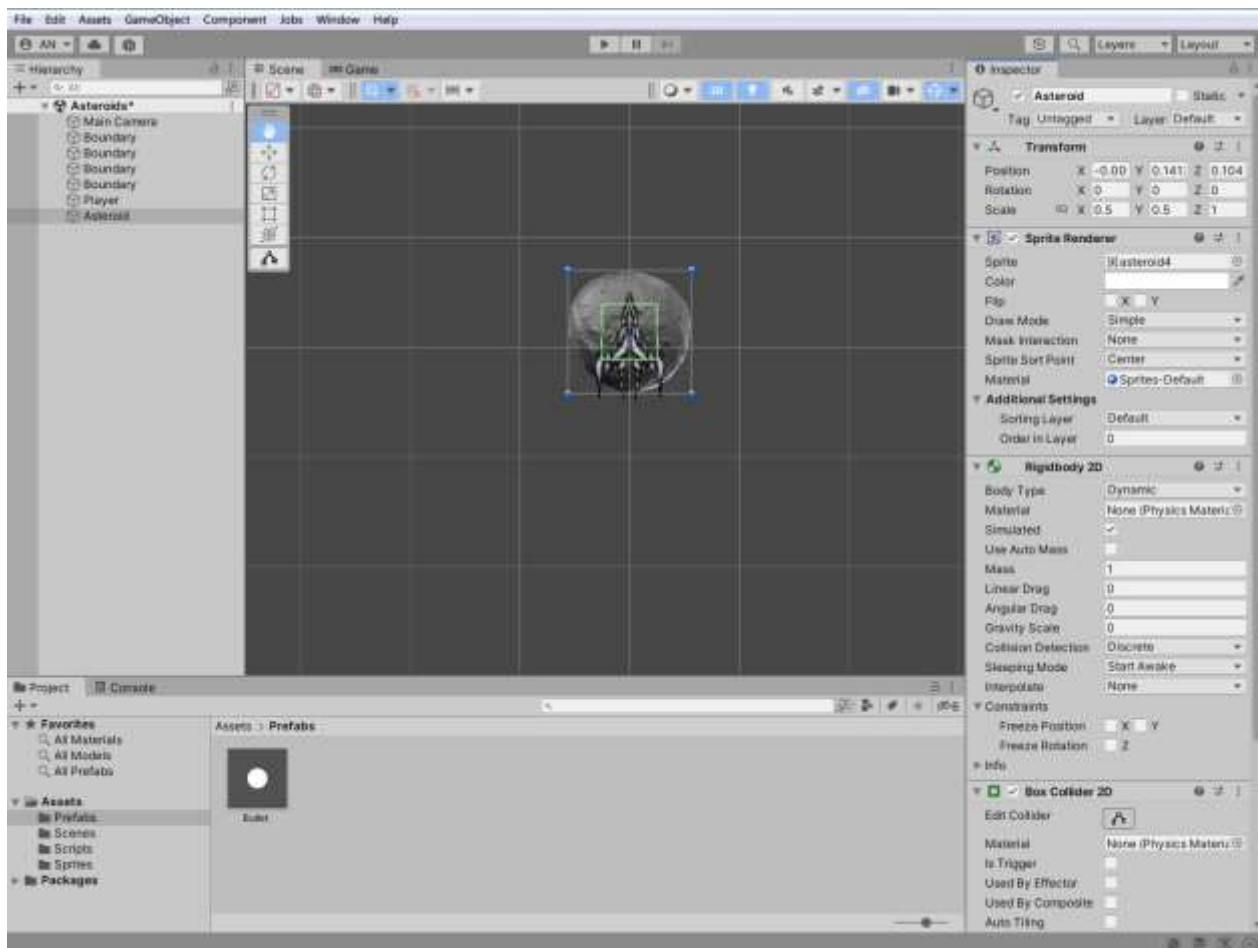


Рис. 3.18 — Об'єкт *Asteroid*

Оскільки астероїди мають різнитися один від одного зовнішнім виглядом, розмірами та іншими властивостями, у скрипті *Asteroid.cs* треба задати певні випадкові дані: спрайт, розмір, напрям. Маса астероїдів залежатиме від його розміру. Ці дані задамо у методі *Update()*. Отриманий код скрипта:

```
using UnityEngine;

public class Asteroid : MonoBehaviour
{
    public Sprite[] sprites;
    public float size = 1.0f;
    public float minSize = 0.5f;
    public float maxSize = 1.5f;
    private SpriteRenderer _spriteRenderer;
    private Rigidbody2D _rigidbody;

    private void Awake()
    {
        _spriteRenderer = GetComponent<SpriteRenderer>();
        _rigidbody = GetComponent<Rigidbody2D>();
    }
}
```

					<i>КРБ.КІ.1.440-03.3.12</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

```

// Start is called before the first frame update
private void Start()
{
    _spriteRenderer.sprite = sprites[Random.Range(0, sprites.Length)];

    this.transform.eulerAngles = new Vector3(0.0f, 0.0f,
        Random.value * 360.0f);
    this.transform.localScale = Vector3.one * this.size;
    _rigidbody.mass = this.size;
}
}

```

Далі нам знадобиться метод, який породжує астероїди – назвемо його *AsteroidSpawner()*. Його задача – час від часу створювати нові астероїди випадкового розміру у випадковому місці, які рухаються з випадковою швидкістю у випадковому напрямі. Для цього введемо необхідні змінні, зокрема для доступу до префабу астероїда, період генерування нових астероїдів (візьмемо 2 с) і кількість астероїдів, що створюються. Астероїди створюватимуться за допомогою методу *Start()*, який періодично викликатиметься за допомогою методу *InvokeRepeating()*. Цей метод класу *MonoBehaviour* викликатиме вказаний метод в заданий час, а потім повторно знов і знов – із вказаною періодичністю у секундах.

```

using UnityEngine;

public class AsteroidSpawner : MonoBehaviour
{
    public Asteroid asteroidPrefab;
    public float spawnDistance = 12f; // радіус зони породження астероїдів
    public float spawnRate = 1f;
    public int amountPerSpawn = 1;
    [Range(0f, 45f)]
    public float trajectoryVariance = 15f; // кут розходження траєкторій

    private void Start()
    {
        InvokeRepeating(nameof(Spawn), spawnRate, spawnRate);
    }

    public void Spawn()
    {
        for (int i = 0; i < amountPerSpawn; i++)
        {
            // Вибираємо випадковий напрямок від центру породжувача та
            // створюємо астероїд на деякій відстані від нього
            Vector2 spawnDirection = Random.insideUnitCircle.normalized;
            Vector3 spawnPoint = spawnDirection * spawnDistance;

```

						КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			52

```

// Зміщуємо точку породження астероїда відносно позиції
// породжувача
spawnPoint += transform.position;

// Розраховуємо випадкове відхилення траєкторії астероїда
float variance = Random.Range(-trajectoryVariance,
    trajectoryVariance);
Quaternion rotation = Quaternion.AngleAxis(variance,
    Vector3.forward);

// Створюємо новий астероїд, клонувавши префаб і задаючи
// випадковий розмір у межах певного діапазону
Asteroid asteroid = Instantiate(asteroidPrefab, spawnPoint,
    rotation);
asteroid.size = Random.Range(asteroid.minSize,
    asteroid.maxSize);

// Встановлюємо траєкторію руху у напрямку породжувача
Vector2 trajectory = rotation * -spawnDirection;
asteroid.SetTrajectory(trajectory);
    }
}
}

```

Команда *Random.insideUnitCircle.normalized* повертає випадкову точку на колі одиничного радіуса і задає, таким чином, напрям прямолінійної траєкторії астероїда.

Окрім методу створимо ігровий об'єкт *Spawner* і призначимо йому новостворений метод. Він буде розташований в центрі сцени і генерувати астероїди навколо себе в колі заданого радіусу – поза ігровою сценою. Оскільки границі екрану віддалені від центру максимум на 10 одиниць, задамо цей радіус рівним 15 одиницям. Після створення астероїди мають потрапляти в ігровий екран, отже, їх треба спрямовувати в куті, який покриває частину екрану – задамо цей кут рівним 30° (рис. 2.1).

Далі допишемо у скрипті *Asteroid.cs* метод *SetTrajectory()*, який штовхатиме наш астероїд у заданому напрямку:

```

public void SetTrajectory(Vector2 direction)
{
    // До астероїда потрібно додати силу лише один раз,
    // тому що у нього немає опору, щоб зупинити свій рух
    _rigidbody.AddForce(direction * movementSpeed);
}

```

									Арк.
									53
Змн.	Арк.	№ докум.	Підпис	Дата					

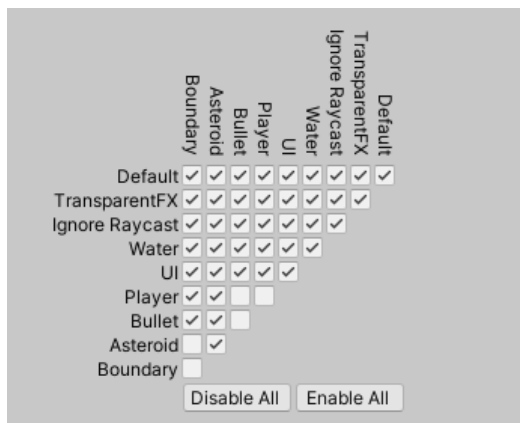


Рис. 3.21 — Додаткове налаштування матриці взаємодій шарів

3.5 Обробка зіткнення астероїдів з кулями

Наступний етап – обробка зіткнення астероїдів з кулями. Коли у великі астероїди потрапляє куля, вони повинні розпадатися на декілька маленьких. Маленькі астероїди при потраплянні кулі будуть повністю знищуватися.

Щоб ідентифікувати об’єкт кулі, коли вона стикається з астероїдом, надамо кулі тег *Bullet* в редакторі Unity Editor (рис. 3.22).

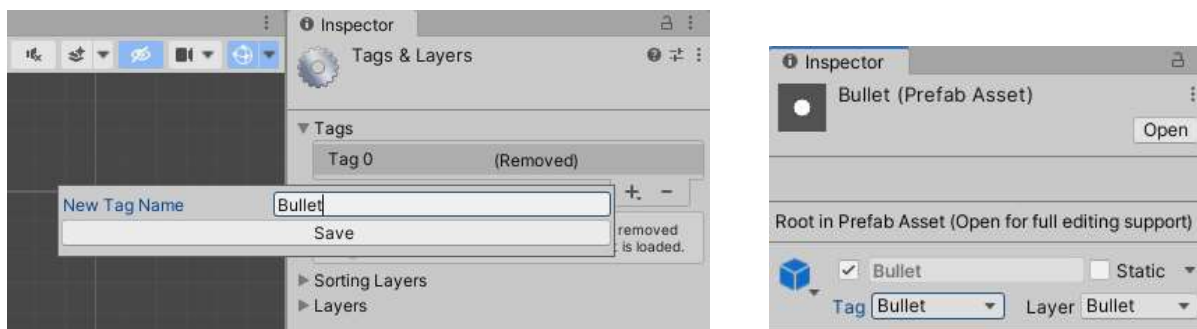


Рис. 3.22 — Створення тегу *Bullet* і надання його префабу *Bullet*

Створюємо метод *OnCollisionEnter2D()* для астероїда:

```
// Обробляємо зіткнення астероїда з кулею
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.tag == "Bullet")
    {
        if ((this.size * 0.5f) >= this.minSize)
        {
            // поділяємо астероїд на два менших
        }
    }
}
```

									Арк.
									55
Змн.	Арк.	№ докум.	Підпис	Дата					

```

        CreateSplit();
        CreateSplit();
    }
    Destroy(this.gameObject);
}
}

```

Реалізуємо тепер метод *CreateSplit()* для створення фрагменту астероїда, який утворюється при потраплянні кулі. Розмір фрагмента задаємо вдвічі менший за розмір «батьківського» астероїда, який розпадається. Положення фрагмента встановлюємо з невеликим зміщенням відносно позиції «батьківського» астероїда. Траєкторію поки що задаємо довільну (в подальшому доцільно було б урахувати при цьому траєкторію «батьківського» астероїда).

```

// Створення фрагменту астероїда
private void CreateSplit()
{
    Vector2 position = this.transform.position; // позиція фрагменту
    // нова позиція буде з невеликим зміщенням відносно позиції "батька"
    position += Random.insideUnitCircle * 0.5f;
    Asteroid half = Instantiate(this, position,
                               this.transform.rotation);
    // розмір фрагменту вдвічі менший за "батька"
    half.size = this.size * 0.5f;
    // траєкторія - довільна
    half.SetTrajectory(Random.insideUnitCircle.normalized);
}

```

3.6 Обробка зіткнення корабля з астероїдами

Коли корабель стикається з астероїдом, він має знищуватися. Для цього додаємо у скрипт *Player()* метод *OnCollisionEnter2D()*. Також додаємо тег «Asteroid» до префабу *Asteroid* по аналогії з тим, як ми додавали тег до кулі (рис. 3.22).

```

private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.tag == "Asteroid") {
        // зупиняємо рух корабля
        _rigidbody.velocity = Vector3.zero;
        _rigidbody.angularVelocity = 0.0f;
        this.gameObject.SetActive(false); // деактивуємо об'єкт
    }
}

```

						КРБ.КІ.1.440-03.3.12	Арк.
							56
Змн.	Арк.	№ докум.	Підпис	Дата			

Після знищення корабель має відродитися через декілька секунд, якщо у гравця ще залишились «життя». Для цього створюємо спеціальний скрипт *GameManager*, за допомогою якого можна відслідковувати загальний стан гри і ігрових об'єктів (рис. 3.23).



Рис. 3.23 — Сценарій *GameManager* в редакторі Unity

В цьому скрипті ми задамо змінні для кількості «життів» *lives* і для часу до відновлення корабля гравця *respawnTime*. Також визначаємо методи для обробки загибелі корабля гравця *PlayerDied()*, відновлення корабля *Respawn()* і для закінчення гри *GameOver()*.

```
public class GameManager : MonoBehaviour
{
    public Player player;
    public float respawnTime = 3.0f;
    public int lives = 3;

    public void PlayerDied()
    {
        this.lives--;
        if (this.lives <= 0) {
            GameOver();
        }
        else {
            Invoke(nameof(Respawn), this.respawnTime);
        }
    }
}
```

Коли корабель гравця знищується, ми віднімаємо 1 від кількості життів. Якщо кількість життів не нульова, ми викликаємо метод відновлення корабля *Respawn()*. В іншому випадку викликаємо метод закінчення гри *GameOver()*.

Метод *PlayerDied()* ми викликаємо у сценарії *Player*, в методі *OnCollisionEnter2D()*, при обробці зіткнення з астероїдом:

					КРБ.КІ.1.440-03.3.12	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

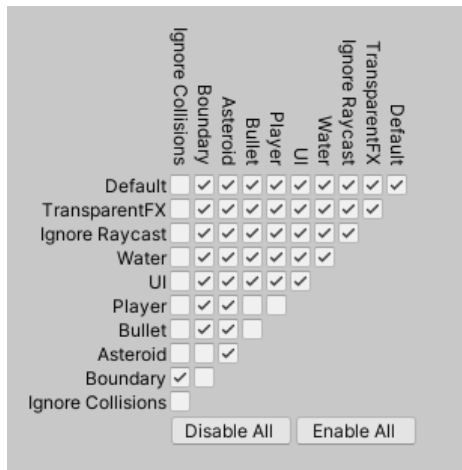


Рис. 3.25 — Налаштування шару *Ignore Collisions*

Активувати шар *Ignore Collisions* для об'єкту *Player* будемо програмним способом. У методі відродження корабля *Respawn()* активуємо шар «*Ignore Collisions*», а через 3 секунди стану невразливості повертаємо шар «*Player*»:

```
public float respawnInvulnerabilityTime = 3.0f;
. . .
private void Respawn()
{
    this.player.transform.position = Vector3.zero;
    this.player.gameObject.layer =
        LayerMask.NameToLayer("Ignore Collisions");
    this.player.gameObject.SetActive(true);
    Invoke(nameof(TurnOnCollisions), this.respawnInvulnerabilityTime);
}

private void TurnOnCollisions()
{
    this.player.gameObject.layer = LayerMask.NameToLayer("Player");
}
```

3.7 Реалізація ефекту вибуху при руйнуванні астероїда і корабля

Для того, щоб зробити гру дещо більш привабливою, додаємо невеликий ефект вибуху при зіткненні кулі або корабля з астероїдом. Використовуємо для цього вбудований в Unity ефект під назвою *Particle System*. Створюємо ігровий об'єкт даного типу, перейменовуємо його у «*Explosion*» і налаштовуємо властивості (рис. 3.26).

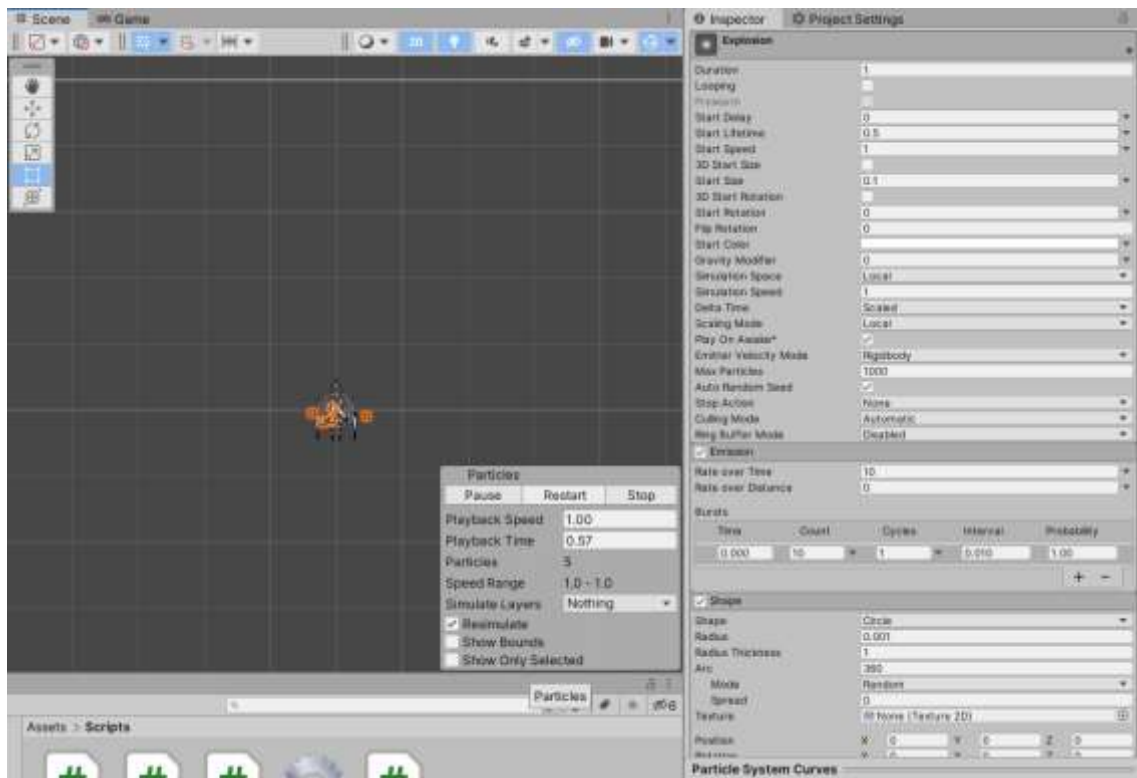


Рис. 3.26 — Налаштування ефекту *Explosion*

Тривалість ефекту задаємо 1 с, без зациклювання, час існування кожної частки – 0,5 с, швидкість руху частки – 1, розмір частки – 0,1. Також змінюємо параметри випромінювання часток і їх кількість (10). Форму області випромінювання задаємо кругову. Щоб частки випромінювалися з середини кола, задаємо параметр *Radius* = 0,001. Також відключаємо параметр «*Play on Awake*», щоб запускати цей ефект не одразу автоматично, а у потрібний момент програмно.

Крім того, налаштовуємо параметр «*Color over Lifetime*», який дозволить задати поступове зникнення часток, коли вони пролетять половину свого шляху (рис. 3.27).

Тепер треба запрограмувати відтворення ефекту частинок усякий раз, коли відбувається руйнування корабля або астероїда на сцені. Для цього, перш за все, треба мати посилання на ігровий об'єкт ефекту. Створюємо це посилання у нашому класі *GameManager*:

```
public ParticleSystem explosion;
```

										Арк.
										60
Змн.	Арк.	№ докум.	Підпис	Дата						

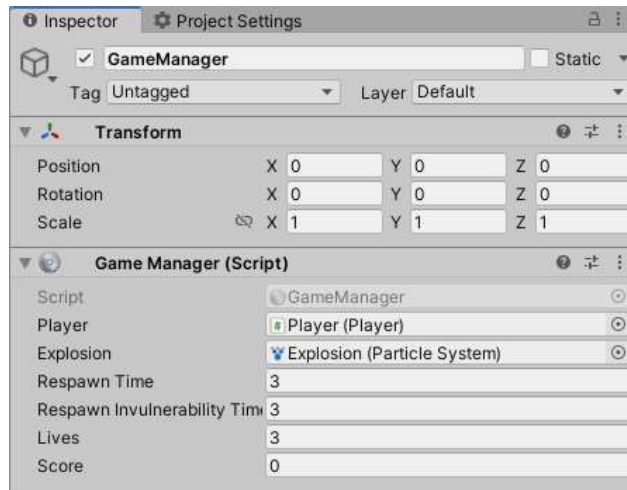


Рис. 3.28 — Прив'язка об'єкту вибуху до класу *GameManager*

3.8 Підрахунок очок гри

Реалізуємо тепер підрахунок очок гри. Будемо нараховувати очки при потраплянні кулі в астероїд. Оскільки в менші астероїди складніше влучити, будемо нараховувати за їх знищення більше очок, ніж за великі. Різниця між мінімальним та максимальним розмірами астероїда складає в нас $1: maxSize - minSize = 1,5 - 0,5 = 1$. Введемо три класи розмірів астероїдів: маленькі, середні та великі. Діапазон розмірів кожного класу складе $1 / 3 = 0,333$. Відповідно, маленькі астероїди будуть розмірами від 0,5 до $0,5 + 0,333 = 0,833$; середні астероїди – від 0,833 до $0,833 + 0,333 = 1,168$; великі астероїди – від 1,168 до 1,5. За влучення в маленький астероїд будемо додавати 100 очок, за середній – 50 очок і за великий – 25 очок.

Код для підрахунку очок додаємо в метод *AsteroidDestroyed()*:

```
public void AsteroidDestroyed(Asteroid asteroid)
{
    this.explosion.transform.position = asteroid.transform.position;
    this.explosion.Play();

    // збільшуємо рахунок
    if (asteroid.size < 0.834f) {
        this.score += 100;
    }
    else if (asteroid.size < 1.168f) {
        this.score += 50;
    }
}
```

					КРБ.КІ.1.440-03.3.12	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки третього розділу

В даному розділі була реалізована перша версія аркадної гри «Астероїди». Вона реалізує усі основні механіки гри. У наступних релізах отримана реалізація може бути дооснащена новими функціями, такими як:

- таблиця рекордів;
- випадкове переміщення в іншу позицію сцени;
- поява супротивників, які стрілятимуть по кораблю гравця, тощо.
- покращення графічного оформлення сцени, фону та інших елементів зовнішнього вигляду, створення для деяких з них додаткової анімації, можливо, розробка 3D-моделі сцени;
- більш достовірний рух частин після розколу астероїда;
- можливість багатьом гравцям діяти в одному ігровому просторі.

Також гра може бути оптимізована с точки зору продуктивності і використання обчислювальних ресурсів.

					КРБ.КІ.1.440-03.3.12	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4

ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЕКТУ

4.1 Організаційно-економічна характеристика роботи

Об'єктом розробки в даній дипломній роботі є програмний продукт – аркадна комп'ютерна гра для платформи Windows. Інструмент розробки – ігровий рушій і середовище розробки Unity.

Областю використання даного проекту є сфера розваг, зокрема – галузь відеоігор. За обігом коштів галузь відеоігор рік у рік випереджає музичну індустрію та кіновиробництво разом узяті. Загальна кількість геймерів у світі становить близько трьох мільярдів.

Основні принципи розробки відеоігор були сформовані між 1978 та 1984 роками. У 1978 році розробникам стало ясно, що існують певні причини, що роблять відеоігри особливими. Так, серія ігор Space Invaders від геймдизайнера Томохіро Нісікадо стала світовою сенсацією, долучивши до відеоігор ціле покоління людей, які раніше в них практично не грали.

Першою популярною відеоігрою, що використовувала мікротранзакції для покупки внутрішньоігрових предметів, стала аркадна гра «Double Dragon 3: The Rosetta Stone» (1990). Гравці вставляли монети в аркадні автомати, щоб купувати взагалі все: бонуси, здоров'я, зброю, спеціальні рухи та навіть персонажів. Ця бізнес-модель призвела до шаленого успіху аркадних автоматів. Інша гра, Space Invaders, принесла компанії Taito 2,7 млрд доларів до 1982-го року, а виторг Pacman від Namco склав більше 3,5 млрд доларів у 1990 році.

З часом люди хотіли грати не лише у спеціальних залах гральних автоматів, а й у себе вдома. Тому цей попит із лишком задовольнила поява перших консолей. Домінуючою домашньою приставкою з 1977-го року стала Atari VS S, що отримала пізню назву Atari 2600. Ігри в той час були

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

набагато простішими з погляду дизайну, ніж зараз. Однак вони дали старт активному розвитку ринку.

З розвитком Інтернету стали набувати популярності безкоштовні ігри, що розповсюджуються за моделлю free-to-play, тобто це ігри, за які не потрібно було платити, щоб в них пограти. Проте, саме в них стали реалізовуватися мікротранзакції для внутрішньо ігрових покупок. Такі ігри дозволяють користувачам, наприклад, налаштовувати одяг та стиль своїх персонажів, купувати та збирати меблі, а також набувати особливих, «яскравих» анімацій тощо. Модель free-to-play переважає в мобільному просторі, але також зустрічається в іграх у соціальних мережах, MMO та комп'ютерних іграх.

У грі Second Life є своя віртуальна валюта – липовий долар (L\$), яка використовується для забезпечення внутрішньої економіки гри. L\$ можна купити за реальні гроші на торговому майданчику LindeX, розробленому Linden Lab. У вересні 2005 року на цій платформі було оброблено транзакцій на суму 3 596 674 доларів США.

Індустрія інтерактивних розваг дає сьогодні своїм учасникам колосальний дохід. За весь час існування відеоігор змінювалася бізнес-модель їхнього поширення, впроваджувалися різні схеми придбання контенту, а деякі компанії розробили власні економічні системи. Бюджети під час існування промисловості зростають рік у рік; і хоч кількість гравців також збільшилася, видавці вже не хочуть отримувати менше грошей. Тому економічні системи в іграх точно розвиватимуться й надалі.

За типом комп'ютерна гра, яка розробляється в даному проекті, є казуальною. Тобто, вона не потребує спеціальних пристроїв для гри, має прості правила і короткий цикл проходження рівнів.

Існує дуже багато аналогів гри, що розробляється в даному проекті. Серед них, зокрема, браузерні ігри: Atari® Asteroids від AARP [5], Asteroids від Free Video Games Online (<https://freeasteroids.org/>), Asteroids від Novel Games (<https://www.novelgames.com/en/asteroids/>) та інші. Для Android:

									Арк.
									66
Змн.	Арк.	№ докум.	Підпис	Дата					

комп'ютерні ігри. Очікуваною конкурентною перевагою являється низька ціна продукту.

Для досягнення мети були проаналізовані сучасні тенденції у розробці програмних застосунків, досліджені засоби розробки програмних застосунків і деякі проекти конкурентів.

Передбачається поширення продукту за допомогою продажу його компанії замовнику, або самостійне розміщення на платформах – ігрових магазинах, таких як Steam, itch.io, Kongregate. Так як гра не відрізняється технологічністю та виконана переважно однією людиною за короткий строк, планується виставлення мінімально можливої ціни на усіх платформах. Це 0.5\$ для Steam, та 1.5\$ для itch.io. Kongregate працює за моделлю розділення прибутку від реклами. Реклама продукту не передбачена. За необхідності можливе залучення до роботи над проектом вільнонайманих працівників, з оплатою за правилами біржі заказів.

4.2 Розрахунок науково-технічної ефективності проекту

Під ефективністю розуміють здатність приносити ефект, результативність процесу, проекту, які визначаються як відношення ефекту, результату до витрат, що забезпечили цей результат.

Ефект в інноваційній діяльності – це кінцевий результат впроваджуваної діяльності в одному з економічних вимірах – матеріальному або грошовому, який може бути виражений через підвищення якості продукції; скорочення часу виробництва та обігу; вивільнення ресурсів.

Основними категоріями ефективності науково-дослідних та дослідно-конструкторських робіт (НДДКР) є економічний, науково-технічний, соціальний, маркетинговий та екологічний ефект.

Економічний ефект розробки та впровадження проекту оцінюється системою вартісних показників і проявляється у підвищенні економічної ефективності роботи підприємства, раціональному використанні ресурсів,

									Арк.
									68
Змн.	Арк.	№ докум.	Підпис	Дата					

КРБ.КІ.1.440-03.3.12

збільшенні прибутку від реалізації продукції, покращенні використання виробничих потужностей, впровадженні корисних винаходів.

Науково-технічний ефект полягає у прирості практично орієнтовних (прикладних) науково-технічних знань і умінь. На етапах практичного використання науково-технічний ефект інноваційної діяльності можна оцінити шляхом оцінки таких показників: підвищення науково-технічного рівня виробництва; підвищення організаційного рівня виробництва; кількість зареєстрованих охоронних документів; збільшення частки використання нових інформаційних технологій; збільшення частки нових технологічних процесів; зростання кількості і статусу науково-технічних публікацій; підвищення конкурентоспроможності підприємства та його продукції.

Соціальний ефект інноваційної діяльності, сприяючи розвитку суспільства та задовольняючи його потреби, проявляється насамперед у досягненні якісно нового рівня життя населення, підвищенні рівня освіти та створенні нових нематеріальних цінностей й оцінюється в основному якісними показниками. Крім позитивних впливів також треба враховувати потенційні негативні впливи тривалого використання техніки під час гри. Щоб їх знизити, треба обмежувати час, який користувач (особливо, якщо він дитина) витрачає на гру.

Екологічне оцінювання враховує вплив інновацій на розв'язання проблем охорони довкілля, що особливо важливо у реалізації інноваційних проектів, які можуть змінювати рівень екологічної безпеки території.

Маркетинговий ефект відображає потреби ринку в наукових дослідженнях, реалізації маркетингової стратегії, сукупності заходів, що проводяться, виражений через збільшення показників задоволеності споживачів і, як результат, поліпшення бізнес-показників компанії.

«Науково-технічну ефективність (НТЕ) результатів прикладних робіт визначаємо на основі показників науково-технічного рівня. Оцінка науково-технічної ефективності НДДКР відбувається на основі показника ($O_{НТЕ}$), який представляє собою ступінь досягнення максимально можливого рівня, значення якого дорівнює 1 (одиниці)» [1]:

					КРБ.КІ.1.440-03.3.12	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

$$O_{HTE} = K^{\Phi}_{HTE} / K^{\Pi}_{HTE}, \quad (4.1)$$

де K^{Φ}_{HTE} – показник (коефіцієнт) фактичного рівня науково-технічної ефективності;

K^{Π}_{HTE} – показник (коефіцієнт) потенційно можливого рівня науково-технічної ефективності (дорівнює одиниці).

Значення показника K^{Φ}_{HTE} визначаємо на основі шкали експертних оцінок (табл. 4.1) [1].

Таблиця 4.1

Шкала експертних оцінок для виміру рівня науково-технічної ефективності проектів

№	Групи показників	Характеристика показників	Інтервал рейтингового числа	Коефіцієнт значущості показників
1	Науково-технічний рівень	Перевищує кращі світові аналоги	10	0,30
		Відповідає світовому рівню	7 – 9	
		Нижче кращих світових аналогів	5 – 6	
		Перевищує кращі вітчизняні аналоги	3 – 4	
		Відповідає вітчизняному рівню	1 – 2	
		Нижче вітчизняного рівня	0	
2	Перспективність	Першочергова значущість	8 – 10	0,25
		Значущий	5 – 7	
		Корисний	1 – 4	
3	Потенційний масштаб практичного використання	Світовий ринок	10	0,20
		Галузі національної економіки	7 – 9	
		Галузь (регіон)	3 – 6	
		Окремі підприємства (об'єднання)	1 – 2	
4	Ступінь вірогідності досягнення позитивних результатів	Великий	10	0,25
		Середній	5 – 9	
		Малий	1 – 4	

(Таблиця узята з [1])

«Об'єкт оцінки і аналоги, які порівнюють за однаковими показниками, наведеними у співставленому вигляді відхилення в значеннях кожного з показників, мають бути однаковими для варіантів, що порівнюються». [1]

4.3 Проведення оцінки науково-технічного рівня розробки

Визначаємо K^{Φ}_{HTE} на основі експертної оцінки науково-технічного рівня розробки.

«З цією метою:

- розробляємо перелік специфічних показників, необхідних для виміру науково-технічного рівня розробки;
- формуємо групу аналогів, які реалізовані на світовому і вітчизняному ринках;
- здійснюємо відповідні розрахунки для співставлення показників і визначення балів по таблиці 4.1.

До числа специфічних показників відносять:

- для нової техніки: продуктивність, споживання інженерних ресурсів на виробітку одиниці продукції, потреба в робочих, які обслуговують обладнання, експлуатаційні витрати на одиницю продукції;
- для нових матеріалів і речовин: вміст корисних речовин для виробітки готової продукції, питома вага відходів у загальному обсязі переробленої сировини, вартість одиниці нового матеріалу, додаткові витрати на екологічну компенсацію;
- для нових технологій: якість виробленої продукції, енергоємність і трудомісткість продукції, собівартість одиниці продукції.» [1]

На основі співставлення даних таблиці 4.2 встановлюємо бали по характеристиках чотирьох груп і на цій основі розраховуємо значення інтегрального показника HTE [1]:

									Арк.
									71
Змн.	Арк.	№ докум.	Підпис	Дата					

КРБ.КІ.1.440-03.3.12

$$HTE = \sum B_i \cdot K_i^3, \quad (4.2)$$

де $i = 1 \div 4$,

B_i – бали (рейтингове число),

K – коефіцієнт значущості показників.

Таблиця 4.2

Порівняльні показники для виконання оцінки НТЕ

ПОКАЗНИКИ	Варіанти технології	
	Розробленої	співвідносної (аналога)
Рівень новизни	Високий	Середній
Якість продукції	Висока	Середня
Споживання на 1 версію продукту		
– тепла, Гкал	5	8
– електроенергії, кВт·годину	900	1200
– води, м ³	3	6
Трудомісткість виробництва, людино-годин	70	60

Рівень науково-технічної ефективності НДДКР розраховано на основі наведених даних прикладу (таблиця 4.3).

Таблиця 4.3

Експертна оцінка і розрахунок величини інтегрального показника НТЕ

№	Групи показників	Рейтинг експертів			Середня за експертними оцінками	НТЕ
		1	2	3		
1	Науково-технічний рівень	8	7	8	7,7	2,7 (7,7 · 0,35)
2	Перспективність	7	6	8	7	2,45 (7 · 0,35)
3	Потенційний масштаб практичного використання	7	8	7	7,3	1,46 (7,3 x 0,20)
4	Ступінь вирогідності досягнення позитивних результатів	9	8	9	8,66	0,87 (8,66 · 0,1)
Всього						7,48

$$HTE = 7,7 \cdot 0,35 + 7 \cdot 0,35 + 7,3 \cdot 0,20 + 8,66 \cdot 0,1 = 2,7 + 2,45 + 1,46 + 0,87 = 7,48.$$

Отриманий результат слід порівняти з максимально можливим значенням, яке дорівнює 10 балам ($10 \cdot 0,35 + 10 \cdot 0,35 + 10 \cdot 0,20 + 10 \cdot 0,1$).

Отже, оцінка рівня *HTE* може бути зроблена за допомогою інтегрального коефіцієнта оцінки *HTE* (K_{HTE}) [1]:

$$K_{HTE} = \frac{HTE}{10} \cdot 100 \% \quad (4.3)$$

На основі даних таблиці 4.3, можна дійти до висновку, що K_{HTE} відповідає 74,8 %, тобто:

$$\frac{7,48}{10} \cdot 100 = 74,8 \%$$

В тому випадку, коли значення K_{HTE} перевищує середнє значення, яке дорівнює 5,0, має бути зроблено висновок про достатній рівень НТЕ:

- цілком достатній 5,0 – 6,0;
- достатній 6,1 – 8,0;
- достатньо високий 8,1 – 9,0;
- високий 9,1 – 10.

Рівень НТЕ технології визначаємо достатнім. Отже, розроблений ПП пропонується випускати на ринок.

4.4 Розрахунок економічної ефективності проекту

Розрахунок економічної ефективності проекту – це процес визначення потенційного доходу та витрат проекту, а також оцінювання того, наскільки доцільно вкладати гроші у реалізацію проекту.

Для розрахунку економічної ефективності проекту необхідно визначити наступні показники:

					КРБ.КІ.1.440-03.3.12	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Вартість проекту: це загальна вартість всіх витрат, пов'язаних з реалізацією проекту, включаючи затрати на розробку, впровадження та експлуатацію.
2. Прибуток від проекту: це дохід, отриманий від реалізації продукту або послуги, яку реалізовує проект.
3. Термін окупності: це період, за який витрати на реалізацію проекту повернуться від прибутку.
4. Рентабельність інвестицій: це відношення чистого дисконтованого доходу до вартості інвестицій.

Для розрахунку економічної ефективності проекту, потрібно мати інформацію про витрати та прибуток проекту.

4.5 Розрахунок ціни продукту

Програмна продукція розробляється з урахуванням, що даний програмний продукт буде використовуватися сторонніми особами (користувачами). Отже ціна програмного продукту визначається за формулою:

$$Ц = K \cdot C + Пр; \quad (4.4)$$

де: C – витрати на розробку проекту (кошторисна собівартість);

K – коефіцієнт обліку витрат на виготовлення дослідного зразка P як продукції виробничо-технічного призначення ($K = 1.1$);

$Пр$ – нормативний прибуток, що розраховується за формулою:

$$Пр = (C - C_m) \cdot P_n / 100; \quad (4.5)$$

де: P_n – норматив рентабельності, (рекомендується 25%);

C_m – матеріальні витрати, грн.

Витрати на розробку проекту можуть бути представлені у вигляді кошторису витрат, що включає в себе наступні пункти:

									Арк.
									74
Змн.	Арк.	№ докум.	Підпис	Дата	КРБ.КІ.1.440-03.3.12				

- матеріали;
- спеціальне обладнання;
- основна заробітна плата;
- додаткова заробітна плата.

4.6 Витрати на матеріали

Розрахунки за даною статтею витрат можуть бути представлені у вигляді таблиці 4.4:

Таблиця 4.4

Витрати на матеріали

Найменування матеріальних витрат	Одиниця виміру	Кількість	Ціна за одиницю, грн.	Вартість, грн.
Папір для друку	Пачка	1	280	240
Фарба (картридж)	шт.	1	380	350
Накопичувач даних, 8 Gb	шт.	1	160	130
Разом				820
Транспортно-заготівельні витрати – 10 %				820 · 0,10 = = 82
Усього:				902

4.7 Розрахунок заробітної плати

В статтю включається основна заробітна платня виконавців, безпосередньо зайнятих розробкою даного П, з урахуванням його окладу, часу і ступеню участі в розробці. Розрахунок ведеться по формулі (4.6):

$$ЗП_{осн} = \sum Z_i \cdot K_0 \cdot \tau_i / D_p \quad (4.6)$$

де Z_i – середньомісячний оклад i -го виконавця, грн;

D_p – середня кількість робочих днів в місяці (22);

									Арк.
									75
Змн.	Арк.	№ докум.	Підпис	Дата					

КРБ.КІ.1.440-03.3.12

τ_i – трудомісткість робіт, виконуваних i -м виконавцем, дні (визначається з розрахунків мережного трафіка);

K_0 – ступінь участі керівника в проекті (0,1).

Приймаємо заробітну платню у розмірі 7000 грн/місяць та керівника у розмірі 10500 грн/місяць.

$$ЗП_{осн} = 10500 \cdot 0,1 / 22 \cdot 44 + 7000 / 22 \cdot 90 = 28636 + 2100 = 30736 \text{ грн.}$$

Додаткова заробітна плата враховує оплату чергових відпусток, премії, інші доплати. Приймається в розрахунок 10% від основної.

$$ЗП_{дод} = ЗП_{осн} \cdot 0,10. \quad (4.7)$$

$$ЗП_{дод} = 30736 \cdot 0,10 = 3074 \text{ грн.}$$

Єдиний соціальний внесок ($\epsilon_{св}$) приймається в розмірі 22% від суми основної і додаткової заробітної плати.

$$\epsilon_{св} = (ЗП_{осн} + ЗП_{дод}) \cdot 0,22. \quad (4.8)$$

$$\epsilon_{св} = (30736 + 3074) \cdot 0,22 = 7438 \text{ грн.}$$

«Витрати, пов'язані з використанням обчислювальної техніки, визначаються за формулою:

$$C_{еом} = T_{еом} \cdot K_{еом} \cdot Ц_{еом} \cdot K_{еом}; \quad (4.9)$$

де $T_{еом}$ – час використання ЕОМ для розробки даного ПП, години;

$$T_{еом} = 564 \text{ год};$$

$K_{еом}$ – поправочний коефіцієнт обліку часу використання ЕОМ (1,08);

$Ц_{еом}$ – ціна однієї години роботи на ЕОМ, грн, (5 грн.);

$K_{еом}$ – коефіцієнт обліку швидкодії ЕОМ (1,0 – швидкодія ЕОМ більш 20x10 опер./с; 1,2 – швидкодія ЕОМ менш 20x10 опер./с.)» [2]

$$C_{еом} = 564 \cdot 1,08 \cdot 5 \cdot 1,0 = 3045 \text{ грн.}$$

									Арк.
									76
Змн.	Арк.	№ докум.	Підпис	Дата					

КРБ.КІ.1.440-03.3.12

Накладні витрати враховують адміністративні, загальновиробничі витрати, витрати на збут. Приймаються в розмірі 50% від основної заробітної плати.

$$H_6 = 0,50 \cdot 3П_{осн} \quad (4.10)$$

$$H_6 = 0,50 \cdot 30736 = 15368 \text{ грн.}$$

На підставі здійснених розрахунків складається калькуляція планової собівартості ПП.

4.8 Калькуляція собівартості продукту

Таблиця 4.5

Витрати для розрахунку собівартості ПП

Найменування статей витрат	Сума витрат, грн.	Питома вага, %
1. Матеріали	902	1,5
2. Основна заробітна плата	30736	50,8
3. Додаткова заробітна плата	3074	5
4. Єдиний соціальний внесок	7438	12,3
5. Витрати, зв'язані з обчислювальною технікою	3045	5
6. Накладні витрати	15368	25,4
Разом:	60563	100

Таким чином, $C = 60563$ грн.

Прибуток у ціні ПП становить: $П = (60563 - 902) \cdot 0,25 = 14915$ грн.

Ціна ПП становить: $Ц = 60563 \cdot 1,1 + 14915 = 81534$ грн.

										Арк.
										77
Змн.	Арк.	№ докум.	Підпис	Дата						

КРБ.КІ.1.440-03.3.12

4.9 Розрахунок капітальних витрат

Для розрахунку економічної ефективності проекту визначають капітальні і поточні витрати, зв'язані з використанням програмного продукту.

«Розрахунок капітальних витрат, пов'язаних із впровадженням програмного продукту здійснюється по формулі:

$$K_2 = K_{nn} + K_n + K_{ко} + K_{во}, \quad (4.11)$$

де K_{nn} – ціна програмного продукту;

K_n – поперед виробничі витрати;

$K_{ко}$ – вартість комп'ютерного устаткування;

$K_{во}$ – вартість допоміжного устаткування, необхідного для надійної роботи продукту.» [2]

Ціна програмного продукту складає 81534 грн.

Попередвиробничі витрати містять у собі усі витрати, пов'язані з налагодженням і впровадженням гри – постановка задач та їхня алгоритмізація, розробка, налагодження і впровадження програмного продукту (ПП).

Приймаються K_n у розмірі 100% від вартості розробленого ПП:

$$K_n = 81534 \text{ грн.}$$

Вартість комп'ютера ($K_{ко}$) становить 30000 грн. Вартість допоміжного устаткування визначається укрупнено в розмірі 10% від вартості комп'ютера.

$$K_{во} = 30000 \cdot 0,10 = 3000 \text{ грн.}$$

$$K_2 = 81534 + 30000 + 3000 = 114534 \text{ грн.}$$

					КРБ.КІ.1.440-03.3.12	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

4.10 Розрахунок поточних експлуатаційних витрат

«Розрахунок поточних (експлуатаційних) витрат, пов'язаних з використанням ПП (C_i), здійснюється по формулі:

$$C_i = C_{опл} + C_a + C_{ел} + C_p + C_{доп} \quad (4.12)$$

де $C_{опл}$ – річний фонд основної і додаткової оплати праці персоналу, що обслуговує програмний продукт з нарахуваннями;

C_a – сума річних амортизаційних відрахувань від вартості основного і допоміжного устаткування програмного продукту;

$C_{ел}$ – вартість витрат на електроенергію на рік;

C_p – вартість річного ремонту основного і допоміжного устаткування;

$C_{доп}$ – річна вартість допоміжних матеріалів, зв'язаних з впровадженням ПП.» [2]

Весь процес контролю виконує один співробітник, заробітна плата якого складає 7800 грн. Розрахунок річного фонду основної і додаткової оплати праці персоналу з нарахуванням ($C_{опл}$).

Річний фонд основної заробітної плати персоналу, що обслуговує ПП:

$$ЗП_{осн} = (ЗП_{окі} \cdot Ч_i) \cdot 12, \quad (4.13)$$

де $Ч_i$ – чисельність, фахівців і-ї категорії, що обслуговують ПП;

$ЗП_{окі}$ – місячний оклад фахівця і-ї категорії;

$$ЗП_{осн} = (7800 \cdot 1) \cdot 12 = 93600 \text{ грн.}$$

Фонд додаткової заробітної плати;

$$З_{доп} = ЗП_{осн} \cdot K_{доп}, \quad (4.14)$$

					КРБ.КІ.1.440-03.3.12	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

де $K_{\text{дод}}$ – коефіцієнт додаткової заробітної плати, приймається в розмірі $K_{\text{дод}} = 0,2$.

$$З_{\text{дод}} = 93600 \cdot 0,2 = 18720 \text{ грн.}$$

Єдиний соціальний внесок приймається в розмірі 22% від суми основної і додаткової заробітної плати.

$$Є_{\text{св}} = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) \cdot 0,22,$$

$$Є_{\text{св}} = (93600 + 18720) \cdot 0,22 = 24710 \text{ грн.}$$

Загальні витрати на оплату праці;

$$С_{\text{опл}} = ЗП_{\text{осн}} + ЗП_{\text{дод}} + Є_{\text{св}} = 93600 + 18720 + 24710 = 137030 \text{ грн.}$$

Розрахунок амортизаційних відрахувань визначається по формулі;

$$C_a = (K_{\text{ко}} \cdot K_{\text{во}}) \cdot H_a / 100, \quad (4.15)$$

де H_a – норма амортизаційних відрахувань (для комп'ютерних устаткувань $H_a = 50\%$).

$$C_a = (30000 + 3000) \cdot 0,5 = 16500 \text{ грн.}$$

Річна вартість споживаної електроенергії $C_{\text{ел}}$, визначається по формулі:

$$C_{\text{ел}} = M_y \cdot T_{\text{ко}} \cdot Ц_e \cdot K_n, \quad (4.16)$$

де M_y – установлена сумарна потужність комп'ютерного устаткування, $M_y = 0,45$ кВт;

					КРБ.КІ.1.440-03.3.12	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

$T_{ко}$ – річний фонд часу роботи ЕОМ, який визначається виходячи з кількості робочих днів в році (D_p), тривалості робочого дня (T) і з урахуванням часу на профілактичні огляди за рік ($T_{огл}$);

$$T_{ко} = D_p \cdot T - T_{огл} ,$$

$$T_{ко} = 250 \cdot 8 - 300 = 1700 \text{ год.}$$

C_e – вартість 1 кВт-години електричної енергії (3,25 грн.);

K_n – коефіцієнт інтенсивного використання потужності ($K_n \text{ реком} = 0,9$).

$$C_{ел} = 0,45 \cdot 1700 \cdot 3,25 \cdot 0,9 = 2237 \text{ грн.}$$

Витрати на допоміжні матеріали ($C_{дон}$) приймаються в розмірі 2% від вартості комп'ютерного устаткування:

$$C_{дон} = 30000 \cdot 0,02 = 600 \text{ грн.}$$

Вартість річного ремонту основного й допоміжного устаткування (6% від $K_{ко}$);

$$C_p = 30000 \cdot 0,06 = 1800 \text{ грн.}$$

Загальні витрати:

$$C_2 = 137030 + 16500 + 2237 + 600 + 1800 = 158167 \text{ грн.}$$

4.11 Розрахунок показників економічної ефективності проекту

«Очікуваний економічний ефект визначається по формулі:

$$E_o = (C_1 - C_2) - E_n \cdot (K_2 - K_1), \quad (4.17)$$

					КРБ.КІ.1.440-03.3.12	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

де C_1, C_2 – поточні витрати відповідно до і після впровадження проекту;
 $(C_1 - C_2)$ – річна економія на поточних витратах, грн;
 K_2 – капітальні витрати на впровадження ПП, грн;
 K_1 – капітальні витрати до впровадження ПП, грн;
 E_n – нормативний коефіцієнт ефективності одноразових витрат
(рекомендовано $E_n = 0,25$).» [2]

Капітальні вкладення до впровадження застосунка були відсутні, тому $K_1 = 0$.

$$E_o = (158167 - 60563) - 0,25 \cdot (114534 - 0) = 97604 - 28633 = 68970 \text{ грн.}$$

Коефіцієнт ефективності капітальних витрат по формулі:

$$E = (C_1 - C_2) / K_2 - K_1, \quad (4.18)$$

$$E = 97604 / 114534 = 0,9.$$

Так як $E > E_n$, то проект ефективний.

Рентабельність проекту можна розрахувати за формулою:

$$R = \text{Прибуток} / \text{Витрати} \cdot 100\% \quad (4.19)$$

Припустимо, що прибуток складає 210000 грн.

$$R = 210000 / (158167 + 60563) \cdot 100\% = 210000 / 218730 \cdot 100\% = 96\%.$$

Розраховуємо строк окупності капітальних витрат на впровадження проекту:

$$T = 1 / R \cdot 100\% = 1 / 96 \cdot 100\% \approx 1 \text{ рік} = 12 \text{ міс.}$$

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		82

Результати економічних розрахунків відображаються в підсумковій таблиці 4.6.

Таблиця 4.6

Техніко-економічні показники програмного продукту

Найменування показників	Одиниця виміру	Значення показника	
		до впровадження проекту	після впровадження проекту
Трудомісткість розробки проекту	дні		86
Ціна ПП	грн		81534
Капітальні витрати	грн	–	114534
Поточні витрати	грн/рік	60563	158167
Економічний ефект від реалізації проекту	грн/рік		68970
Строк окупності	роки		1
Рентабельність	%		96

Висновки четвертого розділу

На основі зроблених економічних розрахунків можна дійти висновку, що впровадження даного продукту є досить вигідним, про що свідчить коефіцієнт економічної ефективності – 0,9. Строк окупності проекту – близько 12 місяців.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ НА РОБОЧОМУ МІСЦІ

5.1 Мікrokлімат виробничого середовища

Самопочуття та працездатність людини залежать від погодних умов у місці роботи. Такі фактори, як температура повітря (у градусах Цельсія), відносна вологість (у відсотках), швидкість руху повітря (у метрах за секунду), інтенсивність теплового випромінювання (у ваттах на квадратний метр або кілокалорій на квадратний метр на годину) і барометричний тиск (у міліметрах ртутного стовпа) називаються погодними умовами або мікrokліматом.

Вологість повітря має значний вплив на самопочуття та працездатність людини. Вона може бути абсолютною або відносною.

Абсолютна вологість визначається як кількість води (у грамах), яка міститься в одному кубічному метрі повітря при певній температурі (г/м³).

Відносна вологість визначається як відношення абсолютної кількості водяних парів у повітрі до їх максимально можливої кількості при даній температурі, виражене у відсотках.

На робочому місці мікrokліматичні показники часто впливають на людину комбіновано, взаємно посилюючи або послаблюючи один одного. Наприклад, збільшення швидкості повітря підсилює ефект низької температури і, навпаки, послаблює дію підвищеної температури на організм людини. Збільшення вологості погіршує самопочуття людини як при низькій, так і при високій температурі. Таким чином, комбінація метеорологічних параметрів у робочому середовищі може бути сприятливою або несприятливою для самопочуття людини.

									Арк.
									84
Змн.	Арк.	№ докум.	Підпис	Дата					

Здорова людина підтримує температуру свого організму на рівні 36,5-37 °С незалежно від погодних умов навколишнього середовища. Цей рівень підтримується завдяки підсвідомим механізмам терморегуляції. Терморегуляція відбувається через кілька шляхів: 30 % за рахунок конвекції (прямого нагрівання повітря шкірою людини), 45 % за рахунок випромінювання, 20 % через випаровування і 5 % через дихання.

Якщо температура навколишнього середовища піднімається до 25°C і вище, а відносна вологість становить більше ніж 75%, теплообмін між людиною та оточуючим середовищем порушується, що призводить до підвищення температури тіла. Терморегуляція в таких умовах відбувається переважно шляхом випаровування. При перегріві збільшується кровообіг до периферійних кровоносних судин. Розширення судин призводить до збільшення об'єму крові і тепловіддачі. При таких показниках людина втрачає від 5 до 8 літрів рідини, 50-80 г солей, що порушує водно-сольовий та вітамінний обмін в організмі. Внаслідок цього виникають слабкість, головний біль, шум у вухах і нудота. Дихання та пульс стають частішими, артеріальний тиск спочатку зростає, а потім падає. У важких випадках може настати тепловий удар, що вважається нещасним випадком. Також може виникнути судомна хвороба. Якщо людина втрачає 20 % води, настає смерть.

Робота в умовах низьких температур може призвести до переохолодження організму людини. Периферійні кровоносні судини стискаються, що призводить до зменшення кровотоку та тепловіддачі. Людина відчуває бажання активно рухатись, що сприяє обміну речовин в організмі та утворенню тепла. Якщо температура тіла знижується до 34,6°C, людина відчуває слабкість, а при температурах 25-26°C може настати смерть. Обмороження також вважається нещасним випадком.

Швидкість руху повітря має вплив на теплообмін між організмом людини та оточуючим середовищем. При високій температурі, збільшення швидкості руху повітря має позитивний вплив на організм. Однак, при

					КРБ.КІ.1.440-03.3.12	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

низькій температурі він має негативний ефект. Дуже низькі швидкості повітря, менші за 0,2 м/с, негативно впливають на самопочуття людини, особливо при виконанні одноманітної та монотонної роботи. Внаслідок цього людина швидко втомлюється та втрачає працездатність. Різкі перепади температур, як правило, часто спричиняють простудні захворювання.

Умови мікроклімату виробничого середовища регулюються нормативними документами, зокрема ДСН 3.3.6.042-99 "Санітарні норми мікроклімату виробничих приміщень". Ці нормативи встановлюють оптимальні та допустимі метеорологічні умови на робочому місці.

Допустимими параметрами мікроклімату називаються ті, які при тривалій та систематичній дії на людину можуть викликати тимчасові, але швидко нормалізуються зміни теплового стану організму, супроводжувані активністю механізмів терморегуляції, не виходячи за межі фізіологічних пристосувань. При цьому не виникає пошкоджень або порушень стану здоров'я, але можуть спостерігатися дискомфортні теплові відчуття, погіршення самопочуття та зниження працездатності.

Оптимальними параметрами мікроклімату є ті, які при тривалій та систематичній впливі на людину забезпечують збереження нормального теплового стану організму без напруження механізмів терморегуляції. Ці параметри створюють відчуття теплового комфорту та сприяють високому рівню працездатності людини.

Комфортність виробничого середовища визначається оптимальним поєднанням метеорологічних умов, що забезпечують комфорт та зручність для праці.

5.2 Пожежна профілактика

Будівлі і ті їх частини, в яких розміщуються ПК, повинні бути не нижче другого ступеня вогнестійкості. Приміщення для обслуговування, ремонту

					КРБ.КІ.1.440-03.3.12	Арк.
						86
Змн.	Арк.	№ докум.	Підпис	Дата		

та налагодження ПК повинні належати за пожежо-вибухонебезпечністю до категорії В відповідно до НАПБ Б.03.002-2007, а за класом приміщення – до П-2а за ПУЕ. Ми розглядаємо приміщення категорії Д.

Приміщення з ПК, крім приміщень, в яких розташовуються великі комп'ютери спеціального призначення (сервери, мейнфрейми), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до вимог Переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації, затвердженого наказом Міністерства внутрішніх справ України від 20.11.97 № 779 та зареєстрованого в Міністерстві юстиції України 28.11.97 за № 567/2371, та СНіП 2.04.09-84 «Пожежна автоматика будинків і споруд» з димовими пожежними оповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 кв.м площі приміщення з урахуванням гранично допустимих концентрацій вогнегасної рідини відповідно до вимог Правил пожежної безпеки в Україні. В інших приміщеннях допускається встановлювати теплові пожежні оповіщувачі. Підходи до засобів пожежогасіння повинні бути вільними.

Виконаємо розрахунок об'єму недоторканого запасу води для зовнішнього пожежогасіння за такими вхідними даними:

- будівля ступеня вогнестійкості I, категорія А;
- площа території $S = 250$ га;
- час гасіння пожежі $\tau = 3$ год;
- кількість одночасних пожеж для даного підприємства $n_{\text{пож}} = 2$;
- витрата води на зовнішнє пожежогасіння $g = 80$ л/с.

Витрата води на зовнішнє пожежогасіння протягом розрахункового часу:

$$V_B = \frac{k * g * \tau}{1000} * 3600 \text{ м}^3,$$

$$V_B = \frac{1,1 * 80 * 2 * 3}{1000} * 3600 \text{ м}^3$$

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		87

Визначимо необхідну кількість водойм, якщо об'єкт квадратної форми площею 250 га зі стороною: $L = \sqrt{250 \cdot 10^4} = 1581$ м та радіусом дії насосу $R = 200$ м:

$$n_{\text{вод}} = \frac{L}{2R} = \frac{1581}{2 \cdot 200} = 3,95 \text{ шт}$$

Приймаємо $n_{\text{вод}} = 4$ шт.

Розрахуємо кількість балонів з вуглекислотою для внутрішнього пожежогасіння приміщення об'ємом 576 м^3 .

Кількість вогнегасного газового складу:

$$G_{\Gamma} = 1,25 \cdot (G_{\text{в}} \cdot V_{\text{п}} \cdot K_{\text{у}}), \text{ кг}$$

де $G_{\text{в}}$ – вогнегасна концентрація газового складу для вуглекислоти ($G_{\text{в}} = 0,7 \text{ кг/м}^3$);

$V_{\text{п}}$ – об'єм приміщення, м^3 ;

$K_{\text{у}}$ – коефіцієнт, що враховує особливості процесу газообміну, витік вуглекислоти крізь нещільності приміщення, приймаємо 1,1.

$$G_{\Gamma} = 1,25 \cdot (0,7 \cdot 576 \cdot 1,1) = 554,4 \text{ кг}$$

Потрібна кількість робочих балонів $N_{\text{б}}$ (од.) із вуглекислотою визначається наступним чином:

$$N_{\text{б}} = \frac{G_{\Gamma}}{V_{\text{б}} \cdot \rho \cdot \alpha_{\text{н}}} \text{ од.}$$

де $V_{\text{б}} = 25$ л – об'єм балону (при 25 л у балоні міститься 15,6 кг вуглекислоти);

$\rho = 0,625 \text{ кг/л}$ – щільність засобу гасіння;

$\alpha_{\text{н}} = 1$ – коефіцієнт наповнення.

$$N_{\text{б}} = \frac{554,4}{25 \cdot 0,625 \cdot 1} = 35,48 \approx 36 \text{ од.}$$

					КРБ.КІ.1.440-03.3.12	Арк.
						88
Змн.	Арк.	№ докум.	Підпис	Дата		

Кількість резервних балонів приймаємо такою ж, як і робочих:

$$N_{\bar{6}} = N_p = 36 \text{ од.}$$

$$N_{\text{заг.}} = 36 + 36 = 72 \text{ од.}$$

Висновки п'ятого розділу

У даній частині дипломного проекту були розглянуті питання щодо гігієнічних норм організації і обладнання робочих місць користувачів персональних комп'ютерів, питання пожежної профілактики приміщень, виконані необхідні розрахунки об'єму недоторканого запасу води для зовнішнього пожежогасіння. Отримані результати дозволяють обладнати приміщення для роботи користувачів обчислювальної техніки з дотриманням вимог безпеки їх праці.

					<i>КРБ.КІ.1.440-03.3.12</i>	Арк.
						89
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ЗАГАЛЬНІ ВИСНОВКИ

Під час цієї роботи було проведено аналіз аркадних відеоігор загалом, включаючи принципи та засоби їх створення. Аналіз жанру дозволив детально приступити до розробки концепції гри "Астероїди", а також його проектування та реалізації.

Узагальнення постановки завдання визначило вимоги до проекту гри як з боку розробника, так і з боку користувача. На основі цього завдання був створений проект гри, який послужив основою для технічної реалізації в програмному продукті "Астероїди" для платформи Windows. Засновуючись на функціональних вимогах, були визначені зовнішній вигляд ігрового поля, параметри фізики, алгоритми геймплею, а також вибрані інструменти реалізації. В результаті цієї роботи було створено комп'ютерну гру, яка отримала програмне втілення. Загальна архітектура програмного проекту базується на концепціях побудови проектів у середовищі Unity.

Крім того, було проведено техніко-економічне обґрунтування проекту а також досліджені суміжні питання охорони праці.

Незважаючи на те, що перший рівень реалізації гри ще не дозволяє її випускати в загальний доступ, закладені ідеї та механізми гри закладають потенціал для її майбутньої популярності. Основні перспективи розвитку гри в конкурентному середовищі з аналогічними іграми включають:

- покращення графічного оформлення сцени, фону та інших елементів зовнішнього вигляду, створення для деяких з них додаткової анімації, можливо, розробка 3D-моделі сцени;
- більш достовірний рух частин після розколу астероїда;
- реалізація рейтингу;
- у майбутньому можна зробити гру мережевою і багатокористувацькою, додавши можливість багатьом гравцям діяти в одному ігровому просторі.

									Арк.
									90
Змн.	Арк.	№ докум.	Підпис	Дата					

КРБ.КІ.1.440-03.3.12

11. De Byl P. Holistic Game Development with Unity: An All-in-One Guide to Implementing Game Mechanics, Art, Design and Programming. – 2nd Edition. – A K Peters/CRC Press, 2017. – 456 p.
12. Halpern Jared. Developing 2D Games with Unity. – Apress, 2019. – 380 p.
13. Hocking J. Unity in Action: Multiplatform Game Development in C# with Unity 5. – Manning Publications, 2015. – 352 p.
14. Jackson S. Mastering Unity 2D Game Development. – Packt Publishing, 2014. – 474 p.
15. Lukosek Greg. Learning C# by Developing Games with Unity 5.x. – 2nd Edition. – Packt Publishing, 2016. – 230 p.
16. Mlot, Stephanie. MoMA Adds Seven Video Games to Art Collection [Електронний ресурс] // PC Magazine. June 28, 2013. — Режим доступу: <https://www.pcmag.com/article2/0,2817,2421168,00.asp>.
17. Moakley B., Berg M., Duffy S., Van de Kerckhove E., Uccello F. Unity Games by Tutorials: Make 4 Complete Unity Games from Scratch Using C#. – Razeware LLC, 2016. – 634 p.
18. Norton T. Learning C# by Developing Games with Unity 3D: Beginner's Guide. – Packt Publishing, 2013. – 292 p.
19. Okita A. Learning C# Programming with Unity 3D. – CRC Press, 2014. – 690 p.
20. Unity Manual // Unity Documentation [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/index.html>.
21. Welcome To Gaming Lite // Computer Gaming World. – September, 1992. – p. 74.
22. Wells R. Unity 2020 By Example. – Packt Publishing, 2020. – 677 p.

					КРБ.КІ.1.440-03.3.12	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		92