

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

**Завідувач кафедри**

\_\_\_\_\_ Сергій СТРЕНКО

«\_\_\_\_\_» \_\_\_\_\_ 2021 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Комп'ютерні системи та мережі»  
спеціальності 123 «Комп'ютерна інженерія»**

на тему: Комп'ютерна 3D гра з саунд-продюсуванням

Виконав: студент 4 курсу, групи ІВ-73

Люндовський Артем Олексійович \_\_\_\_\_

Керівник:

асистент Кочура Юрій Петрович \_\_\_\_\_

Консультант (нормконтроль):

Професор, доктор технічних наук

Сімоненко Валерій Павлович \_\_\_\_\_

Рецензент \_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

«Комп’ютерні системи та мережі»

Спеціальність 123 «Комп’ютерна інженерія»

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри**  
Сергій СТИРЕНКО

«\_\_» \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ**

на бакалаврський дипломний проєкт студента

Людковського Артема Олексійовича

1. Тема проєкту *Комп’ютерна 3D гра з саунд-продюсуванням*  
керівник проєкту *Кочура Юрій Петрович, асистент*  
Затверджені наказом по університету від «11» травня 2021 року №1139-с
2. Термін здачі студентом закінченого проєкту (роботи) \_\_\_\_\_
3. Вихідні дані до проєкту: технічне завдання, науково-технічна література.
4. Зміст розрахунково-пояснювальної записки: опис предметної області, розгляд існуючих рішень, реалізація програмного продукту.
5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо): \_\_\_\_\_
6. Консультанти проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормконтроль	д.т.н., проф. Сімоненко В. П		

7. Дата видачі завдання \_\_\_\_\_

### Календарний план

№ п/п	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	<i>22.01</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>17.02</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>02.03</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>10.04</i>	
5.	<i>Програмна реалізація системи</i>	<i>05.04</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>28.05</i>	
7.	<i>Захист програмного продукту</i>		
8.	<i>Передзахист</i>		
9.	<i>Захист</i>		

Студент

\_\_\_\_\_

Люндовський А.О.

Керівник проєкту

\_\_\_\_\_

Кочура Ю.П.

### **Анотація**

Дана дипломна робота присвячена створенню комп'ютерної 3D гри, а саме головоломки, на базі рушія Unity та реалізації музикального проєкту на платформі Ableton Live, що призначена для проведення дозвілля та розвитку навичок. Гравцю потрібно пройти від точки А до точки В, пересуваючи перешкоди.

Для створення 3D гри використовується ігровий рушій Unity та програмне середовище – Microsoft Visual Studio, для реалізації музикального проєкту – Ableton Live.

### **Аннотация**

Данная дипломная работа посвящена созданию компьютерной 3D игры, а именно головоломки, на базе двигателя Unity и реализация музикального проекта на платформе Ableton Live, которая предназначена для проведения досуга и развития навыков. Игроку нужно пройти от точки А к точке В, передвигая препятствия.

Для создания 3D-игры используется игровой движок Unity и программная среда - Microsoft Visual Studio, для реализации музикального проекта - Ableton Live.

### **Annotation**

This thesis is devoted to the creation of a computer 3D game, namely a puzzle, based on the Unity engine and the implementation of a music project on the platform Ableton Live, which is created for leisure and skills development. The player needs to go from point A to point B, moving obstacles.

To create a 3D game was used Unity game engine and software environment - Microsoft Visual Studio, to implement a music project - Ableton Live.

# **Опис альбому**

## **до дипломного проєкту**

на тему: «Комп'ютерна 3D гра з саунд-продюсуванням»



# **Технічне завдання до дипломного проєкту**

на тему: «Комп'ютерна 3D гра з саунд-продюсуванням»

## ЗМІСТ

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ.....	2
2. ПІДСТАВИ РОЗРОБКИ.....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ .....	2
4. ДЖЕРЕЛА РОЗРОБКИ .....	2
5. ТЕХНІЧНІ ВИМОГИ .....	3
5.1 Вимоги до розробленого продукту .....	3
5.2 Вимоги до програмного забезпечення.....	3
6. ЕТАПИ РОЗРОБКИ.....	3

					<b><i>ІАЛЦ.467800.002 ТЗ</i></b>									
<b><i>Зм</i></b>	<b><i>Арк.</i></b>	<b><i>№ докум</i></b>	<b><i>Підпис</i></b>	<b><i>Дата</i></b>										
<b><i>Розроб.</i></b>		<i>Людковський А.О</i>			<b><i>Комп'ютерна 3D гра з саунд-продюсуванням Технічне завдання</i></b>									
<b><i>Первір.</i></b>		<i>Кочура Ю.П</i>								<b><i>Літ.</i></b>	<b><i>Арк.</i></b>	<b><i>Аркушіє</i></b>		
										1	3			
<b><i>Н.Контр.</i></b>		<i>Сімоненко В.П</i>								<b><i>КПІ ім. Ігоря Сікорського ФІОТ, ІВ-73</i></b>				
<b><i>Затверд.</i></b>														

## **1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ**

Найменування: комп'ютерна 3D гра з саунд-продюсуванням.

Область застосування: повсякденне життя, відпочинок.

## **2. ПІДСТАВИ РОЗРОБКИ**

Підставою для розробки є завдання на виконання роботи кваліфікаційно-освітнього рівня «бакалавр», затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

## **3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ**

Метою даного проєкту є створення комп'ютерної 3D гри з саунд-продюсуванням (головоломки), для розвитку інтелектуальних навичок та дозвілля.

## **4. ДЖЕРЕЛА РОЗРОБКИ**

Джерелом розробки є наукова література з інформаційних технологій, технічна документація програм, які використовуються під час розробки проєкту, статті в мережі Інтернет з питань програмних середовищ та мов програмування.

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

***ІАЛЦ.467800.002 ТЗ***

**Арк.**

**2**

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1 Вимоги до розробленого продукту

- Зручність і простота інтерфейсу додатку.
- Технічна коректність.
- Присутність музикального супровіду.

### 5.2 Вимоги до програмного забезпечення

- Windows/Linux/macOS.

## 6. ЕТАПИ РОЗРОБКИ

	Дата
Вивчення літератури	10.02.2021
Складання і узгодження технічного завдання	12.02.2021
Аналіз структури програмного забезпечення	04.03.2021
Створення модулів розроблюваної системи	05.04.2021
Тестування окремих модулів системи	05.05.2021
Доопрацювання, налагодження і виправлення помилок	28.05.2021
Оформлення документації дипломної роботи	xx.xx.2021

					<b>ІАЛЦ.467800.002 ТЗ</b>	Арк.
						3
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		

**Пояснювальна записка**  
**до дипломного проєкту**

на тему: «Комп'ютерна 3D гра з саунд-продюсуванням»

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1.....	4
1.1 Розвиток ігрової індустрії.....	4
1.1.1 До 1980го року.....	4
1.1.2 1980-і роки.....	11
1.1.3 Початок 21 століття.....	13
1.2 Різниця між 2D та 3D іграми.....	16
1.3 Аналіз сучасних існуючих варіантів.....	17
1.3.1 Grand Theft Auto 5.....	18
1.3.2 Minecraft.....	18
1.3.3 Mafia 2.....	19
1.3.4 Call of duty.....	20
1.3.5 Assasin's creed.....	21
1.4 Категоризація ігор.....	22
1.4.1 3D-шутери .....	23
1.4.2 Файтинг.....	24
1.4.3 Пісочниця або SandBox.....	25
1.4.4 Симулятори .....	25
1.4.5 Стратегії.....	26
1.4.6 Спортивні.....	27
ВИСНОВКИ ДО РОЗДІЛУ 1.....	29
РОЗДІЛ 2. ОГЛЯД СЕРЕДОВИЩ РОЗРОБКИ.....	30
2.2 Існуючі ігрові платформи.....	30

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>							
<b>Зм</b>	<b>Арк.</b>	<b>№ докум</b>	<b>Підпис</b>	<b>Дата</b>								
<b>Розроб.</b>		<i>Людковський А.О</i>			<b>Комп'ютерна 3D гра з саунд-продюсуванням</b> <b>Пояснювальна записка</b>			<b>Літ.</b>	<b>Арк.</b>	<b>Аркушів</b>		
<b>Первір.</b>		<i>Кочура Ю.П.</i>								1	83	
<b>Н.Контр.</b>		<i>Сімоненко В.П</i>						<b>КПІ ім. Ігоря Сікорського</b> <b>ФІОТ, ІВ-73</b>				
<b>Затверд.</b>												

2.2.1 Мобільні пристрої.....	30
2.2.2 ПК.....	31
2.2.3 Консолі.....	32
2.2.4 Web.....	32
2.2.5 VR.....	33
2.3 Огляд середовищ розробки.....	33
2.3.1 Unity.....	33
2.3.2 Unreal Engine.....	36
2.3.3 CryEngine.....	37
2.3.4 Lumberyard.....	38
2.3.5 Інші двигуни.....	39
2.3 Аналіз платформ музикального продюсування.....	40
2.3.1 Steinberg Cubase.....	40
2.3.2 Ableton Live.....	41
2.3.3 FL Studio (FruityLoops).....	42
ВИСНОВКИ ДО РОЗДІЛУ 2.....	44
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ.....	45
3.1 Вибір технологій. Вибір ігрового рушія.....	45
3.2 Створення гри.....	51
ВИСНОВКИ ДО РОЗДІЛУ 3.....	63
РОЗДІЛ 4.....	64
4.1 Розробка саунд-дизайну.....	64
4.2 Фінальний вигляд гри.....	71
ВИСНОВКИ ДО РОЗДІЛУ 4.....	77
ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	79

					<b>ІАЛЦ.467800.003 ПЗ</b>		
<b>Зм</b>	<b>Арк.</b>	<b>№ докум</b>	<b>Підпис</b>	<b>Дата</b>			
<b>Розроб.</b>		Людковський А.О			<b>Літ.</b>	<b>Арк.</b>	<b>Аркушіє</b>
<b>Первір.</b>		Кочура Ю.П.				2	83
<b>Н.Контр.</b>		Сімоненко В.П			<b>Комп'ютерна 3D гра з саунд-продюсуванням</b> <b>Пояснювальна записка</b> <b>КПІ ім. Ігоря Сікорського</b> <b>ФІОТ, ІВ-73</b>		
<b>Затверд.</b>							

## ВСТУП

Ми живемо в час технологій, в той час, коли маючи у себе вдома невелику коробку, названу комп'ютером, або маючи телефон у своїй кишені, ми можемо розвиватися, вивчати будь-яку літературу, яка нам може знадобитися, реалізовувати свої ідеї, навчатися, заробляти гроші, дивитися фільми, а також грати в ігри; і це все не виходячи з дому.

Протягом історії людства ігри були способом розважати себе та оточуючих. Взагалі ігри мають більш серйозне значення еволюції людини, оскільки вони надають нам природний спосіб розвиватися та вчитися.

Розвиток комп'ютерних технологій за останнє десятиліття було значним. Сучасні комп'ютери дозволяють створювати все досконаліші та складніші віртуальні середовища. Крім того, загальне ставлення до комп'ютерних технологій різко змінилося, і сьогодні комп'ютери (всіх видів) стали нормальною частиною повсякденного життя. Ці зміни також відбилися в індустрії цифрових ігор, зробивши цифрові ігри ще більш популярними та зробивши їх більш прийнятими.

Мета даного дипломного проєкту — ознайомитися з процесом створення відеоігор. Комп'ютерна тривимірна гра – гра, візуальний простір, якої є повністю побудований з тривимірних, тобто 3d об'єктів. Персонаж, або головний елемент, знаходиться в тривимірному просторі та в деяких іграх має повну свободу пересування.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	Арк.
						3
<b><i>Зм.</i></b>	<b><i>Арк.</i></b>	<b><i>№ докум.</i></b>	<b><i>Підпис</i></b>	<b><i>Дата</i></b>		

## РОЗДІЛ 1

Завдання дипломної роботи полягає у розробці комп'ютерної 3D гри з саунд-продюсуванням, яка в свою чергу надасть своїм користувачам можливість грати в гру, слухаючи саундтрек до цієї гри.

Для виконання задачі потрібно використовувати платформи для написання гри та платформу для створення саундтреку до гри, цьому посвячений другий розділ.

Для початку потрібно розібратися що таке комп'ютерна гра? Цифрова комп'ютерна гра взагалі розглядається як певний набір дій, які виконуються двома та/або більше сторонами. Де перша сторона завжди є гравцем (людиною, яка грає), а друга сторона, або сторони гри можуть складатися з інших гравців (людей), чи з комп'ютера, або і з гравців і з комп'ютера. Існують правила, завдяки яким регулюються ці взаємодії в таких комп'ютерних іграх і у них є мета до якої вони прагнуть. Потрібно відзначити, що учасники гри можуть і не знати правила гри заздалегідь з самого початку, оскільки правила гравець може дізнатися, граючи в нову гру.

Наразі існує безліч комп'ютерних 3D ігор, але всього 100 років назад ніхто не міг і подумати, що люди будуть розважати себе та оточення цифровими іграми.

### 1.1 Розвиток ігрової індустрії

#### 1.1.1 До 1980го року

Коли розмова йде про комп'ютерні ігри, значна більшість осіб починає думати, що початок цифрових ігор зароджується з кінця 1970-х або початку 1980-х років. Авжеж, частково це правда, тому що за той період цифрові ігри

					<b>ІАЛЦ.467800.003 ПЗ</b>	<b>Арк.</b>
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		4

набули чималої популярності та стали відомі великій кількості людей, вони стали настільки відомими що комп'ютерні ігри почали відносити до домашніх розваг. Але якщо подивитися назад, то ми дізнаємося, що історія комп'ютерних ігор насправді починає зароджуватися в 1950-і і 1960-і роки, аж до самих перших днів існування комп'ютерів. Автором ідеї відеоігор вважають американського винахідника Ральфа Бера, зображеного на рисунку 1.1, який народився в 1922 році в Німеччині [1].



Рис.1.1 – Ральф Бер [1]

Проста, здавалася б на сьогодні, ідея перетворити звичайний телевізор в масове пристосування для розваг з'явилася у Бера ще в 1951 році, але така ідея не змогла знайти підтримки у його керівництва. І лише через 15 років, в 1966-му, будучи сам керівником великого відділу, Бер вперше представив схему ігрового пристрою на двох гравців, де вони переміщали курсори, у вигляді цвітних квадратів на моніторі, в вертикальному і горизонтальному напрямку. Квадрат одного гравця повинен був зловити квадрат іншого гравця.

Історія комп'ютерних відеоігор зароджується у далеких 47 роках того століття, в той час, як було реалізовано першу ігрову програму.

Розвиток програмних комп'ютерів для військових планів і став причиною

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

***ІАЛЦ.467800.003 ПЗ***

*Арк.*

5

виникнення перших цифрових ігор. Також їх виникненню, як розважальному типу дуже посприяв процес росту телебачення. Авжеж ТБ (телебачення) існувало задовго до світової війни, але загострення змусили відкласти вироблення телевізорів на користь матеріалів для обладнання збройних сил. У 40му році на виставці, яка проходила в Нью-Йорку вперше показано Nimatron — комп'ютерну ігрову машину, яку зображено на рисунку 1.2. Едвард Кондон — ім'я американського фізика, який розробив цей винахід. Незважаючи, що вплив цієї машини на майбутній розвиток відеоігор і цифрових комп'ютерів не вважають значним, це був найперший комп'ютер, який призначався для розваг, і являвся претендентом на статус першої комп'ютерної гри.

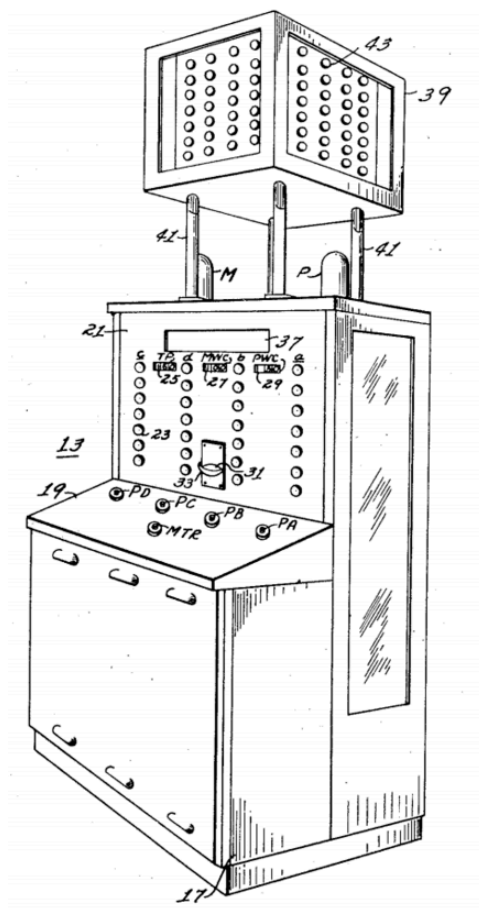


Рис 1.2 – Nimatron [3]

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

6

У 1950-х роках МІТ (Массачусетський технологічний інститут) представив перші курси комп'ютерних досліджень. У той же час вони також заснували дослідний відділ штучного інтелекту. Спочатку відділ фінансувався в основному військовими, але з часом вони також встановили тісні відносини з Digital Equipment Corporation (DEC), яка працювала на новому ринку міні-комп'ютерів. DEC підтримала департамент, надавши Массачусетському технологічному інституту міні-комп'ютери, навчивши ними користуватися. Натомість в якості оплати МІТ надав DEC дослідження, консультації та безкоштовні програми. Комп'ютерні ігри тривалий час не були популярні, і тільки в 1970-х та 1980-х роках, коли з'явилися для широкої публіки аркадні автомати, ігрові консолі та домашні комп'ютери, комп'ютерні ігри стають частиною попкультури. У 1970-х і початку 1980-х років світ став свідком справжнього зльоту цифрових ігор у вигляді аркад і домашніх відеоігор. Якщо більш ранні ігри були побічними продуктами комп'ютерних технологій і призначалися не для розваги, то аркадні і домашні відеоігри були призначені виключно для розважальних цілей. Аркадні ігри запускалися на ігрових автоматах з монетоприймачами і вони розташовувалися в барах, магазинах і парках розваг по всій Америці. Аркадні ігри існують і сьогодні, але їх популярність значно впала в порівнянні з їх золотим століттям у 80-і роки.

Нолана Бушнелла можна вважати хрещеним батьком аркадних ігор. Разом зі своєю командою він відповідає за розробку і виробництво однієї з перших аркадних ігор - Computer Space, яка насправді була копією Spacemar, гру зображено на рисунку 1.3.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	<b>Арк.</b>
						7
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		



Рис 1.3 – Spacewar [4]

Computer Space не мала такого великого успіху, як передбачалося, але пізніше Бушнелл і його команда створили Pong, зображену на рисунку 1.4, яка стала дуже популярною.

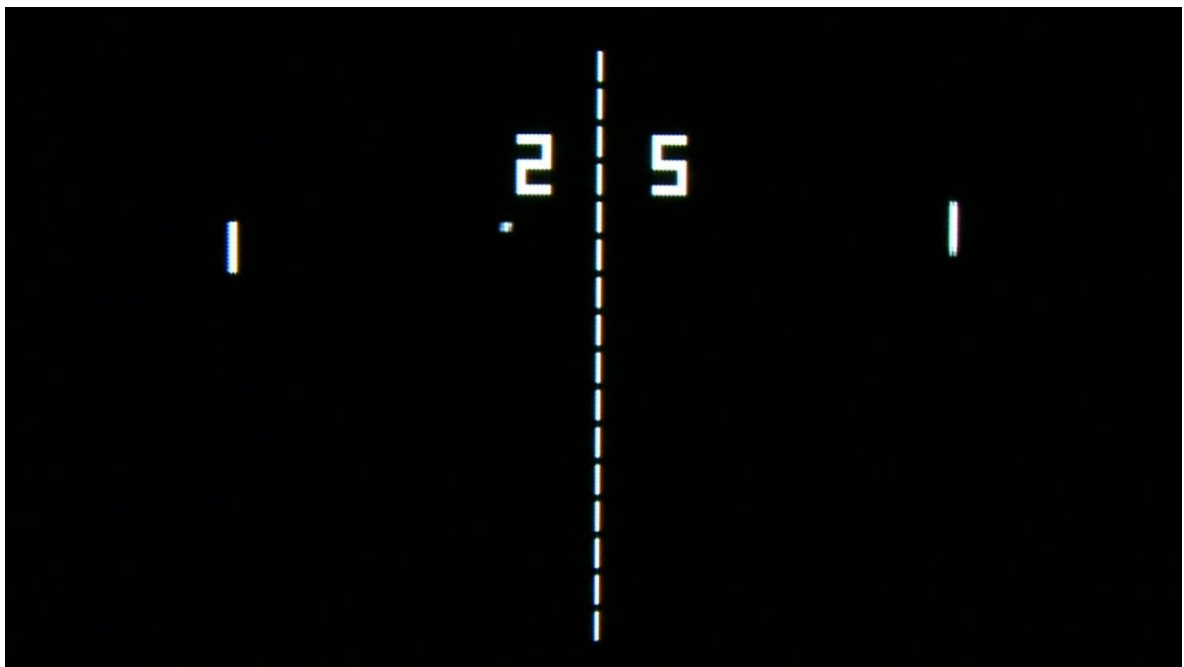


Рис 1.4 – Pong [5]

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

У 1979 році аркадні ігри досягли великого успіху у вигляді Space Invaders (рис. 1.5).

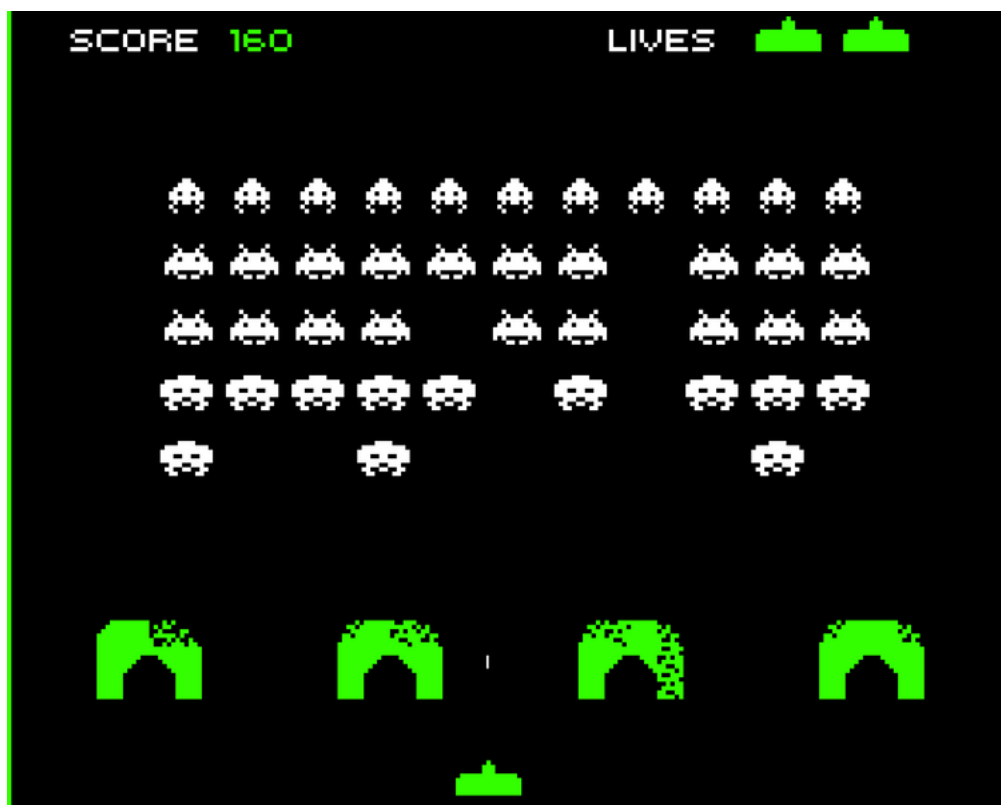


Рис 1.5 – Space Invaders [6]

Space Invaders підняли популярність аркадних ігор до нових розмірів. У період з 1979 по 1981 рік продажі автоматів для аркадних ігор зросли з 40 до 500 мільйонів доларів. І що ще більш важливо, ринки ігрових автоматів вийшли за межі США і стали міжнародними.

Приблизно в той же час з аркадними іграми на ринок вийшли домашні відеоігри. У ці ігри можна було грати вдома за допомогою телевізора. Однією з перших домашніх відеоігор була Odyssey (рис 1.6), яка була випущена в 1972 році.

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

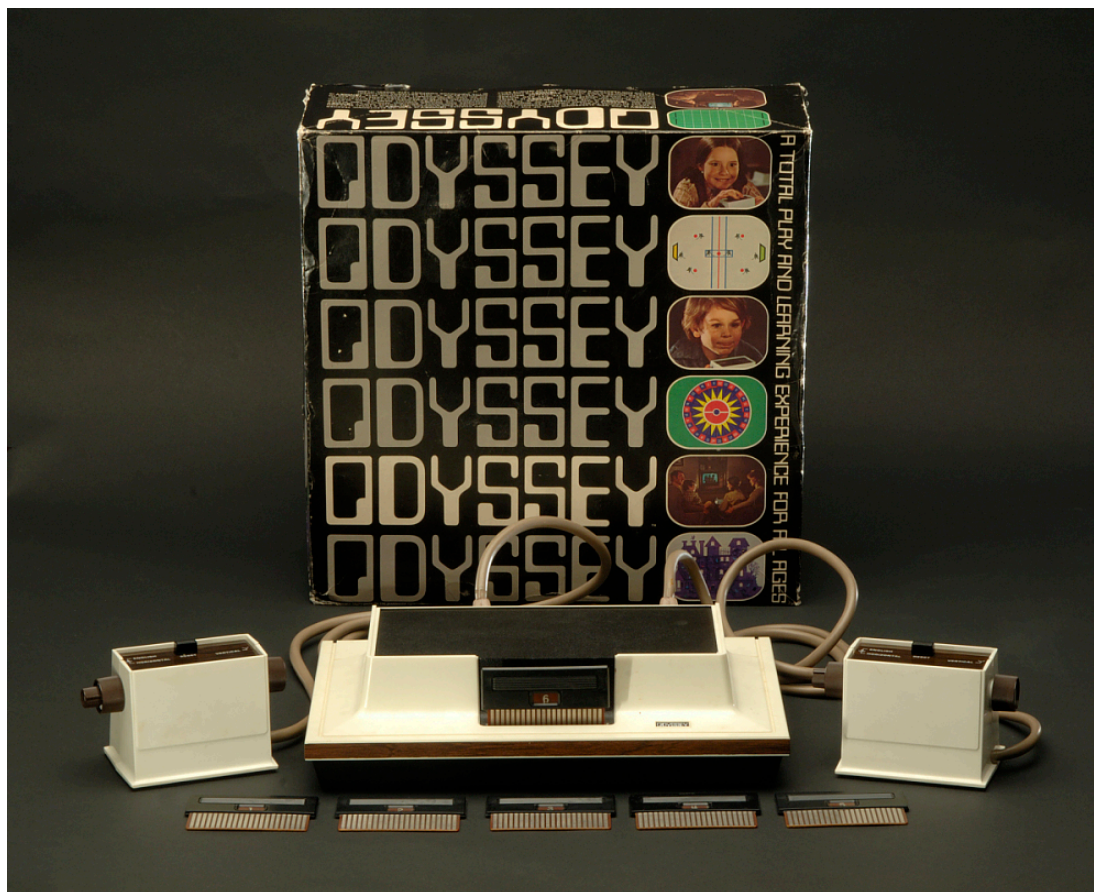


Рис 1.6 – Odyssey [7]

Незважаючи на те, що телевізійні ігри не були так високо помічені в засобах масової інформації, як аркадні, вони були досить успішними на ринку споживчої електроніки. Аж до середини 1970-х років телевізійні ігри були засновані на технології інтегральних схем, яка складалася з одного або декількох чіпів, на яких були фіксовані програми. У 1976 році кілька компаній шукали рішення для заміни чіпів мікропроцесорами, які дозволили б виробляти програмовані домашні відеоігри та ігрові приставки.

Концепція ігрових консолей була великим кроком до нинішньої форми домашніх відеоігор. Він змінив природу домашніх відеоігор як продукту, провівши межу між апаратним та програмним забезпеченням. Споживачам не потрібно було купувати нову машину (обладнання), щоб грати в кожен нову

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

гру, їм потрібно було тільки купити нову гру (програмне забезпечення) і грати в неї на машині, яка вже була у них вдома.

Як було сказано раніше, ця розробка заклала основу для використовуваної сьогодні технологічної архітектури, в якій комп'ютери і консолі є платформами для запуску ігор. Цілісність програмного забезпечення та обладнання також створило нові можливості для бізнесу у вигляді розробки ігор.

### 1.1.2 1980-і роки

На початку 80-х років минулого століття продажі консольного обладнання та програмного забезпечення досягли свого піку. Але зародився новий спосіб гри - комп'ютерні ігри. Однією з головних причин, по якій комп'ютерні ігри не розвивалися так швидко, як консольні, була ціна мікрокомп'ютерів. Але приблизно в 1983 році все змінилося, коли ціни на мікрокомп'ютери впали, і домашні комп'ютери стали з'являтися в будинках. Двома найбільш відомими мікрокомп'ютерами цієї епохи є Commodore 64 і ZX Spectrum (рис 1.7).



Рис 1.7 – Commodore 64 і ZX Spectrum [8]

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

11

Незважаючи на те, що мікрокомп'ютери могли стати важливими ігровими машинами, ця здатність не використовувалася активно в маркетингу. У деяких виробників комп'ютерів були амбітні довгострокові плани по перетворенню домашніх комп'ютерів в центральну частину домашнього господарства, які будуть регулярно використовуватися всією сім'єю. І оскільки ігри розглядалися як дитячі іграшки, маркетинг домашніх комп'ютерів ніколи не виділяв ігри як їх центральну функцію, а тільки як один з варіантів.

Незважаючи на зусилля виробників, профілювати мікрокомп'ютери не як ігрові пристрої, а як комп'ютери загального користування, ігрові можливості комп'ютера дійсно направляли в розробку нових комп'ютерів. 1980-ті роки були часом мікрокомп'ютерів, і здавалося, що консольні ігри померли за цей час. Протягом 1990-х років консольні ігри повернулися, і в цьому десятилітті найбільш помітний розвиток апаратного забезпечення сталося з домашніми відеоіграми, а не з комп'ютерами, повернувши перевагу консолей. Лідером цієї розробки була Nintendo, зображена на рисунку 1.8.



Рис.1.8 – Nintendo [9]

<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>

**ІАЛЦ.467800.003 ПЗ**

**Арк.**

**12**

Нові консолі мали кілька основних конкурентних переваг в порівнянні з комп'ютерами. Багато з привабливих нових ігор були доступні виключно для нових консолей, і тому пропозиція консольних ігор була конкурентоспроможною в порівнянні з іграми, пропонованими для комп'ютерів. Нові консолі також були відносно дешевші, ніж мікрокомп'ютери, що полегшувало їх отримання. І, завдяки технологічному розвитку, ігрова продуктивність консолей була краще.

Незважаючи на те, що консолі повернулися на ринок, розробка комп'ютерних ігор і комп'ютерного обладнання також тривала. Розробка орієнтована в основному на IBM-сумісні платформи ПК. Комп'ютерні технології розвивалися дуже швидко протягом 1990-х років, і обчислювальна потужність комп'ютерів збільшувалася щорічно, що дозволило поліпшити багато моментів, зокрема графіку. Цей розвиток зробив ПК все більш привабливою платформою для видавців програмного забезпечення.

### **1.1.3 Початок 21 століття**

Незважаючи на те, що перші ігри, такі як Spacemar, були іграми з примітивними мережевими можливостями, в наступні десятиліття це не було напрямком розвитку ігор. Аж до середини 1990-х ігри були в основному автономними. Використання цих багатокористувацьких здібностей зазвичай вимагало, щоб гравці перебували в одній кімнаті, граючи в гру на одній машині, наприклад, використовуючи загальний екран.

Першим ігровим жанром, який дійсно почав використовувати можливості мережевої багатокористувацької гри, були шутери від першої особи (FPS-FirstPersonShooter). Такі ігри, як Quake (випущена в 1995 році), зображена на рисунку 1.9, проклали шлях сьгоднішнім FPS-іграм.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	<b>Арк.</b>
						13
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		



Рис.1.9 – Quake [10]

Інший жанр, що пішов за FPS - це розраховані на багато користувачів рольові онлайн-ігри (MMORPG-MassivelyMultiplayerOnlineRolePlayingGame). Фактично, одна з найбільших історій успіху ігрового світу - MMORPG під назвою World of Warcraft (WoW, рис. 1.10). У січні 2011 року у WoW було понад 12 мільйонів передплатників, що зробило її найпопулярнішою MMORPG, заснованої на підписці (Blizzard Entertainment 2011).



Рис. 1.10 – World of Warcraft [11]

Сьогодні все більше і більше ігор пропонують розширені онлайн-

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

можливості. Одним з переваг цих ігор є можливість грати з іншими людьми і проти них, а не з комп'ютером. Гра в ігри стала більш соціальним заходом, ніж раніше, і онлайн-ігри об'єднують безліч людей, незалежно від часу і місця.

Багато людей в якості прикладів перших 3D-ігор зазвичай згадують Doom або Wolfenstein 3D, але справжнім прабатьком тривимірних шутерів (ще й з мультиплеєрів) була гра, випущена в стінах NASA (NationalAeronauticsSpaceAdministration) в 1973 році - називалася вона Maze War (рис 1.11).



Рис. 1.11 – Maze War [12]

В ті часи не було движків, програмісти з великими труднощами домагалися ліцензій і початкових кодів якоїсь існуючої гри, щоб її доопрацювати, або ж просто писали все з нуля. На це йшла лівова частка часу розробки самої гри.

До речі, проводилося безліч дослідів, де вчені вивчали вплив ігор на дітей та дорослих. В одному з експериментів Берлінські вчені запропонували учасникам протягом двох місяців кожен день по 30 хвилин грати в Mario. В результаті виявилось, що гра розвиває відділи мозку, що відповідають за пам'ять, чуттєве сприйняття і м'язовий контроль. Особливо хороші результати виявилися у тих, хто зізнався, що грає із задоволенням. А в дослідженні опублікованому в журналі Social Psychiatry and Psychiatric Epidemiology [2] вчені з Колумбійського університету в Нью-Йорку змогли дослідити, що ігри допомагають навчатися краще. Діти, які грають у відеоігри по п'ять і більше годин на тиждень, часто відчувають менше психологічних проблем і краще

<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>

**ІАЛЦ.467800.003 ПЗ**

**Арк.**

**15**

вчатся, ніж ті, хто грає менше.

## 1.2 Різниця між 2D та 3D іграми.

2D-ігри, як випливає з назви, є іграми з двома осями руху. Як правило, це так звані «плоскі» ігри, в яких можна переміщатися вліво і вправо, а також можливо вгору і вниз. Оскільки у них не так багато варіантів руху, двовимірні ігри часто простіше, ніж їх 3D-аналоги. Крім того, елементи управління в 2D-іграх зазвичай відносно прості. Оскільки у персонажа немає повного діапазону тривимірних рухів, у нього менше можливих рухів і взаємодій з іншими об'єктами. Через 2D-ландшафт кожен спрайт (об'єкт) має координати X / Y, які точно визначають його місце розташування. Камера в 2D-іграх теж сильно спрощена. Зазвичай вона дивиться прямо на сцену з боку, тому немає такої перспективи, як в 3D-іграх.

3D-ігри, навпаки, включають повний рух в тривимірних площинах. Це означає, що гравець може переміщатися в «реальному світі», де він може повертатися на 360 градусів, а об'єкти мають довжину, висоту і глибину.

Тривимірні ігри принципово відрізняються від двовимірних (2D), в яких ігровий простір складається з готових плоских зображень - мальованих від руки або створених художником в тривимірному середовищі моделювання і потім збережених, як звичайні зображення. Завданням комп'ютера являється лише комбінування цих зображень, ось чому такі низькі вимоги 2D ігор до комп'ютера. Тобто, 2D графіка – розділ комп'ютерної графіки, що працює з зображенням, розташованим у двох вимірах, має доступ тільки до двох параметрів – ширина і висота. А 3D графіка – розділ комп'ютерної графіки, об'єктом якого є зображення, що формується в трьох вимірах, до параметрів.

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						16
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		

якого доданий третій вимір - глибина. Простіше кажучи, 3D – це об’ємне зображення. Можемо зробити висновок, що 3D-ігри набагато складніше 2D-ігор.

### 1.3 Аналіз сучасних існуючих варіантів

Ігрова індустрія з кожним роком все більше набирає обертів, все більше ігор створюються, які стають частиною попкультури. Розробка ігор з великими бюджетами, так звані «AAA-ігри» (triple-A), може коштувати десятки, а то й сотні мільйонів доларів, як в випадку з Grand Theft Auto 5, бюджет якої понад 200 мільйонів доларів.

У статті [17] подано перелік із найдорожчих ігор за останні роки, можна побачити що бюджети які виділяються на ігри вже з легкістю можна порівняти з бюджетами голлівудських фільмів. Ось, наприклад, Cyberpunk 2077, з бюджетом в \$330 млн., який вийшов в 2020 році, можна зрівняти з культовим фільмом 2015 року Месники: Ера Альтрона, бюджет якого склав 360 млн\$, а це один з найдорожчих фільмів в історії. Індустрія дуже швидко розвивається, технології сьогодні дозволяють гравцю повністю зануритися в гру і не думати про навколишній світ, графіку в іграх роблять неймовірно реалістичною, так як компанії роблять акцент на об’єктивності своєї гри, і сучасні комп’ютери дозволяють своїм користувачам забути про маленький FPS та отримувати всі можливості сучасних ігор. Аналізуючи всю інформацію, можна зробити висновок, що в найближчі роки ми зможемо побачити проєкти з бюджетами, які будуть перевищують ВВП (Валовий Внутрішній Продукт) деяких країн.

Розглянемо найпопулярніші варіанти 3D ігор. За допомогою інтернету та сервісу по швидкому пошуку інформації, я знайшов сайт зі списком 3D ігор, починаючи з найпопулярніших і рейтингових.

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

### 1.3.1 Grand Theft Auto 5

Grand Theft Auto V або просто GTA V – консольно-комп'ютерна відеогра в жанрі екшн-пригода з величезним відкритим для гравців світом, видана компанією Rockstar Games, зображена на рисунку 1.12. Спочатку гра була випущена тільки на ігрові консолі PlayStation та Xbox в 2013 році, а через 2 роки - для персональних комп'ютерів з ОС (Операційна Система) Windows. Є п'ятнадцятою за рахунком грою серії Grand Theft Auto, які випускаються з 1997 року. Бюджет цієї гри склав 256 млн\$. Так як і у всіх минулих іграх, тут також є свій саундтрек і навіть не один.



Рис 1.12 – Grand Theft Auto

### 1.3.2 Minecraft

Minecraft - комп'ютерна інді-гра в жанрі так званої пісочниці, написана шведським розробником Маркусом Перссоном. Назва складається з двох слів Mine – шахта (з англ.) + craft – ремесло (з англ.). Перша версія гри з'явилася в 2009 році, а вже через 2 роки гра була випущена на ПК. У найближчі роки гра була оптимізована під Linux та macOS, а також на Android, iOS і Windows Phone, і на ігрові консолі теж. Ще через 3 роки компанія Microsoft купила права на Minecraft, а заодно і компанію Маркуса, Mojang AB (яка випустила

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	<b>Арк.</b>
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		<b>18</b>

Minecraft) за 2,5 мільярда долларів.

Гра була написана на Java. Minecraft дає гравцю доступ в змінюваний тривимірний світ (рис 1.13), який повністю складається з кубів – гравець може вільно переробляти, створюючи з цих кубів різні побудови, як прості, так і складні споруди, все що гравець захоче. Ця ексклюзивна та новаторська особливість зробила гру схожою з конструкторами, як LEGO. Minecraft пропонує гравцю повну свободу дій: гравець може досліджувати світ, гравець може добувати корисні копалини, гравець може боротися з суперниками, гравець може будувати різні споруди та багато іншого. У грі також є свої фірмові звуки та саундтрек який з легкістю впізнає кожен з фанатів.



Рис. 1.13 – Minecraft [14]

### 1.3.3 Mafia 2

Mafia II (рис 1.14) – культова своїм сюжетом, комп'ютерна гра з відкритим світом, що поєднує в собі декілька жанрів: автомобільний симулятор і шутер від третьої особи.

Гра була випущена в 2010 році на ПК з ОС Windows, а також на ігрових

					<b>ІАЛЦ.467800.003 ПЗ</b>	<b>Арк.</b>
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		19

консолях Xbox та PlayStation; а через рік вийшла для користувачів macOS. Гра отримала у цілому позитивні відгуки критиків, особливо щодо сюжету, ігрового процесу та персонажів, але критикувався обмежений дизайн світу, як для 2010 року, та велика кількість багів, які можна було уникнути на етапі розробки. Mafia 2 заціпила своїх гравців неймовірним сюжетом, котрий наступна частина гри не змогла перевершити. У грі теж є свій саундтрек, який при проходженні гри запам'ятався всім гравцям.



Рис 1.14 – Mafia II [15]

### 1.3.4 Call of duty

Call of Duty - культова серія комп'ютерних ігор в жанрі шутер, які випускаються з 2003 року. Серія складається з сімнадцяти основних ігор і десяти відгалужень. В 2009 році, в день випуску Modern Warfare 2 (рис 1.15) було продано понад 4 мільйонів копій. Таким чином, гра встановила новий рекорд продажів за день обігнав минулого лідера цього списку - GTA IV. В 2012 році на офіційному сайті Playstation, автори опублікували список ігор, серед яких найбільше власників платинових

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

трофеїв. Call of Duty: Modern Warfare 2 зайняла друге місце у списку. Після виходу наступної частини, Call of Duty Black Ops перевершила колишнього лідера Modern Warfare 2 за продажами за день. В 2011 Modern Warfare 3, знову встановила новий світовий рекорд продажів за день, обігнавши своїх попередників. Серед найбільш популярних ігрових серій всіх часів, опублікованому у виданні Книги рекордів Гіннеса за 2010 рік, серія ігор Call of Duty посіла друге місце.

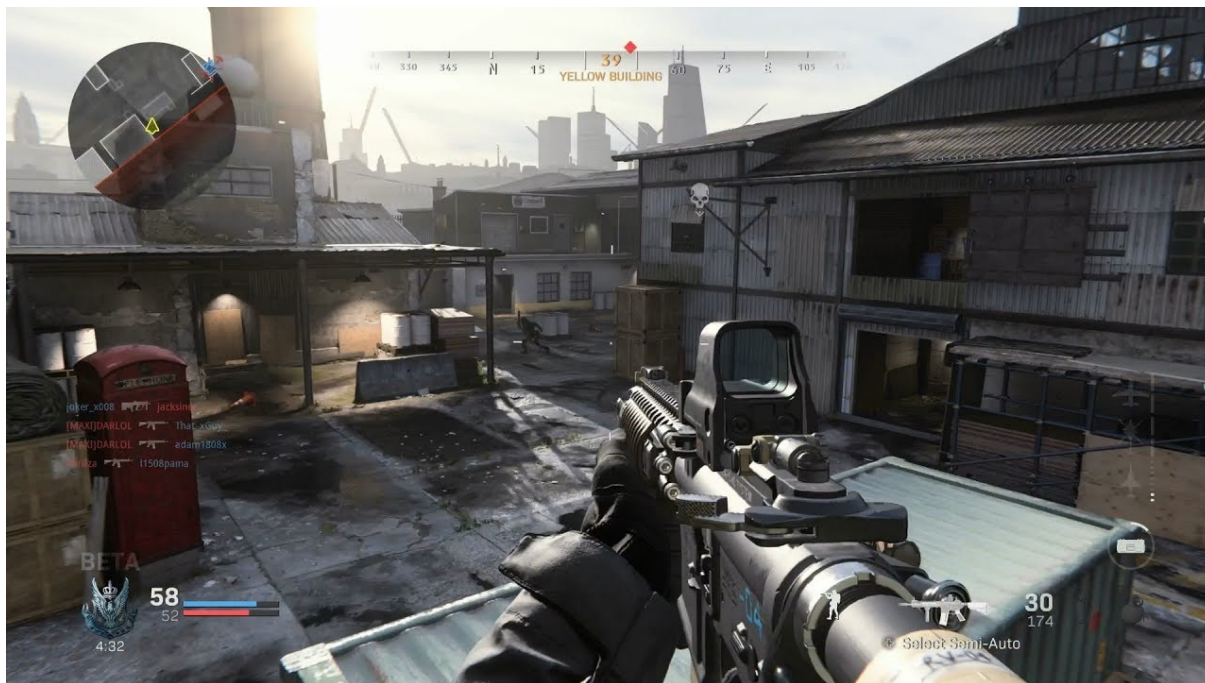


Рис.1.15 – Call of Duty [16]

### 1.3.5 Assassin's creed

Assassin's Creed - це серія ігор жанру пригодницька-стелс гра з відкритим світом, випущена компанією Ubisoft. Перша частина вийшла в 2007 році, нові серії виходять по цей день. Особливістю цієї гри являється особлива увага до прихованого переміщення і паркуру, що робить її унікальною на ринку.

Згідно зі звітом, опублікованим корпорацією Ubisoft 5 років назад, сукупна кількість продажів ігор серії Assassin's Creed перевищили 100 мільйонів. Серія

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

отримала в основному позитивні відгуки критиків. Його хвалили за амбітний дизайн гри, візуальні ефекти і розповідь, але критикували за технічні проблеми і щорічні випуски майже кожної частини.

Жовтня 2020 року загальний обсяг продаж серій досяг 155 мільйонів копій більш ніж 100 мільйонам гравцям, ставши найбільш продаваною франшизою Ubisoft і однією з найбільш продаваних франшиз відеоігор всіх часів. Гру зображено на рисунку 1.16.

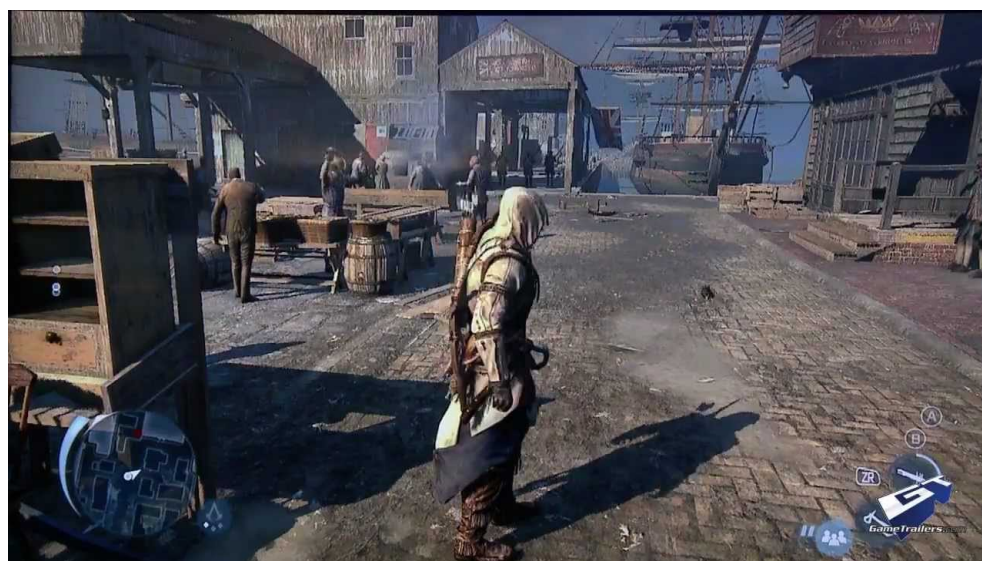


Рис.1.16 – Assassin's Creed [17]

#### 1.4 Категоризація ігор

Сьогодні на ринку представлено незліченну кількість цифрових ігор. Щоб контролювати цю кількість, цифрові ігри зазвичай поділяють за жанрами відповідно за їх змістом і діями по взаємодії, які вони містять. Також необхідно відзначити, що існує не тільки один правильний спосіб категоризації ігор, і деякі ігри можуть бути віднесені до декількох різних жанрів.

Через постійно зростаючу кількість ігор, жанри постійно змінюються.

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

22

Також здається, що (принаймні) у всіх великих видавців ігор є свої визначення ігрових жанрів. Приклади використовуваних жанрів: бойовик, пригоди, стратегія, рольові ігри, симулятори та т. д.

Багато ігор мають елементи різних жанрів, наприклад, та сама GTA. Такі ігри вписують або до одного з тих жанрів, який в грі є основним, або до всіх, які є в грі.

Колись жанри відеоігор були досить чітко визначені, але в наші дні це не так. Необхідно розуміти все більшу різноманітність жанрів і піджанрів, особливо коли розробники ігор змішують і змішують різні типи ігор новими і несподіваними способами.

Це означає, що ландшафт відеоігор постійно розвивається. Студії працюють по щільному графіку і стежать за тенденціями, коли з'являється можливість.

#### **1.4.1 3D-шутери**

Шутер - давній жанр, який розвинув кілька ранніх відгалужень і розділився на два основних піджанра: шутер від першої особи (FPS) і шутер від третьої особи TPS (ThirdPersonShooter).

Ключова відмінність - перспектива. FPS імітує типову людську точку зору, показуючи те, що ігровий персонаж бачить, в таких франшизах, як Half-Life, зображену на рисунку 1.17, Call of Duty і DOOM. TPS відсуває перспективу назад і демонструє всього вашого персонажа і навколишнє середовище, як, наприклад, в серіях Gears of War і Tom Clancy's The Division.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	<b>Арк.</b>
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		<b>23</b>



Рис. 1.17 – Half-Life

### 1.4.2 Файтинг

Геймплей в таких іграх зазвичай складається з битв двох та більше суперників.

Приклади Файтингу: Mortal Combat, Tekken (рисунок 1.18).



Рис. 1.18 – Mortal Combat [18]

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

### 1.4.3 Пісочниця або SandBox

Терміни «пісочниця» або “SandBox” може бути більш впізнаваним через його використання в технологіях або навіть як відкритий режим, доступний в деяких іграх. Це часто пов'язано з вибором гравця, відкритим середовищем і нелінійним ігровим процесом.

У цих іграх у гравців часто менш конкретні цілі. Гравець може зіштовхнутися з безліччю завдань, які ви можете виконати різними способами. Це привертає гравців до більш захоплюючого досвіду, заохочуючи експериментувати в незнайомій механіці.

Приклади SandBox: Minecraft, Grand Theft Auto, Sims (рис. 1.19).



Рис 1.19 – Sims [19]

### 1.4.4 Симулятори

Програма, максимально імітує фізичну поведінку і управління будь-яким складним об'єктом, таким як автомобіль, літак і так далі. В випадку ігор других жанрів, ігри намагаються розважати гравця під час гри, використовуючи різні

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

прийоми, а в випадку симулятора, унікальність є в повноті та реалістичності моделювання.

Приклади Симуляторів: Microsoft Flight Simulator, зображений на рисунку 1.20.



Рис. 1.20 – Microsoft Flight Simulator [20]

### 1.4.5 Стратегії

У типовій стратегії люди управляють різними фракціями і змагаються один з одним, вироблюючи свою стратегію, звідси і назва жанру - стратегія. Ці ігри зазвичай містять в собі управління ресурсами і картами і часто мають вигляд зверху вниз. Гравець управляє не одним персонажем, як в інших жанрах, а цілим підрозділом, армією, країною і т.д.

Приклади Стратегій: Cossacks: Back To War, Heroes of Might and Magic, зображена на рисунку 1.21.

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

***ІАЛЦ.467800.003 ПЗ***

***Арк.***

***26***



Рис. 1.21 – Heroes of Might and Magic

### 1.4.6 Спортивні

Цей жанр пропонує імітацію різних спортивних ігор, найбільшої популярності набули імітації наступних видів спорту: футболу, хокею, баскетболу, тенісу і гольфу. Різноманітність спортивних ігор розширилася, пропонуючи повноцінні партнерські відносини з великими спортивними організаціями.

Приклади Sport simulator : FIFA (рис 1.21), NBA, UFC.

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

27



Рис. 1.22 – FIFA [21]

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ ДО РОЗДІЛУ 1

В даному розділі було розглянуто історію ігор та найвідоміші існуючі ігри на цей момент. Як ми побачили ігри класифікуються великою кількістю різних жанрів, і за останні роки кількість жанрів, як і ігор відповідно, зростає з неймовірною швидкістю, як і бюджети які виділяють на створення однієї гри.

Вивчаючи всю історію ігрової індустрії, було досліджено що, вона розвивалася с тою кількістю технологій і коштів, які були раніше, і відповідно зробити висновок, що ця індустрія буде тільки розвиватися, адже з кожним роком гравців стає все більше і бюджети також зростають. Хто б міг подумати, навіть якісь 20 років назад, що зараз на одну гру буде виділятися 300 мільйонів доларів, як в випадку з Grand Theft Auto V(256 млн. дол), Call of Duty: Modern Warfare 2 (\$302 млн.), Cyberpunk 2077 (\$330 млн.).

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	Арк.
						29
<b><i>Зм.</i></b>	<b><i>Арк.</i></b>	<b><i>№ докум.</i></b>	<b><i>Підпис</i></b>	<b><i>Дата</i></b>		

## РОЗДІЛ 2

### ОГЛЯД СЕРЕДОВИЩ РОЗРОБКИ ТА МУЗИКАЛЬНОГО ПРОДЮСУВАННЯ

#### 2.1 Як зробити 3D гру

Після розмови з людьми які працюють в ігровій індустрії вже багато років, по їх словам, стало зрозуміло, що у головах деяких початківців-розробників при цьому виникає картина шикарної RPG (RolePlayingGame) з відкритим світом. Але спроба на чистому ентузіазмі взятися за реалізацію масштабного проєкту часто призводить до розчарування. Лише через десяток-другий зібраних прототипів приходить розуміння, який обсяг роботи потрібен для розробки кожного елемента гри. Задумана рольова гра поступово перетворюється в інді-хоррор (гра з низькими вимогами), відкритий світ змінюється коридорами, а механіки зводяться до неквапливого збору записок.

Тому краще починати з малого і поступово додавати в гру нові можливості. Тоді отриманий в процесі досвід виявиться набагато ціннішим, а наступна спроба гарантовано буде краще.

Розробка ігор для різних платформ значно відрізняється, хоча б тому, що у них різні технічні характеристики. А поки розглянемо особливості кожної з платформ і перейдемо до движків.

#### 2.2 Існуючі ігрові платформи

##### 2.2.1 Мобільні пристрої

Тачскрін для введення і виведення інформації - отже, пальці не повинні закривати важливі елементи інтерфейсу.

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Смартфони і планшети має бути зручно тримати, щоб грати однією / двома руками. Дуже добре підходить для казуальних ігор (Match-3, Hidden object, інтерактивних історій і так далі), а для шутерів - не дуже.

Обмежена оперативна і графічна пам'ять, які часто поєднані. Потрібно постійно стежити за ними, відвантажувати непотрібні ресурси, текстури і утискувати звук, тобто використовувати формати з компресією.

Можна зробити круті шейдери як в Crysis, але на більшості телефонів це буде моторошно гальмувати.

Часті втрати пакетів, пінг в 200 мс. – є нормою. У разі мережевих ігор це потрібно враховувати.

Поширення через маркети (App Store, Google Play, Amazon). Знадобиться підтримка API (ApplicationProgrammingInterface) покупок, соціальних функцій і так далі.

Для Android доведеться враховувати величезну кількість гаджетів з різною продуктивністю, співвідношенням сторін екрану і роздільною здатністю.

### **2.2.2 ПК**

Введення з клавіатури і миші - те, до чого гравці звикли з дитинства.

Моделей моніторів багато, вони відрізняються частотою зміни кадру, розмірами, роздільною здатністю - це все потрібно враховувати.

Величезна різноманітність процесорів, відеокарт і інших комплектуючих, що робить тестування гри трудомістким процесом.

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

***ІАЛЦ.467800.003 ПЗ***

***Арк.***

***31***

### 2.2.3 Консолі

Управління з джойстика. Краще підходить для аркад, файтингів, ігор від 3-ї особи, але не так зручно для шутерів.

Обмежена кількість конфігурацій пристроїв, наприклад, якщо розробка ведеться під Xbox One або PS4, то потрібно знати особливості тільки цих пристроїв, а значить і тестувати буде простіше. На відміну від різних конфігурацій ПК або пристроїв на Android.

Не всі плагіни переносяться або добре працюють, наприклад, мережеві бібліотеки і плагіни аналітики.

Щоб випустити гру на консолі, потрібно пройти ліцензування - процес перевірки відповідності гри стандартам платформи. Це довгий процес, з безліччю умов і обмежень.

### 2.2.4 Web

Обмеження на розмір гри, оскільки вона буде завантажуватися в браузері. Ніхто не любить довго чекати. А ще дехто грає в браузері телефону і платять за трафік. Загалом, робити повноцінний AAA-тайтл немає сенсу.

Обмеження по 3D (використовується WebGL). Тому в Web в основному виходять 2D-ігри.

Обмеження по мережевій грі, адже звичайні сокети недоступні. Можна робити запити по 'https' або використовувати WebSockets. В основному на Web можна робити прості ігри з невеликою кількістю запитів до сервера.

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2.2.5 VR

Незвичайне управління: головою, рукавичками, переміщенням, джойстиком. Всі ці пристрої потрібно підтримувати, у них зазвичай свій SDK (SoftwareDevelopmentKit). Крім того, управління потрібно зробити «природним» для людини.

«Екран» VR-шолома ділиться на дві частини - по одній на око. Тому, чим вище роздільна здатність, тим якісніше виходить зображення.

Зовсім інший користувацький інтерфейс, в порівнянні з ПК і мобільними пристроями. Зазвичай він тривимірний, а щоб натиснути на який-небудь елемент потрібно затримати погляд на певній кнопці.

## 2.3 Огляд середовищ розробки

У кожного розробника ігор різні цілі і вимоги до функцій. Почнемо з тих які дозволяють розробляти під кілька платформ відразу.

### 2.3.1 Unity

Unity, безсумнівно, дуже популярна ігрова платформа. На цей вражаючий двигун доводиться 45% всієї розробки ігор, майже половина ринку. Інтерфейс рушія зображень на рисунку 2.1.

Платформи: мобільні пристрої, ПК, Mac, Linux, консолі, Facebook, WebGL, VR і інші.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	<b>Арк.</b>
						<b>33</b>
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		

Unity ідеально підходить для розробки під мобільні пристрої. На ньому зроблені Angry Birds 2, Hitman Go, Heartstone, Monument Valley, Fallout Shelter, Ori and the Blind Forest, Pillars of Eternity, Firewatch, Inside, Pokémon Go, Super Mario Run, Cuphead, Escape from Tarkov, Life Is Strange: Before the Storm і безліч інших популярних ігор.

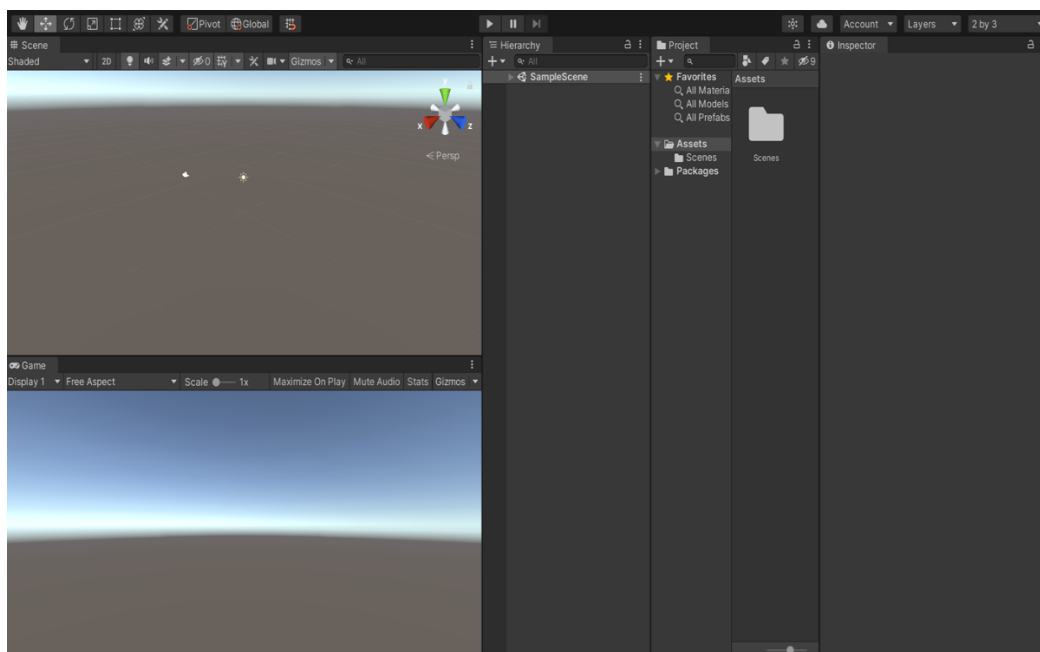


Рис 2.1 – Unity

В Unity можна спокійно розробляти як 2D, так і 3D-проекти. В Asset Store є багато готових платних і безкоштовних рішень: моделі, текстури, анімації і повноцінні проекти. Існує дуже багато навчальних матеріалів, як від самих Unity, так і від ентузіастів на YouTube.

Мова програмування: C#, в порівнянні з C++ у неї менше можливостей, зокрема, це стосується роботи з пам'яттю. Також підтримується JavaScript, який насправді UnityScript. Існують плагіни, які дозволяють робити ігри без написання коду, наприклад, Playmaker. Правда, за нього доведеться заплатити 45 доларів.

<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>

**ІАЛЦ.467800.003 ПЗ**

**Арк.**

**34**

Розробників приваблює двигун Unity, тому що в порівнянні з іншими прогресивними 3D-двигунами він вимагає менше зусиль, щоб звикнути. Unity має безліч функцій, і його універсальність неймовірна. По-перше, за допомогою одного двигуна можна створити практично все, що завгодно: 2D- і 3D-проекти різного масштабу. Основне обмеження полягає в його графічних можливостях. Однак є кордон реалістичного рендерингу, які Unity поки не може перевершити. Але, можливо, ця особливість надає двигуну деяку індивідуальність. У той час як всі інші роблять все можливе, щоб отримати максимально реалістичне зображення, Unity робить його барвистим і приємним на вигляд і концентрується на зручність використання і модульною системою.

Інтерфейс приємний у використанні, дуже інтуїтивно зрозумілий і вимагає не багато часу, щоб освоїти його, не дивлячись на його багату функціональність. Зручність використання Unity забезпечує високу швидкість прототипування, що досягається шляхом підтримки налаштування в реальному часі, що дозволяє розробнику вносити зміни під час гри. Unity виділяється в швидкозростаючій індустрії мобільних ігор, і чимало з найпопулярніших мобільних 3D-ігор були розроблені з її допомогою.

Торговий майданчик Unity містить більше 15 тисяч різних скриптів, шейдерів, 3D-моделей, аудіосемплів і інших матеріалів для використання в проєкті. Володіючи фундаментальними знаннями C#, можна легко створити актив. Крім того, у Unity є активна спільнота з ще більш різноманітним матеріалом.

Серед інших переваг - інструменти аналітики та монетизації, інтеграція з Twitch, хмарні сервіси Unity і управління пам'яттю. До недоліків можна

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						35
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		

віднести програму з закритим кодом і неможливість використання скрипта Unity в проектах, які підтримуються іншими двигунами.

Вартість рушія: безкоштовно, якщо розробник заробляє на грі менше \$ 100 тисяч в рік. Інакше - підписка, але тоді це не буде великою проблемою.

### 2.3.2 Unreal Engine

Дуже просунутий движок (рис 2.2), спільнота якого останнім часом швидко зростає, чому сприяє компанія-розробник Epic Games.

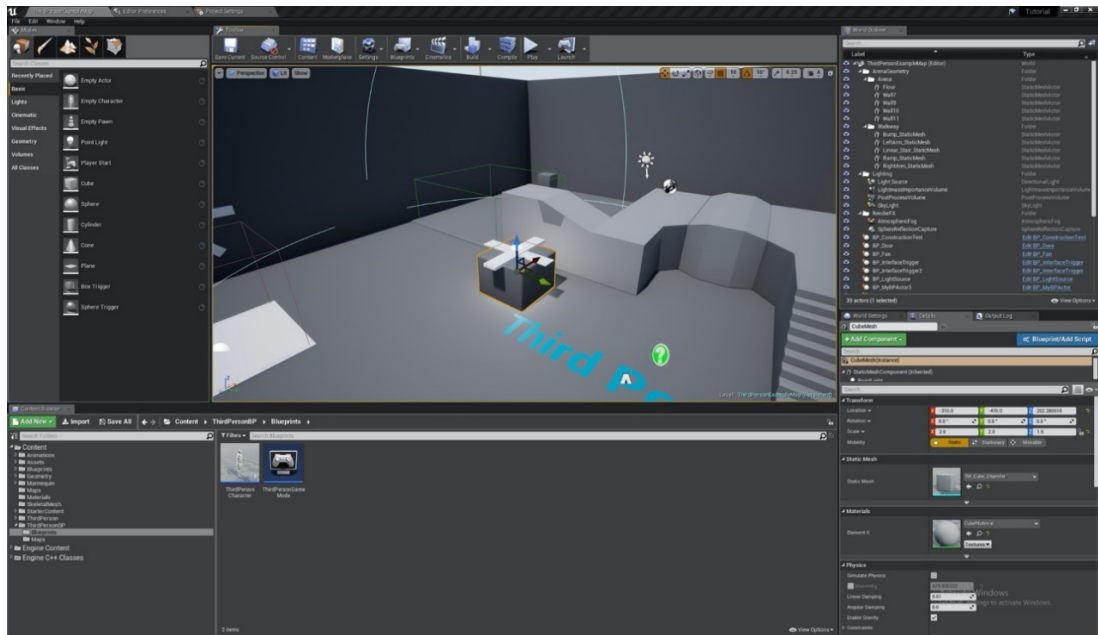


Рис 2.2 – Unreal Engine [22]

Платформи: ПК, консолі та мобільні пристрої. На Unreal Engine зроблені одні з найпопулярніших шутерів за останній час: Fortnite і PUBG. Ще на ньому зроблені серія Infinity Blade, Batman: Arkham Knight і Life is Strange.

Мова розробки: C ++.

Вартість движка: 5% роялті, якщо розробник заробляє на грі більше \$ 3000 за квартал.

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

36

### 2.3.3 CryEngine

У той час як Unity і Unreal домінують на ринку, CryEngine (рис 2.3) також заслуговує на увагу. Метою розробників Crytek було створення барвистих і реалістичних шутерів від першої особи для ПК і консолей, і вони прекрасно впоралися з цим завданням. Движок дозволяє розробникам експериментувати з освітленням, частками, тінями, ландшафтом і т. д., що призводить до відмінної фізики і відточеним реалістичним пейзажам.

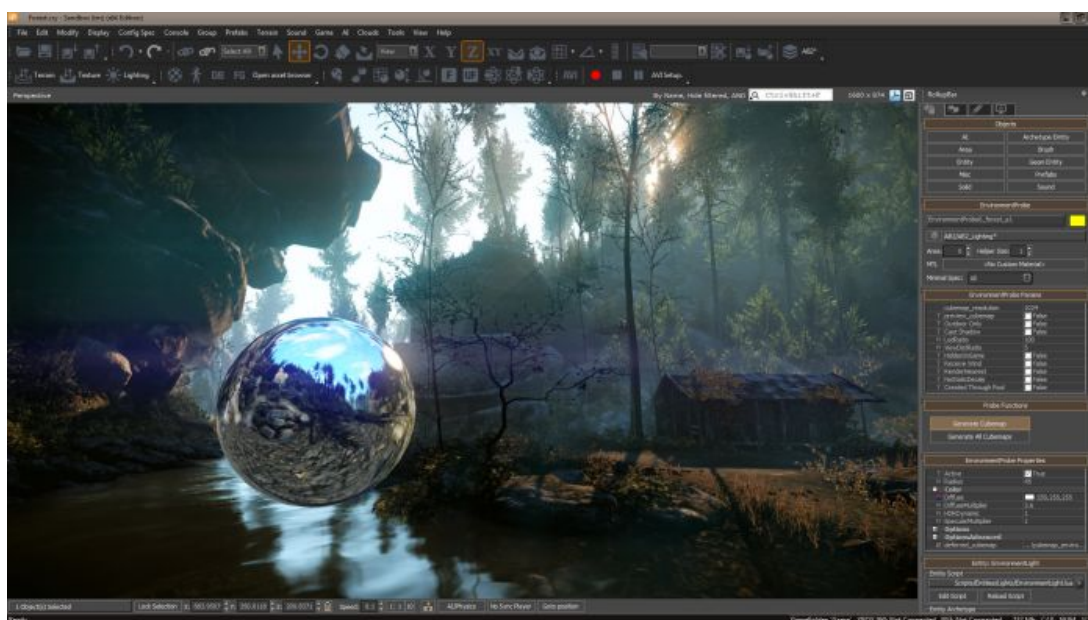


Рис. 2.3 – CryEngine [23]

У CryEngine все ще є проблеми, пов'язані з використанням цієї платформи. Багатьом розробникам довелося нелегко, так як спільнота пасивна і нема до кого звернутися за допомогою. Тим не менш, це не проблема для великих компаній-розробників, які купують CryEngine і адаптують його під свої потреби. Це говорить про те, що CryEngine більш ніж здатний задовольнити більшість потреб розробників ігор.

CryEngine став відомим після виходу Crysis - проривної для свого часу гри. На ньому вийшло дуже багато крутих великих ігор: перший Far Cry,

					<b>ІАЛЦ.467800.003 ПЗ</b>	<b>Арк.</b>
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		<b>37</b>

MechWarrior Online, Sniper: Ghost Warrior 3, Armored Warfare, Homefront: The Revolution, Prey 2017 го року.

Платформи: PC, консолі і VR.

Мова розробки: C ++. Зовсім недавно з'явився Marketplace з Ассет.

Вартість движка: починаючи з п'ятої версії - 5% роялті при доході з гри більше \$ 5000.

### 2.3.4 Lumberyard



Рис 2.4 – Lumberyard [24]

Молодий (2016 року випуску) і безкоштовний движок з відкритим вихідним кодом від Amazon на основі CryEngine для розробки ігор AAA-класу. Головна особливість - вбудована підтримка сервісів від Amazon, наприклад, AWS і Twitch.

На цей момент движок все ще занадто молодий і схожий на свого попередника CryEngine. Інтерфейс був змінений здебільшого, але зручність

<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>

**ІАЛЦ.467800.003 ПЗ**

**Арк.**

**38**

використання все ще потребує поліпшення, щоб повною мірою використовувати багату функціональність. Lumberyard прагне розширити свої можливості за допомогою доступних інтеграцій, включаючи Substance, Perforce і Wwise. Молодий движок означає відсутність спільноти, вичерпних посібників і сховища активів.

Платформи: Windows, PlayStation 4, Xbox One, iOS, Android, VR (Oculus Rift, HTC Vive).

Мова розробки: C ++.

Вартість движка: безкоштовний, але за підтримку сервісів Amazon доведеться заплатити.

### **2.3.5 Інші двигуни**

Останнім часом серед розробників ігор для Web набирають популярність HTML5-движки. В їх основі лежить WebGL, WebAudio і JavaScript. Найпопулярніші движки: Phaser і Turbulenz.

Всі 3D-движки створюють зображення за одним сценарієм.

Моделі і віртуальна камера розташовуються в тривимірному просторі, з урахуванням положення, обертання і масштабу. До анімованих моделей застосовуються відповідні анімації, наприклад, згинається частина моделі, яка прив'язана до суглоба скелета.

Всі моделі покриваються текстурами. Одні текстури повідомляють про колір певних частин моделі, інші - про те, наскільки сильно ці частини відбивають світло, треті містять інформацію про рельєф поверхні і так далі. Текстури - це звичайні картинки. За те, як саме вони будуть накладатися і відображатися відповідають шейдери - свого роду інструкції для відеокарти.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	<b>Арк.</b>
						<b>39</b>
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		

Розраховується освітлення з урахуванням джерел світла, розташування моделей відносно один одного, заздалегідь підготовлених світлових карт (спеціальних текстур, що містять інформацію про освітленості 3D-моделей).

Застосовуються пост-ефекти для фінальної обробки картинки. Наприклад, стилізація під нуар або ефект мініатюри.

Для створення прототипу додавати озвучення в принципі не обов'язково. Можна зробити цілу гру, протестувати геймплей, налаштувати всю графіку і тільки в кінці додати озвучку. Але так як тема цього диплому - комп'ютерна 3D гра з саунд-продюсуванням, це необхідно реалізувати. Як написати саундтрек? Які є ПЗ? Розглянемо самі популярні варіанти.

## **2.3 Аналіз платформ музикального продюсування**

### **2.3.1 Steinberg Cubase**

Платформи: Windows, macOS.

Вартість: від 100 євро, є безкоштовна 30-денна пробна версія.

Віртуальна студія, яка вважається однією з найпотужніших на сьогоднішній день. Це продукт, орієнтований на професійну розробку, але завдяки логічному інтерфейсу любителі і музиканти, які хочуть поліпшити звучання власних композицій, також успішно працюють в Cubase, інтерфейс якої зображений на рисунку 2.5.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	Арк.
						40
<b><i>Зм.</i></b>	<b><i>Арк.</i></b>	<b><i>№ докум.</i></b>	<b><i>Підпис</i></b>	<b><i>Дата</i></b>		

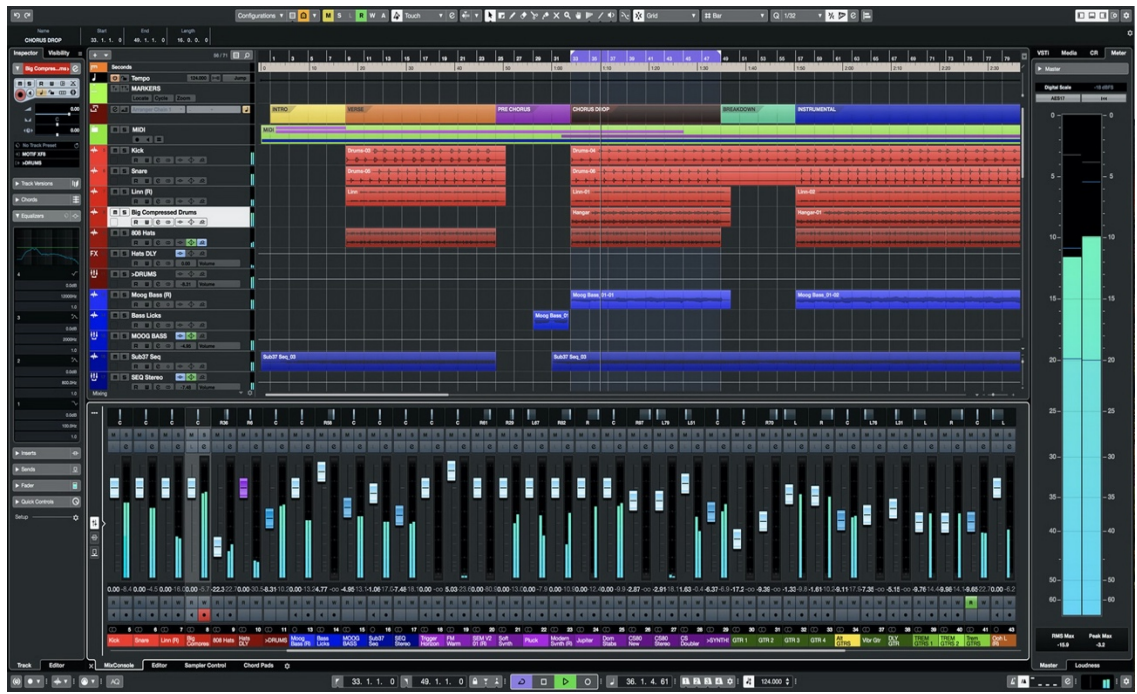


Рис.2.5 – Cubase [25]

У студії доступні всі передові інструменти для редагування звуку на різних етапах: в процесі запису, при зведенні та мастерингу.

Студія працює з VST (VirtualStudioTechnology) і VSTi-плагінами. Тут є зручні шаблони проєктів, багатодоріжковий редактор, засоби для роботи з нотами, комплекти фільтрів, ефектів і вбудований мікшер. У середовищі можна скласти біти, лупи, редагувати і синтезувати нові звуки.

### 2.3.2 Ableton Live

Платформи: Windows, macOS.

Вартість: від 199 доларів, є безкоштовна 90-денна пробна версія.

Цю програму використовують багато діджеїв. Ableton Live (рис 2.6) дозволяє легко створювати, редагувати і програвати музику в режимі реального часу. Зручний інтерфейс, універсальний набір інструментів, звуків, семплів і лупів - все, що потрібно для комфортного творчого процесу.

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

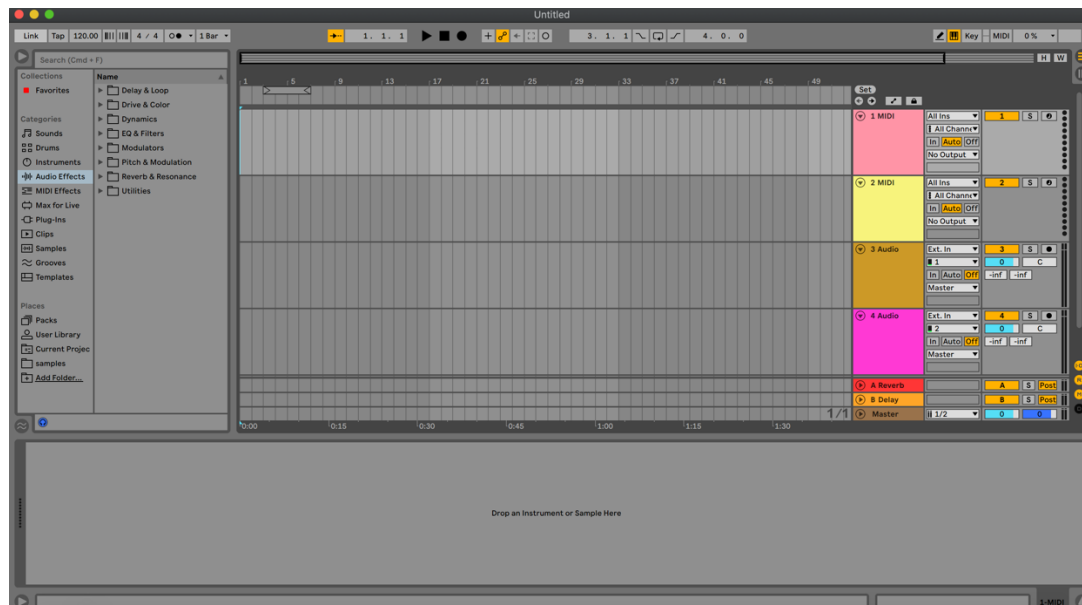


Рис. 2.6 – Ableton Live

Бібліотека Ableton Live включає 13 різних інструментів і 56 ефектів для обробки аудіо і MIDI (MusicalInstrumentDigitalInterface) - треків: Echo, Pedal, Beat Repeat, Amp, Vocoder, Glue та інші. Програма автоматично розпізнає MIDI-клавіатури та інші контролери і, іноді ліцензію на Ableton Live можна знайти в коробці з такими пристроями.

Один з найцікавіших вбудованих інструментів студії - Wavetable. Це оновлений синтезатор, який дозволяє гнучко трансформувати звук за допомогою хвильових таблиць. Їх складали, аналізуючи звук з реальних синтезаторів і інших аналогових інструментів.

### 2.3.3 FL Studio (FruityLoops)

Платформи: Windows, macOS, iOS, Android.

Вартість: для ПК - від 99 доларів, є безкоштовна пробна версія без обмежень за часом.

Потужна віртуальна студія (рис 2.7), в якій особливо зручно працювати з

<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>

**ІАЛЦ.467800.003 ПЗ**

Арк.

42

електронною музикою, хіп-хопом і іншими подібними напрямками.

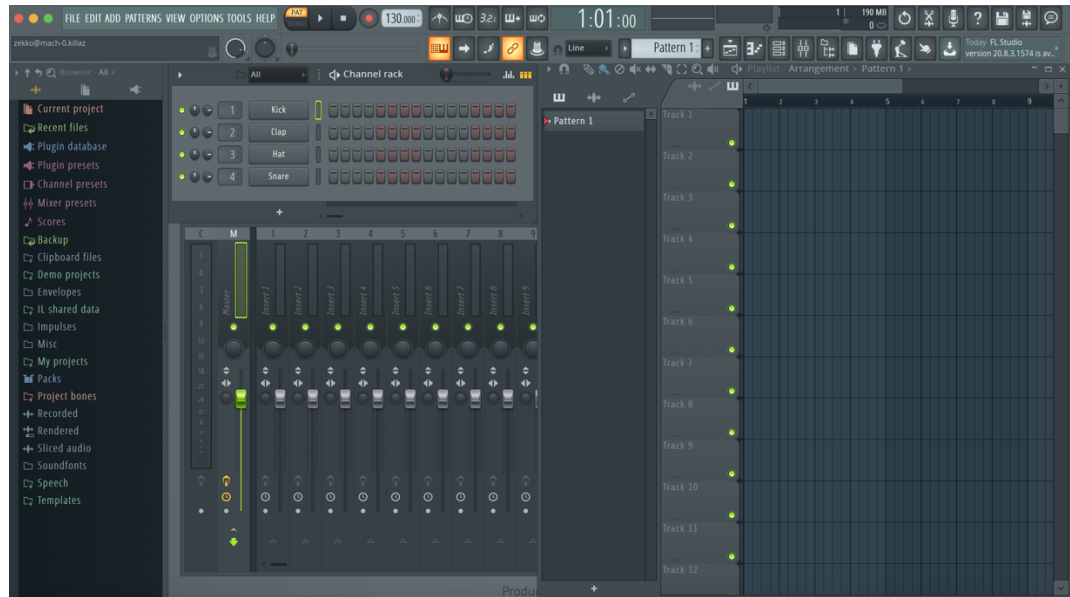


Рис. 2.7 FL Studio

У FL Studio можна зробити цікаве аранжування і домогтися унікального звучання популярних треків. Студія підтримує плагіни VST і VST2, інтеграцію сторонніх інструментів, може приймати дані з декількох MIDI-входів одночасно.

Для кожного віртуального інструменту і ефекту в програмі є безліч пресетів. Також є вікно готових шаблонів і зручний piano roll - щоб було простіше підбирати такти і ноти.

					<b>ІАЛЦ.467800.003 ПЗ</b>	<b>Арк.</b>
						43
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		

## ВИСНОВКИ ДО РОЗДІЛУ 2

В даному розділі ми розібралися як правильно вибрати середовище для розробки і на що потрібно звернути увагу при виборі платформи.

Також в даному розділі було розглянуто платформи для реалізації гри, зокрема Unity, CryEngine, UnrealEngine та Lumberyard. Було проаналізовано всі ці двигуни, порівняно їх плюси та мінуси, і виходячи з задачі було обрано двигун Unity, і саме в ньому буде створена 3D гра.

Також в даному розділі було розглянуто платформи для реалізації саундтреку до нашої 3D гри. Було розібрано основні платформи, такі як Ableton Live, FL studio і Cubase. Для виконання нашої задачі буде використовуватися Ableton Live, так як в цій програмі дуже зручний інтерфейс, з яким не складно розібратися, а також в цій програмі є бібліотека яка складається із великої кількості інструментів та звуків які знадобляться в реалізації проекту.

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

***ІАЛЦ.467800.003 ПЗ***

***Арк.***

***44***

## РОЗДІЛ 3

### РОЗРОБКА СИСТЕМИ

#### 3.1 Вибір технологій. Вибір ігрового рушія

Як вже було вирішено, враховуючи всі задачі щодо гри, всі позитивні та негативні сторони ігрових рушіїв, що розібрані в пункті 2.2 даної дипломної роботи, було вибрано ігровий рушій - Unity.

Важливо зазначити, що в Unity є наявність візуального середовища розробки, міжплатформової підтримки і модульної системи компонентів.

Основою перевагою є те, що двигун Unity3D дає можливість розробляти ігри, не вимагаючи для цього якихось особливих знань. Тут використовується компонентно-орієнтований підхід, в рамках якого розробник створює об'єкти і до них додає різні компоненти. Завдяки зручному Drag & Drop інтерфейсу і функціональності графічного редактора движок дозволяє малювати карти і розставляти об'єкти в реальному часі і відразу ж тестувати результат.

Друга перевага рушія - наявність величезної бібліотеки і плагінів, за допомогою яких можна значно урізноманітнити та прискорити процес розробки гри. Їх можна імпортувати і експортувати, додавати в гру цілі заготовки - рівні, ворогів, патерни, текстури і так далі. Багато продуктів з бібліотеки надаються безкоштовно, інші пропонуються за невелику суму, і при бажанні можна створювати власний контент, публікувати його в Unity Asset Store і отримувати від цього прибуток.

Третя сильна сторона Unity 3D - підтримка величезної кількості платформ,

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						45
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		

технологій, API. Створені на движку ігри можна легко перенести між ОС Windows, Linux, OS X, Android, iOS, на консолі сімейств PlayStation, Xbox, Nintendo, на VR- і AR-пристрої. Unity підтримує DirectX і OpenGL, працює з усіма сучасними ефектами рендеринга.

Фізика твердих тіл, ragdoll і тканин, система Level of Detail, колізії між об'єктами, складні анімації - все це можна реалізувати силами движка.

І в кінці кінців, Unity доступний безкоштовно, що відкриває перед незалежними розробниками двері в ігрову індустрію. Звичайно, існують обмеження: безкоштовна версія двигуна демонструє лого Unity перед запуском гри, а проєкт, створений з її допомогою, не повинен приносити розробнику більше \$ 100 тисяч в рік. Про-версія коштує \$ 125 на місяць, що не так вже й багато в порівнянні з іншими двигунами, причому базова версія містить рівно той же функціонал, що і професійна.

Під ігрову платформу було вирішено обрати ПК, ПК який є в наявності - MacBook Pro (Retina, 15-inch, Mid 2014) з процесором 2,5 GHz 4-ядерный процесор Intel Core i7, відповідно ОС MacOS.

Оскільки гра буде створюватися на Unity, то і використовуватись для даної системи буде мова програмування C#.

В рушії відсутній текстовий редактор, що не дає можливості написання коду на C# прямо в інтерфейсі Unity. Для вирішення цієї проблеми потрібно вибрати середовище розробки для написання коду. Unity також підтримує розробку програм використовуючи Microsoft Visual Studio.

C# - це сучасна і об'єктноорієнтована мова програмування. C# дозволяє розробникам створювати безліч типів безпечних і надійних програм. C# бере

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						46
<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>		

свій початок в сімействі мов С і буде відразу знайомий програмістам на С, С++, Java і JavaScript.

С# надає мовні конструкції для прямої підтримки цих концепцій, що робить С# природною мовою для створення і використання програмних компонентів.

На С# можна створити динамічну сторінку, службу XML, компонент доступу до баз даних, десктопних програм під Windows і клієнтську програму або його компонент. З самого початку, мова програмування С# була спроектована для роботи з .NET. Мова програмування С# підходить не тільки для написання мережевих і web-додатків, а й для створення ігор на ПК, для написання кросплатформених мобільних додатків та інше.

Зовні С# дуже схожий з Java і С ++, але вивчити його набагато легше. Використання С# допомагає зі швидким стартом проєкту і отриманням робочого прототипу.

До можливостей С# відносяться: повна підтримка класів і ООП (Об'єктно орієнтоване програмування); погоджений і чітко визначений набір базових типів; підтримка автоматичної генерації XML-документації; підтримка властивостей і подій в стилі Visual Basic; Можливість використання для написання динамічних web-сторінок ASP.NET і web-служб XML [26].

Microsoft Visual Studio - це інтегроване середовище розробки (IDE) від Microsoft. Використовується для розробки комп'ютерних програм, а також веб-сайтів, веб-додатків, веб-сервісів і мобільних додатків. Visual Studio використовує платформи розробки програмного забезпечення Microsoft, такі як Windows API, Windows Forms, Windows Presentation Foundation, Windows Store і Microsoft Silverlight. Він, звісно, може створювати як власний код, так і керований код.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

У документації [27] Visual Studio включає редактор коду, що підтримує IntelliSense, а також рефакторинг коду. Вбудовані інструменти включають профілювальник коду, конструктор для створення додатків з графічним інтерфейсом, веб-дизайнер, конструктор класів і конструктор схем бази даних. Він приймає плагіни, які розширюють функціональність майже на всіх рівнях, включаючи додавання підтримуючих систем управління версіями (таких як Subversion і Git) і додавання нових наборів інструментів, таких як редактори та візуальні дизайнери, для мов, специфічних для предметної області, або наборів інструментів для інших аспектів розробки програмного забезпечення.

Visual Studio підтримує 36 різних мов програмування і дозволяє редактору коду підтримувати практично будь-яку мову програмування за умови, що існує служба для конкретної мови. Вбудовані мови включають C, C ++, C ++ / CLI, Visual Basic .NET, C #, F #, JavaScript, TypeScript, XML, XSLT, HTML і CSS. Підтримка інших мов, таких як Python, Ruby, Node.js і інших, доступна через плагіни.

Сама базова версія Visual Studio, версія Community, доступна безкоштовно. Слоган Visual Studio Community edition: «Безкоштовна повнофункціональне середовище IDE для студентів, розробників з відкритим вихідним кодом і індивідуальних розробників».

Visual Studio пропонується в трьох версіях:

- Visual Studio Community – безкоштовна версія IDE для студентів і незалежних розробників.
- Visual Studio Professional – платна версія з професійними інструментами для розробки для малих команд. Вартість: 45\$/місяць.

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

***ІАЛЦ.467800.003 ПЗ***

*Арк.*

48

- Visual Studio Enterprise – платна версія з професійними інструментами для продуктивної роботи та координації команд розробників будь-якого розміру. Вартість: 250\$/місяць.

Для даного проєкту буде використовуватись Visual Studio 2019 Community для Mac, версії 8.9.9, оскільки даної безкоштовної версії досить для створення комп'ютерної гри.

Unity дозволяє створювати свої компоненти, використовуючи скрипти. За допомогою цих компонентів контролюється поведінка ігрових об'єктів. Скрипти дозволяють активувати ігрові події, змінювати параметри компонентів, і відповідати на введення користувача яким завгодно способом. Розберемо основні бібліотеки (рис 3.1) які використано при написанні скрипта:

- System.Collections
  - Містить інтерфейси і класи, які визначають різні колекції об'єктів, такі як списки, черги, бітові масиви, хеш-таблиці і словники;
- System.Collections.Generic Namespace

Містить інтерфейси і класи, які визначають загальні колекції, які дозволяють користувачам створювати строго типізовані колекції, які забезпечують кращу безпеку типів і продуктивність, ніж не універсальні строго типізовані колекції.

- UnityEngine – це підключення ігрового рушія та зв'язок з ним.
- UnityEngine.SceneManagement – дозволяє створювати рівні, та переходи між ними.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;

```

Рис 3.1 – Фрагмент коду

Скрипт взаємодіє з внутрішніми механізмами Unity шляхом створення класу, успадкованих від вбудованого класу, званого MonoBehaviour.

Розберемо класи які в нас використовуються:

- **OnTriggerEnter.** Коли GameObject взаємодіє з іншим GameObject, Unity викликає OnTriggerEnter;
- **FixedUpdate.** Фаза поновлення у власному циклі плеєра. Це уявлення C# фази поновлення у власному циклі програвача.
- **OnTriggerExit.** Викликається, коли інший колайдер перестає стосуватися тригера.

Фінальна версія гри зображена на рисунку 3.2:

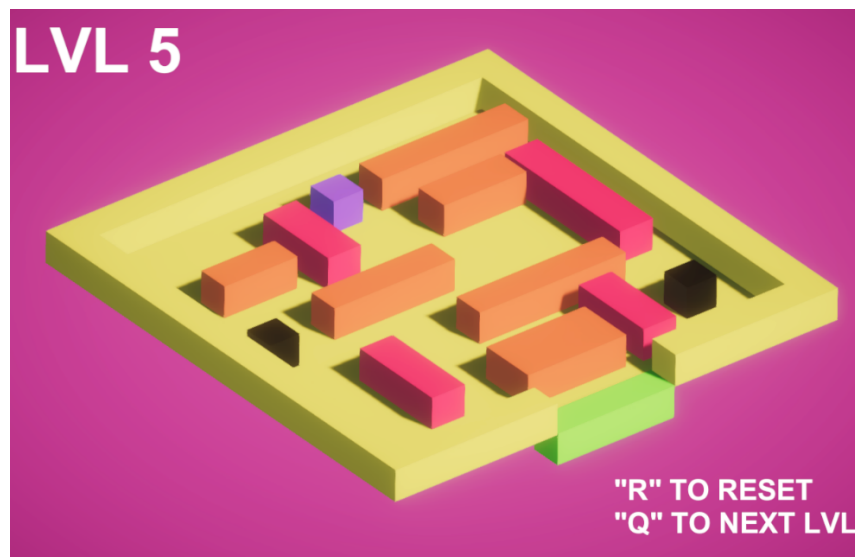


Рис. 3.2 – Фінальна версія гри

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

## 3.2 Створення гри

Для початку нам потрібно зробити перший об'єкт (рис. 3.3).

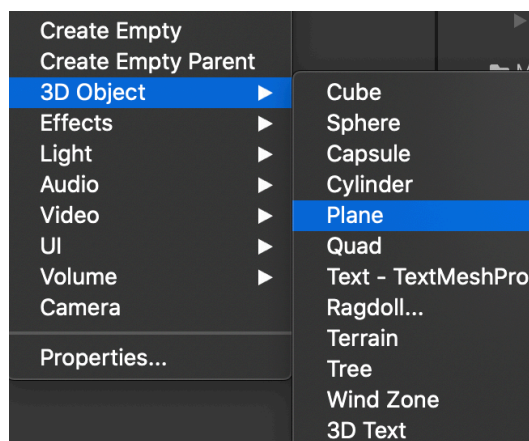


Рис. 3.3 – Створення об'єкту

Натискаючи на праву кнопку миші 3D Object->Plane з'явився перший об'єкт в просторі "Scene". Це буде територія на якій будуть відбуватися всі процеси нашої гри. На рисунку 3.4 зображений перший об'єкт:

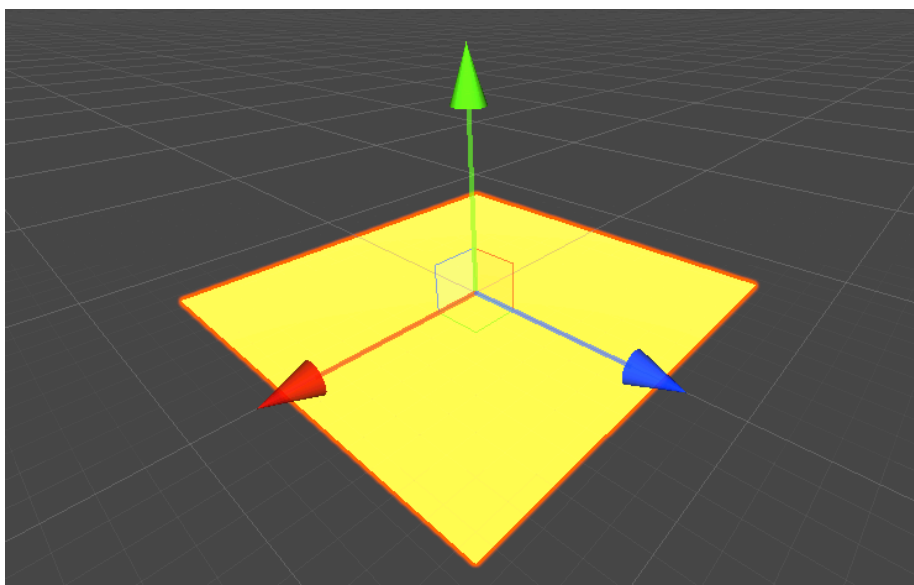


Рис. 3.4 – Територія гри

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

51

Потрібно закінчити платформу, а саме створити об'єкти, які будуть перешкодами для гравця. Ці об'єкти дозволяють залишати гравця на території "Plane". Створюємо об'єкт "Cube" (рис 3.5).

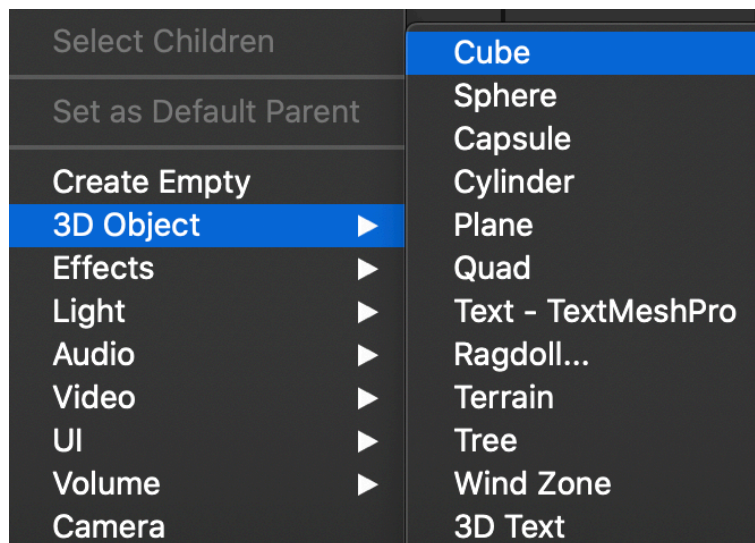


Рис 3.5 – Створення об'єкту куб

Довелося створити 7 кубів, для завершення платформи, які розставлені по периметру (рис. 3.6).

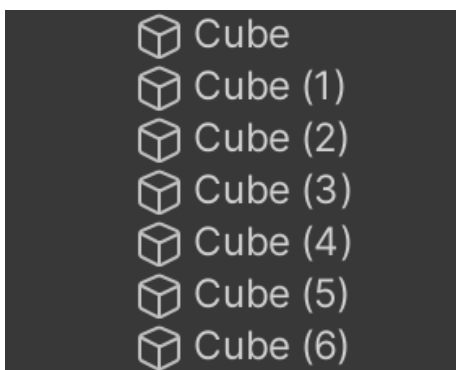


Рис 3.6 – Список об'єктів

Вигляд платформи в GameSpace зображений на рис 3.7:

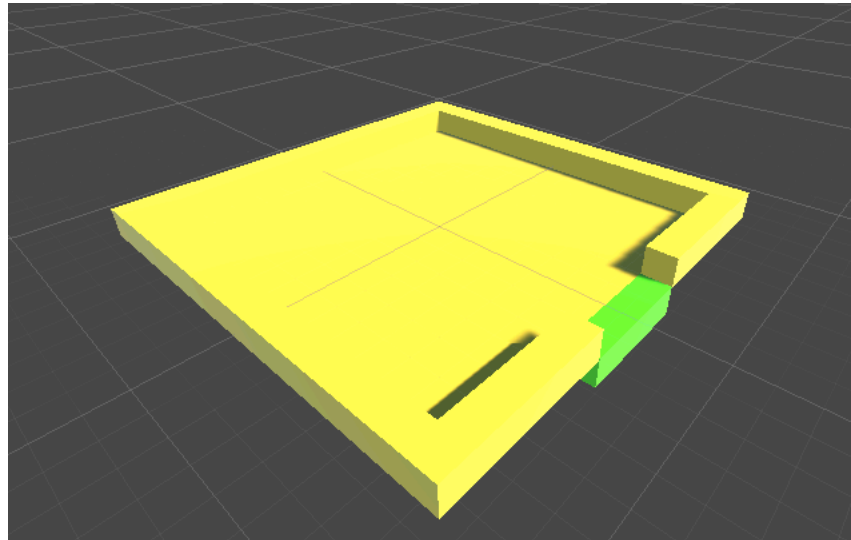


Рис 3.7 – Поле для гри

Тепер потрібно створити перші ігрові об'єкти, які будуть основними в нашій грі. Почнемо з головного об'єкта яким буде керувати гравець. Також створюємо “Cube” (рис 3.8).

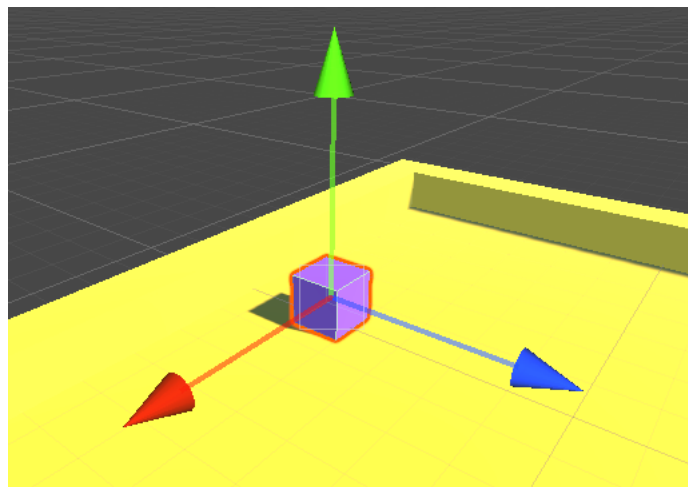


Рис. 3.8 – Об'єкт куб

Було створено звичайний “Cube” іншого кольору методом додавання матеріалу, та був написаний скрипт, зображений на рисунку 3.9б який буде дозволяти цьому об'єкту рухатися.

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

**ІАЛЦ.467800.003 ПЗ**

Арк.

53

```

5
6 public class Player : MonoBehaviour
7 {
8     [SerializeField] KeyCode keyOne;
9     [SerializeField] KeyCode keyTwo;
10    [SerializeField] Vector3 moveDirection;
11
12    private void FixedUpdate()
13    {
14        if(Input.GetKey(keyOne))
15        {
16            GetComponent<Rigidbody>().velocity += moveDirection;
17        }
18        if (Input.GetKey(keyTwo))
19        {
20            GetComponent<Rigidbody>().velocity -= moveDirection;
21        }

```

Рис. 3.9 – Фрагмент коду

Так як потрібно, щоб наш об’єкт рухався в двох напрямках, вгору - вниз чи вліво – вправо, потрібно задати дві клавіші на клавіатурі, при натисканні яких гравець зможе управляти об’єктом. Щоб вказати клавішу ми використовуємо KeyCode. Для того, щоб в Inspector Unity була можливість задати ці клавіші в будь-який момент, потрібно перед KeyCode написати [SerializeField]. В 10 рядку використовуємо Vector3 MoveDirection; щоб вказати напрямок об’єкта, також в початку потрібно написати [SerializeField], щоб корегувати все в самому інтерфейсі Unity.

В наступному коді (рис 3.10) написано скрипт, щоб об’єкт рухався. Створюємо умову, якщо в нас нажата перша клавіша (KeyOne) то до швидкості нашого куба ми будемо додавати напрямлення (vector3).

```

12    private void FixedUpdate()
13    {
14        if(Input.GetKey(keyOne))
15        {
16            GetComponent<Rigidbody>().velocity += moveDirection;
17        }

```

Рис. 3.10 – Фрагмент коду

Той самий код написано для другої клавіші, тільки з іншим ім’ям та “+”

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

замінено на “-”, щоб направлення віднімалося, і об’єкт рухався в іншу сторону (рис. 3.11).

```
18         if (Input.GetKey(keyTwo))
19         {
20             GetComponent<Rigidbody>().velocity -= moveDirection;
21         }
```

Рис. 3.11 – Фрагмент коду

В Unity, в Inspector до “Cube” який створено в минулому кроці, потрібно додавати Component коду і в наступних параметрах, які зображено на рисунку 3.12, необхідно вказати дві клавіші, при натисканні яких об’єкт буде рухатися. Було обрано першу клавішу – “W”, а другу – “S”.

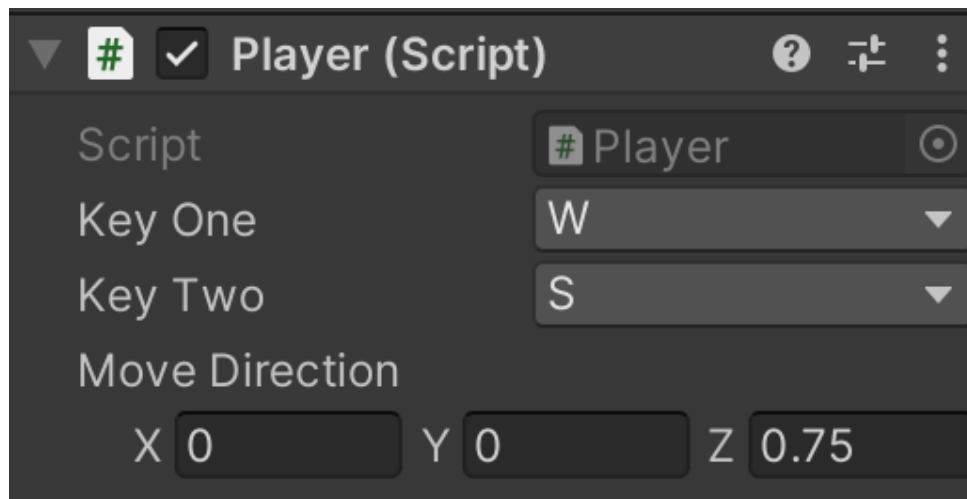


Рис. 3.12 – Налаштування клавіш

В MoveDirection потрібно вказати вісь (X, Y, Z) по якій об’єкт буде пересуватися, вказуємо 0.75 в останній комірці, це означає що об’єкт буде рухатися по осі Z, збільшуючи значення, швидкість об’єкта буде збільшуватися. Вісь Y змінювати не потрібно, так як це рух вгору-вниз. Потрібно створити ще один об’єкт (рис. 3.13) яким гравець зможе керувати, та він буде рухатися, але вже по іншій осі.

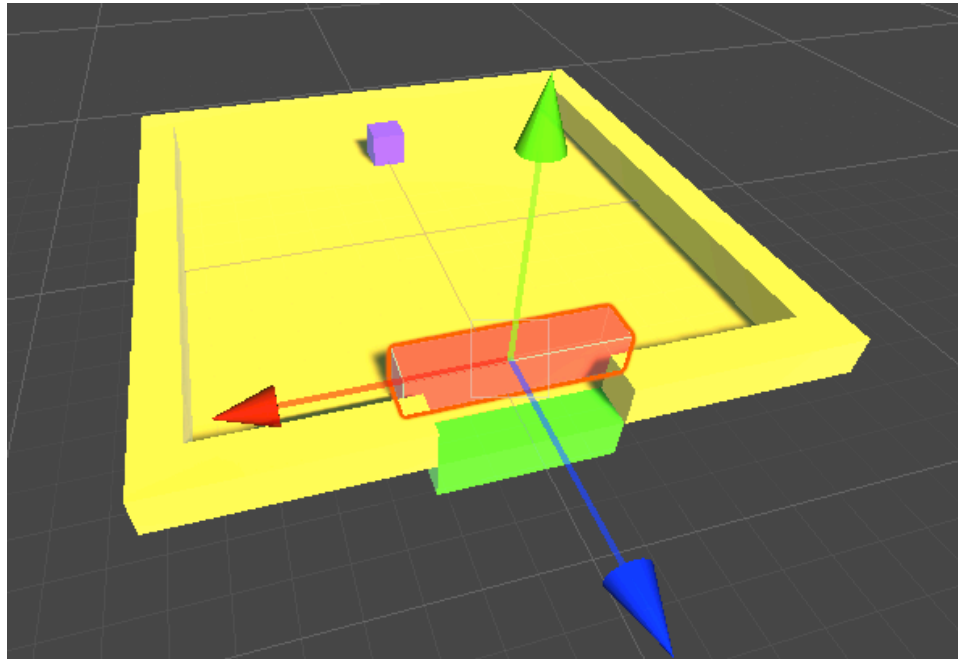


Рис 3.13 – Ігрове поле

Для цього об'єкта було обрано інші клавіші: перша – “A”, друга - ”D”, при натисканні яких він буде рухатися, але вже по другій осі, вибравши вісь X потрібно вказати те ж значення в MoveDirection тільки в комірці X (рис. 3.14).



Рис 3.14 – Налаштування клавіш

Тепер потрібно зробити кінець гри/рівня, для цього створюємо ще один об'єкт CUBE, який робимо непрозорим. Він буде грати роль того самого об'єкта при взаємодії з яким гра закінчиться або перейде на наступний рівень (рис. 3.15).



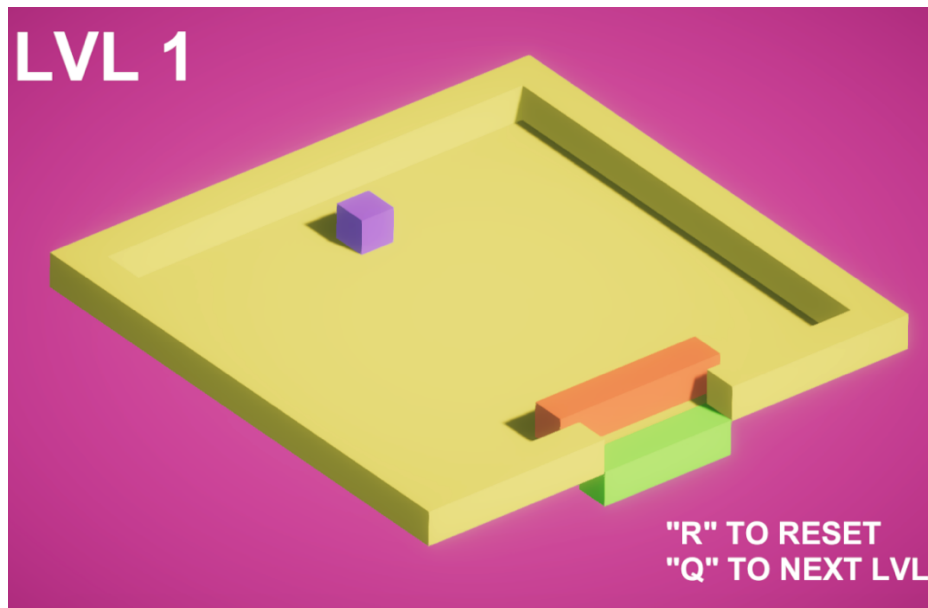


Рис. 3.17 – Рівень гри

Було вирішено додати зверху зліва номер рівня, а внизу справа – дві клавіші: одну для того, щоб почати рівень з самого початку (клавіша “R”), а іншу, щоб перейти на наступний рівень (клавіша “Q”). Для того, щоб при натисканні на клавішу R відбувався перезапуск рівня, налаштовано наступну умову в якій при натисканні клавіші сцена буде перезапускатися (рис. 3.18).

```

22         if(Input.GetKey(KeyCode.R))
23         {
24             SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
25         }

```

Рис. 3.18 – Фрагмент коду

І те саме робимо з клавішею Q, при натисканні на яку гравець буде переходити на наступний рівень (рис. 3.19).

```

26         if (Input.GetKey(KeyCode.Q))
27         {
28             SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1);
29         }
30     }

```

Рис. 3.19 – Фрагмент коду

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

58

Залишилося створити різні рівні і зробити гру цікавішою та різноманітнішою. В наступному рівні, виконуючи задачу по урізноманітненню гри, з'являється новий об'єкт, зображений на рисунку 3.20, яким гравець не може керувати, та сам об'єкт не може рухатися взагалі, він буде виконувати роль перешкоди. Створюємо звичайний "Cube" чорного кольору, тим самим позначаємо тип об'єкту.

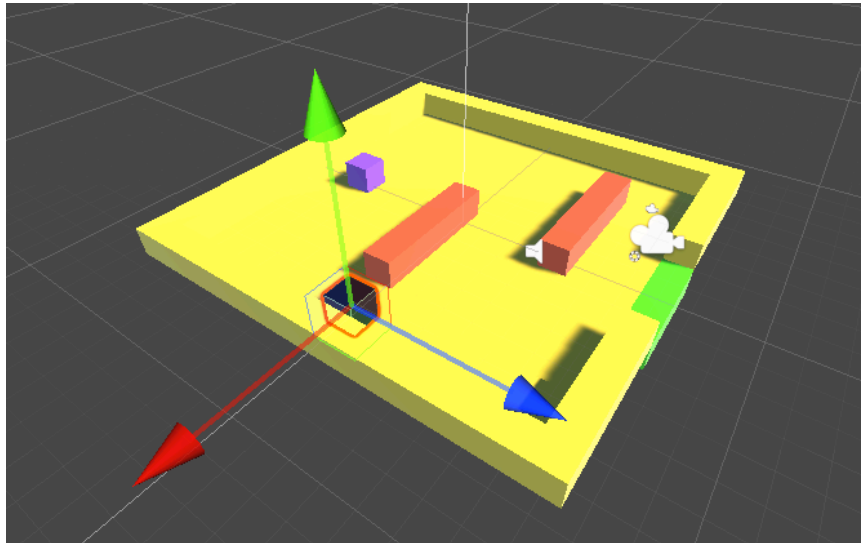


Рис. 3.20 – Ігрове поле

В MoveDirection для цього об'єкта ми встановлюємо значення 0 на всіх осях, щоб гравець не міг керувати цим об'єктом (рис. 3.21).



Рис. 3.21 – Налаштування управління

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

59

Також потрібно заблокувати осі (X, Z) для цього об'єкта, щоб при взаємодії з іншими об'єктами він не рухався (рис. 3.22). Заморожуємо дві основні осі це X та Z, вісь Y не обов'язково блокувати, так як це рух ввєрх та вниз.

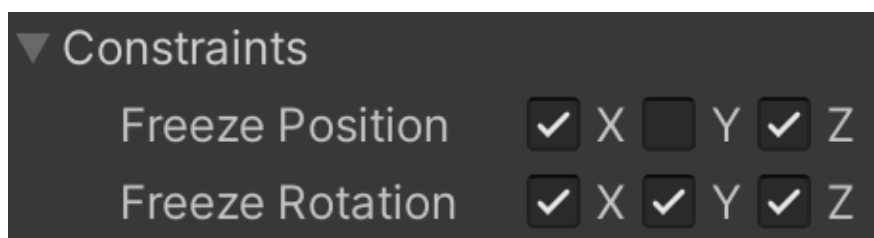


Рис. 3.22 – Налаштування осей

В одному з рівнів з'являється нова механіка з групою прозорих та непрозорих блоків, які при натисканні на новий об'єкт – кнопку, міняють другий тип об'єктів з прозорого на непрозорий і навпаки (рис. 3.23).

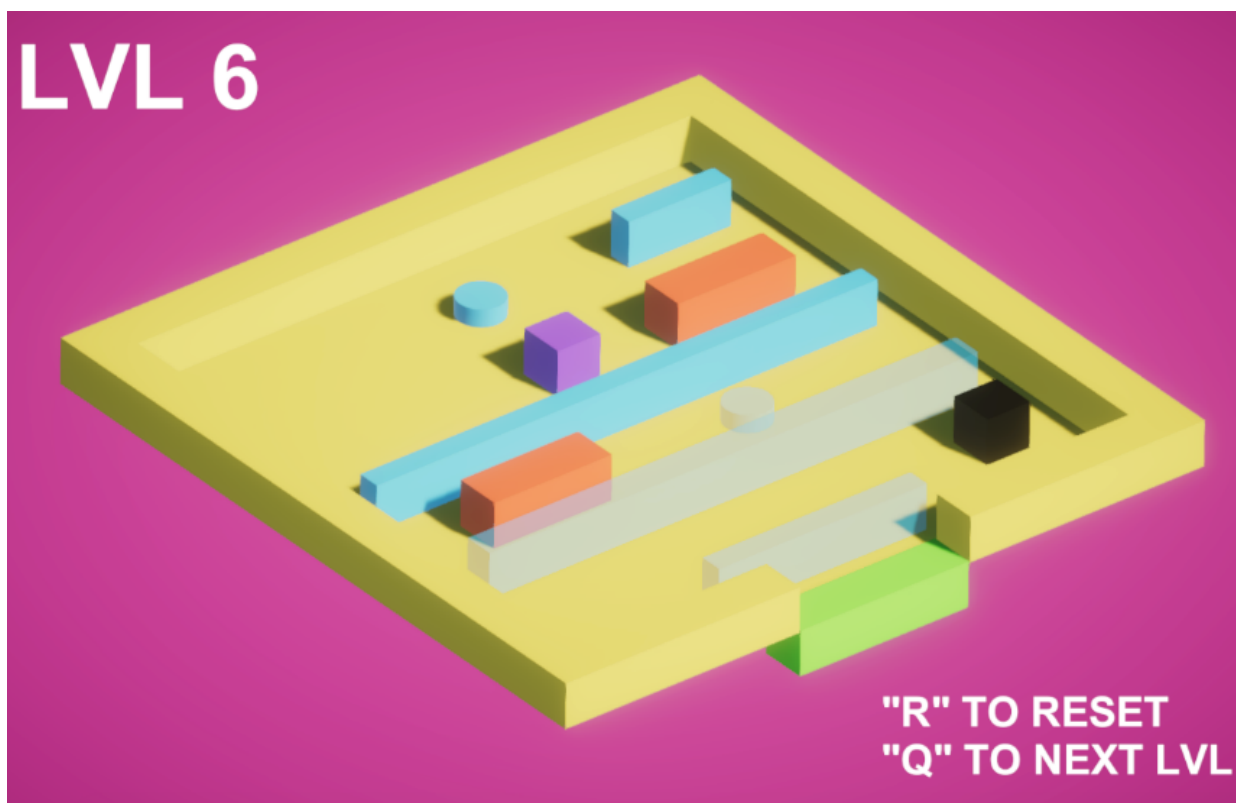


Рис. 3.23 – Рівень гри

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

60

В Visual Studio створюємо потрібну функцію (рис. 3.24).

```
5     public class Activator : MonoBehaviour
6     {
7         public GameObject[] firstGroup;
8         public GameObject[] secondGroup;
9         public Activator button;
10        public Material normal;
11        public Material transparent;
12        public bool canPush;
13    }
```

Рис. 3.24 – Фрагмент коду

Прописуємо два масиви (групи об’єктів), вказуємо кнопку та матеріали. Далі створюємо функцію `private void OnTriggerEnter(Collider other`, рядок 14) в якій робимо умову коли “Cube” або “Player” торкаються кнопки прописуємо цикл `foreach`, тобто для кожного об’єкта в певній групі, щоб відбувалася зміна матеріалу з прозорого на непрозорий (рис. 3.25).

```
14     private void OnTriggerEnter(Collider other)
15     {
16         if (canPush)
17         {
18             if (other.CompareTag("Cube") || other.CompareTag("Player"))
19             {
20                 foreach (GameObject first in firstGroup)
21                 {
22                     first.GetComponent<Renderer>().material = normal;
23                     first.GetComponent<Collider>().isTrigger = false;
24                 }
25                 foreach (GameObject second in secondGroup)
26                 {
27                     second.GetComponent<Renderer>().material = transparent;
28                     second.GetComponent<Collider>().isTrigger = true;
29                 }
30                 GetComponent<Renderer>().material = transparent;
31                 button.GetComponent<Renderer>().material = normal;
32                 button.canPush = true;
33             }
34     }
```

Рис. 3.25 – Фрагмент коду

Також створено перемінну `CanPush`, аби не допустити помилок, а саме щоб

блок не міг застрягти. Вона буде регулювати, щоб кнопка натискалася тільки тоді, коли ніякий блок ніде не застряє. Доповнюючи функцію `private void OnTriggerExit(Collider other)`, де вказуємо умову: якщо один куб доторкається до іншого куба, то ні одна кнопка не може бути активована (рис. 3.26).

```
45     private void OnTriggerExit(Collider other)
46     {
47         if (this.CompareTag("Cube") && other.CompareTag("Cube"))
48         {
49             foreach (Activator button in FindObjectsOfType<Activator>())
50             {
51                 button.canPush = true;
52             }
53         }
54     }
55 }
56
```

Рис 3.26 – Фрагмент коду

<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>

***ІАЛЦ.467800.003 ПЗ***

**Арк.**

**62**

### ВИСНОВКИ ДО РОЗДІЛУ 3

В даному розділі було описано основні інструменти, які було використано при написанні нашої гри. Було реалізовано гру, за допомогою взаємодії ігрового рушія Unity та середовища для програмування – Microsoft Visual Studio. Було створено елементи та механіки, завдяки яким гравцеві цікавий кожен новий рівень. Шляхом зручних інструментів та ідеї проєкту, гру можна модифікувати за короткий проміжок часу, додаючи нові елементи, нові рівні та нові механіки. Були оцінені та виправлені можливі помилки, які могли відбутися під час геймплею: уникнута можливість застрягання елементів один в одному.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	Арк.
						63
<b><i>Зм.</i></b>	<b><i>Арк.</i></b>	<b><i>№ докум.</i></b>	<b><i>Підпис</i></b>	<b><i>Дата</i></b>		

## РОЗДІЛ 4

### 4.1 Розробка саунд-дизайну

В другому розділі в пунктах 2.3.1-2.3.3 виконано аналіз платформ музикального продюсування, і в висновку до другого розділу було визначено платформу, в якій буде реалізований саундтрек до гри, а саме в Ableton Live. Важливо зазначити основу перевагу вибраної платформи, це те що Ableton Live дає можливість написання проєктів для різних задач, не вимагаючи для цього якихось особливих знань. Інтерфейс зображено на рисунку 4.1.

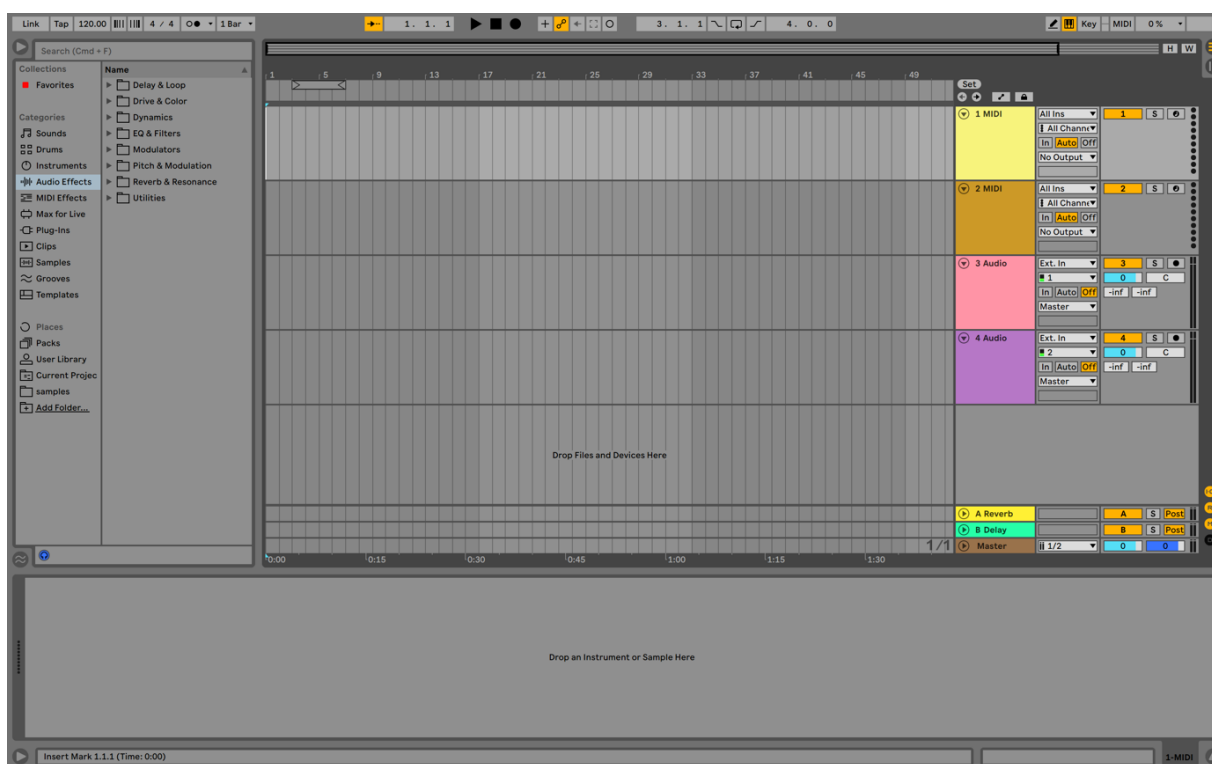


Рис. 4.1 – Інтерфейс програми

Ідея – зробити простий, за кількістю інструментів та звуків, з елементами попкультури 80х, саундтрек, щоб він не відвертав гравця від гри. Почати довелося зі швидкості нашої роботи, так званий BPM (beats per minutes). Взагалі в сучасній музиці BPM використовується в діапазоні 60-180

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

64

одиниць, в проєкті було поставлено значення, яке дорівнює 120 (рис. 4.2).



Рис. 4.2 – Налаштування ритму

Почнемо реалізовувати проєкт з написання драм-партії, з механікою непрямої бочки. В цьому допоможе плагін Battery, в який ми помістимо семпли (звуки) та зробимо перше звучання. В полі пошуку вкладки Plug-ins вписуємо назву нашого плагіну (рис. 4.3).

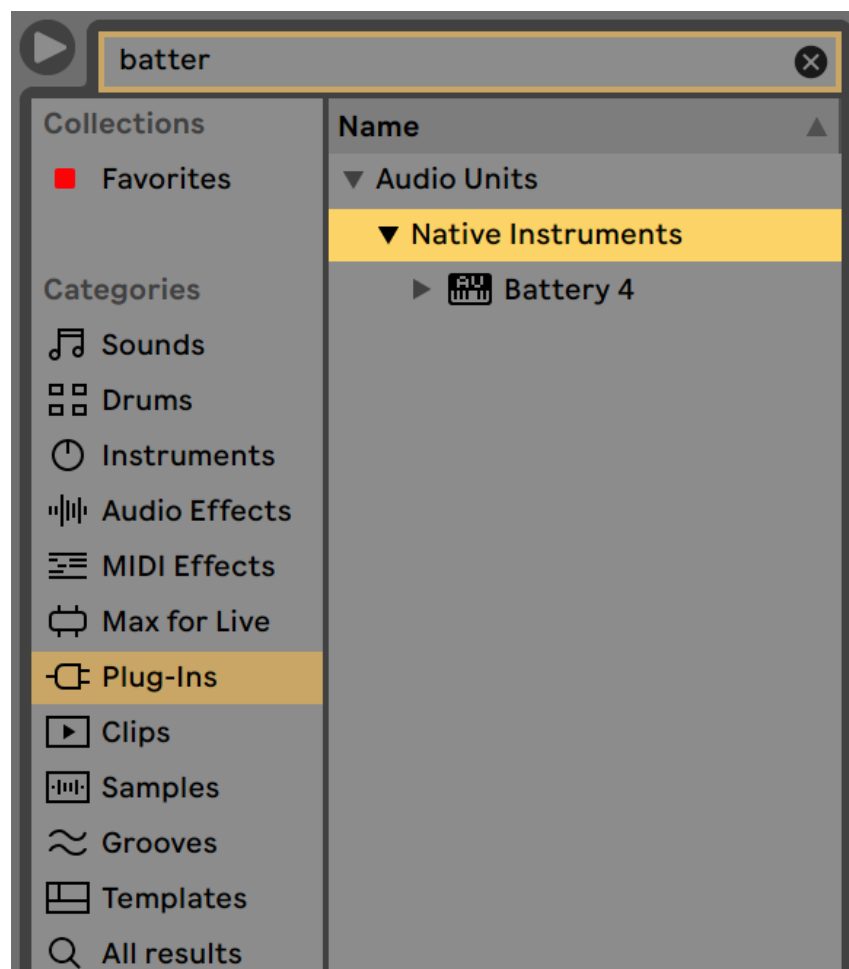


Рис. 4.3 – Підключення плагіну

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

65

Потім створивши вже MIDI-канал в основному полі, потрібно перенести на нього наш плагін. Вигляд середовища зображений на рисунку 4.4.



Рис. 4.4 – Інтерфейс

Тепер потрібно знайти основні звуки, які будуть використовуватися для створення ритму саундтреку. Виходячи з ідеї, довелося обрати класичні Kick та Clap, а ще додати перкусії та hi-hats (рис. 4.5).



Рис. 4.5 – Написання музики

<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>

**ІАЛЦ.467800.003 ПЗ**

**Арк.**

**66**

В Piano Roll створено драм партію, кожний рядок з якої відповідає за відповідний sample (рис. 4.6).

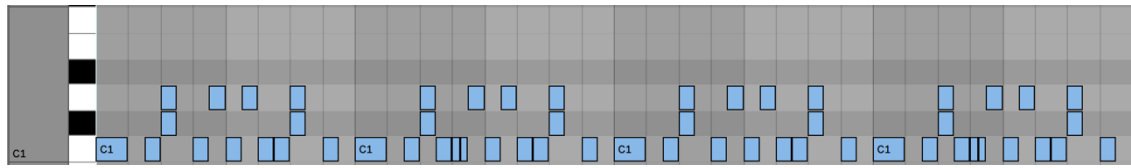


Рис. 4.6 – Драм партія

Наступним кроком, потрібно обробити той звук який ми отримали, використовуючи інші плагіни та ефекти, які зображені на рисунку 4.7.

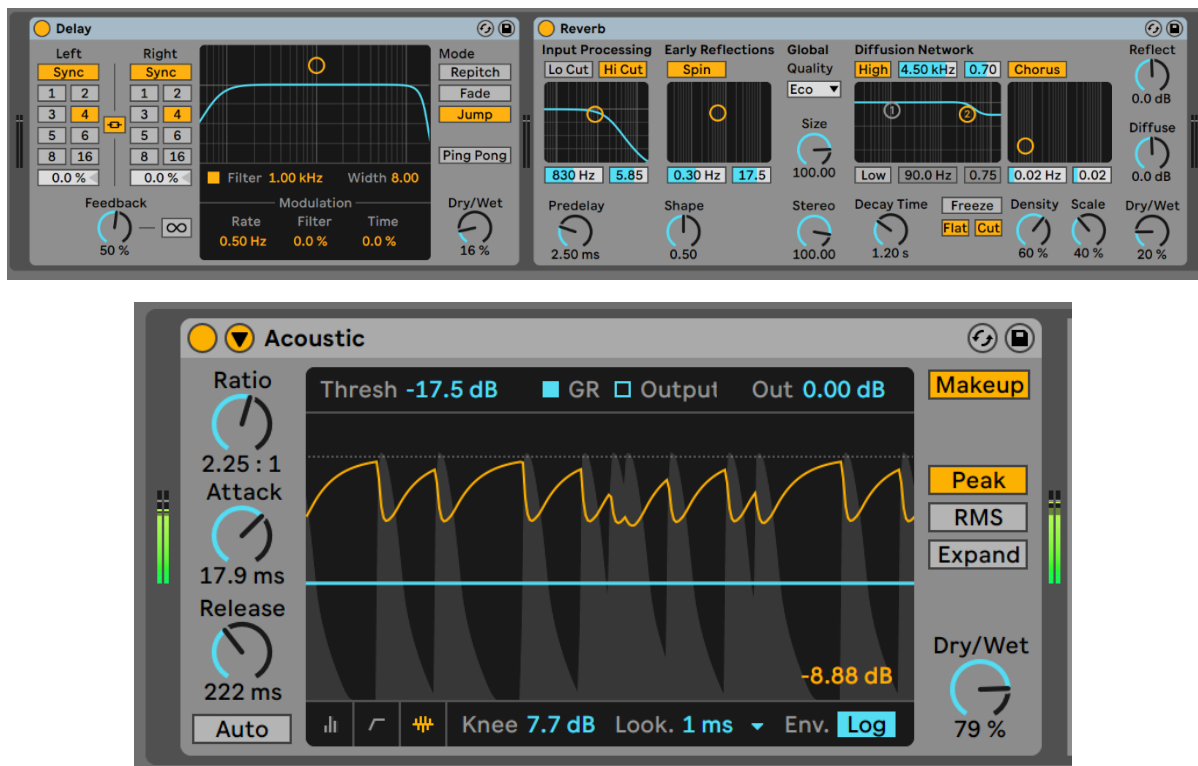


Рис 4.7 – Налаштування ефектів

Переходимо до основної мелодії, та інструменту який буде чути кожен гравець. Так як ідея була взяти деякі елементи з попкультури минулого століття, то буде використано плагін, а саме синтезатор Serum (рис. 4.8), в якому буде вибрано інструмент та написано основну мелодію.

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

67



Рис 4.8 – Інтерфейс налаштування

Створивши звук, переходимо до написання мелодії в Piano Roll (рис. 4.9).  
 Буде використана невелика кількість нот, для того, щоб дотримуватися  
 концепції простого треку.

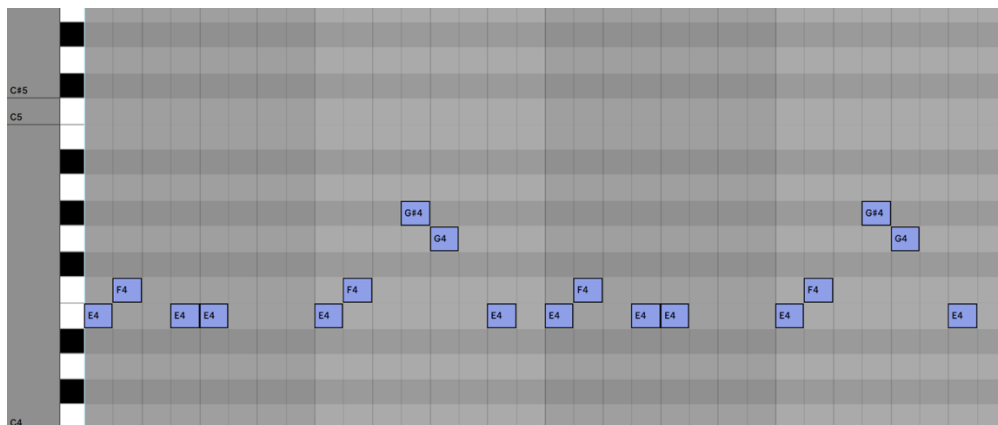


Рис 4.9. – Фрагмент мелодії

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

68

Після написання основної мелодії, вирішено написати ще одну партію, для того, щоб робота була більш різноманітною (рис. 4.10).

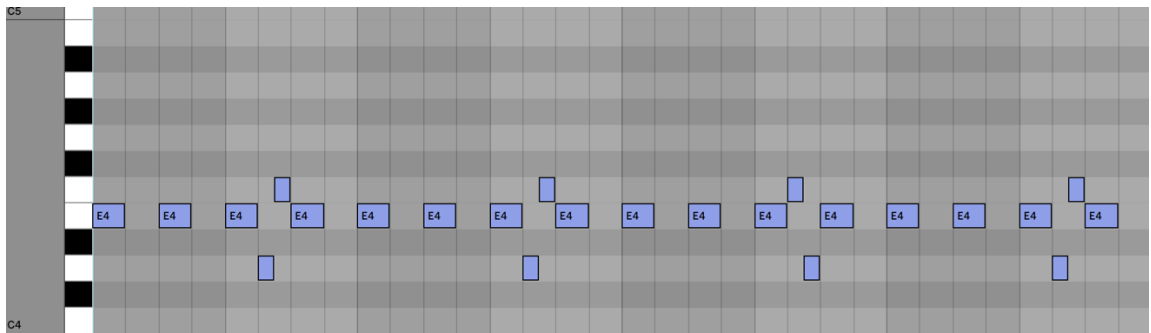


Рис. 4.10 – Фрагмент мелодії

Як і в минулому етапі, цей елемент потрібно обробити. В даному випадку оброблено все в самому плагіні Serum (рис. 4.11).



Рис. 4.11 – Налаштування в плагіні

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

69

Завершивши цей крок, було додано ще декілька доріжок, основні з яких, бас низькочастотний та простий, доповнюючи основну мелодію звук. Оброблено фінально всі доріжки, проведена робота над концептом і вигляд фінальної версії музикального проєкту зображений на рисунку 4.12.

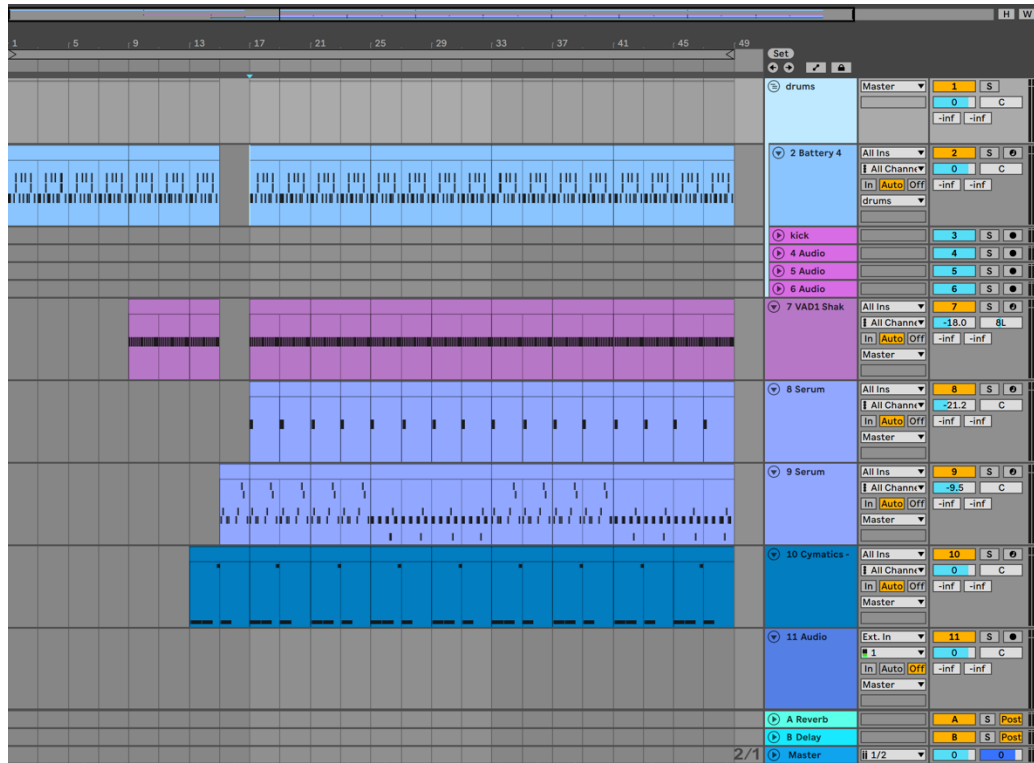


Рис. 4.12 – Готова версія мелодії

Експортуємо проєкт, та в Unity, додаємо Audio Source (рис. 4.13).

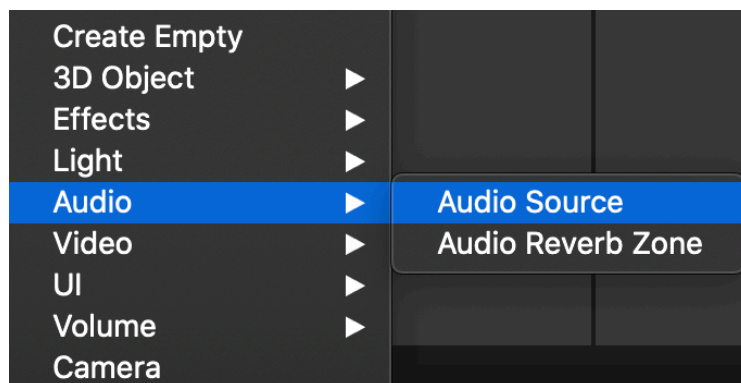


Рис. 4.13 – створення Audio Source

Вже в Inspector-i, в розділі AudioClip вибираємо файл нашого саундтреку, який було експортовано, залишилося налаштувати і все готово (рис. 4.14). Тепер в нас є саундтрек, який під час гри буде слухати кожен гравець.

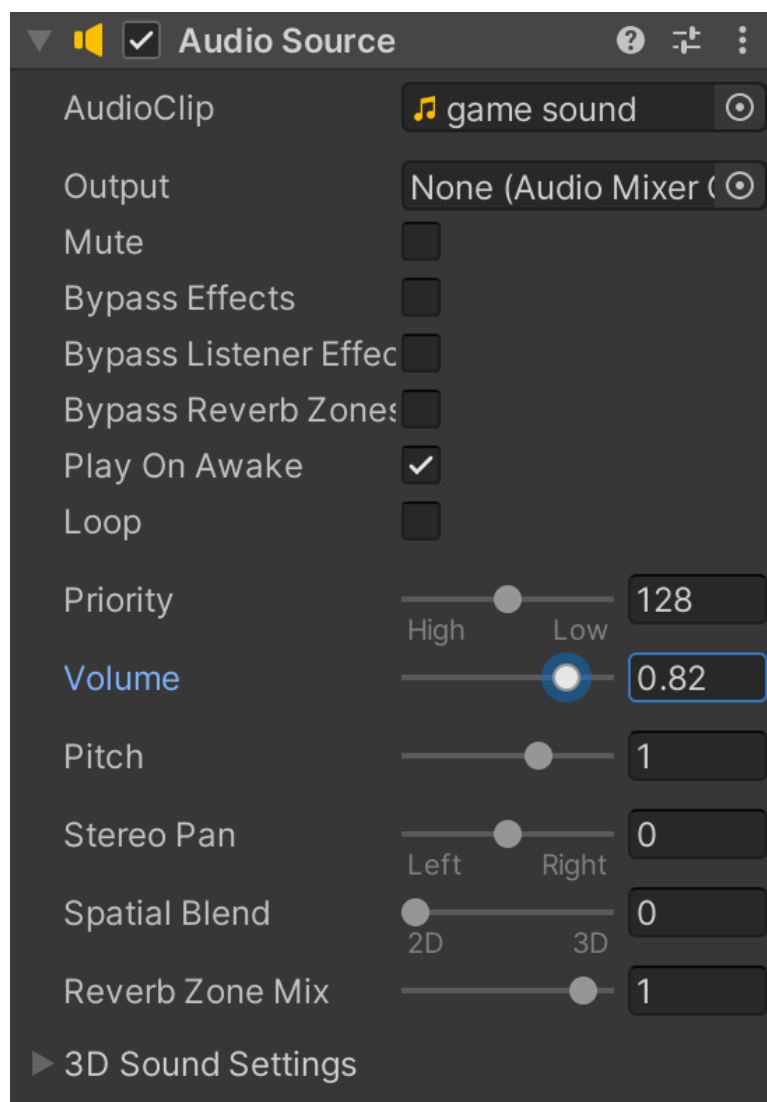


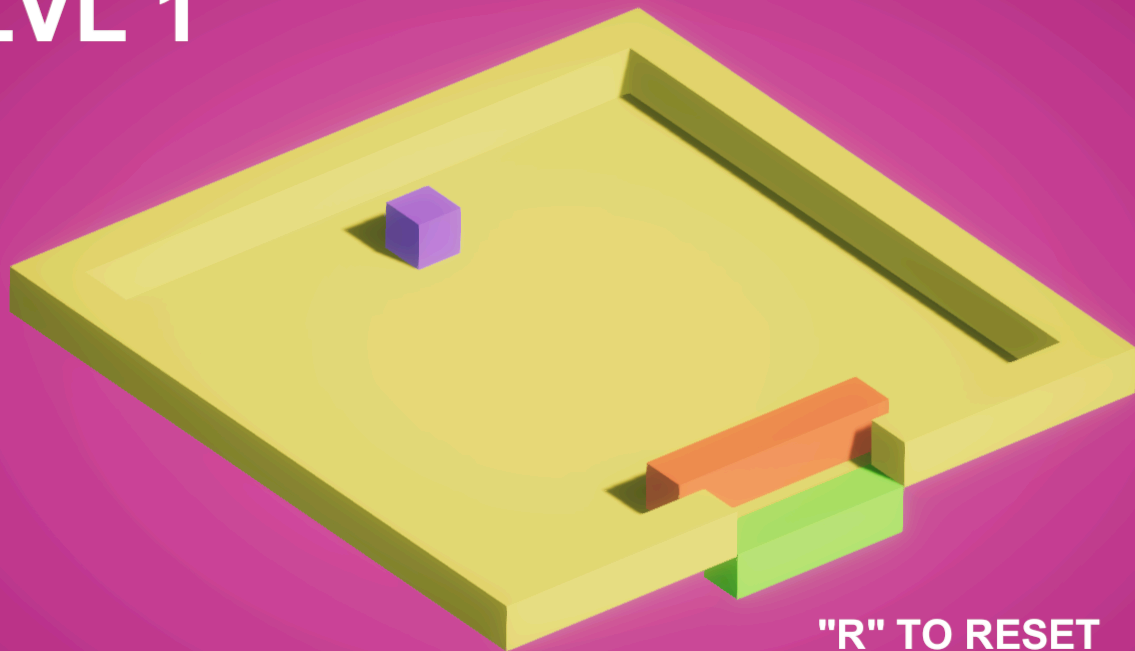
Рис. 4.14 – Налаштування мелодії

## 4.2 Фінальний вигляд гри

В 3-му та в частині 4 розділу було реалізовано нашу задачу, було створено комп'ютерну 3D гру та написано саундтрек до цієї гри, результат зображений на рисунку 4.15.

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дата		

# LVL 1



"R" TO RESET  
"Q" TO NEXT LVL

Рис 4.15 – Перший рівень гри

Це перший рівень гри, який вже було продемонстровано в минулому розділі, він потрібний для того, щоб гравець розібрався з задачею гри та її механікою. Основний об'єкт, яким керує гравець є фіолетовий куб, задачею якого є дійти від своєї точки до зеленої лінії. Як тільки гравець перетне зелену лінію гра перейде на наступний рівень. Щоб змусити куб рухатися гравець має використовувати клавіші "W" та "S" для руху вперед та назад. Також під керівництвом гравця є оранжевий паралелепіпед, для руху якого гравцю потрібно натискати клавіші "A" та "D". Щоб пройти перший рівень гравцю достатньо зрушити помаранчевий куб та фіолетовим кубом перетнути лінію, як зображено на рисунку 4.16.

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

72

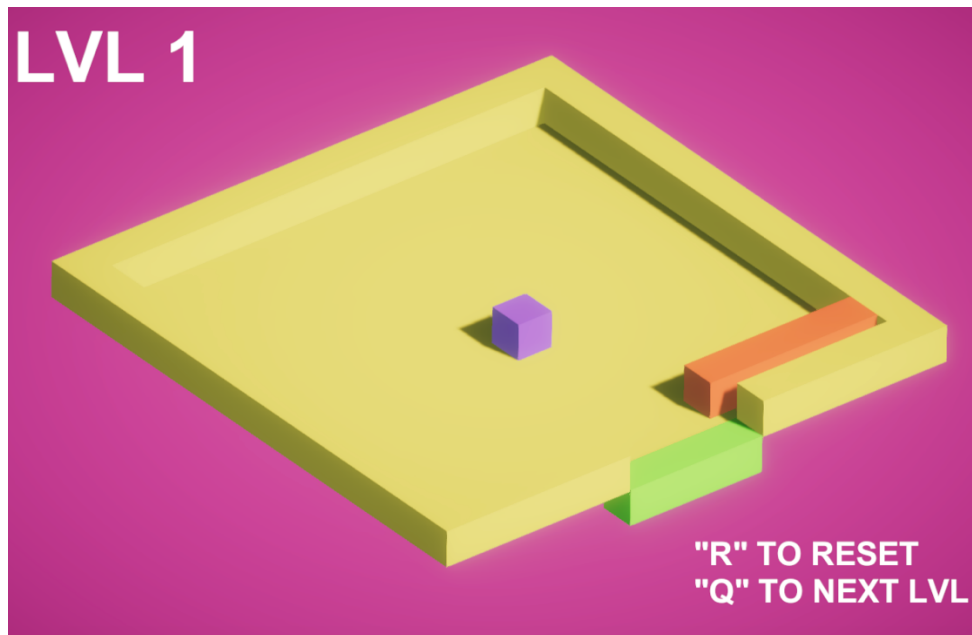


Рис 4.16 – Переміщення об’єкту

Як тільки гравець це зробить він перейде на наступний рівень та вже буде ознайомлений з механікою гри та самим процесом. Також гравець може натиснути “R” та відбудеться перезапуск рівня, або натиснути “Q” та відразу перейти на наступний рівень. В грі реалізовано 7 різних рівнів, але в будь-який час гру можна модифікувати та доповнити.

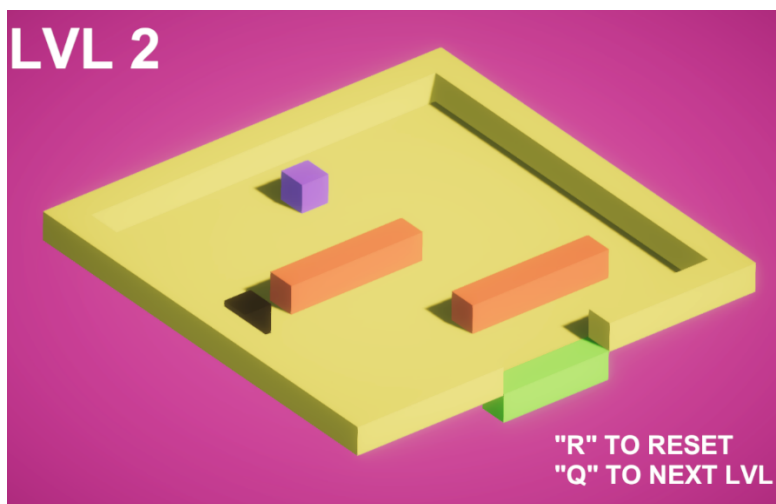


Рис 4.17 – Другий рівень гри

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

73

В другому рівні (рис. 4.17) додається два нових об'єкти, механіка одного з них ще невідома гравцю, чорний куб виступає звичайною не рушійною перешкодою.

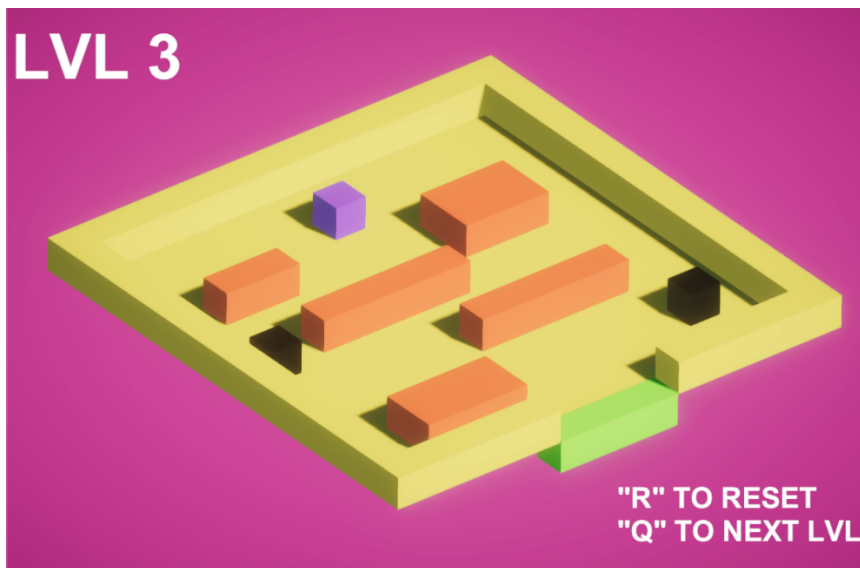


Рис. 4.18 – Третій рівень гри

Третій рівень (рис. 4.18) є модифікованою версією другого рівня. В нього додано декілька нових об'єктів з якими вже знайомий гравець.

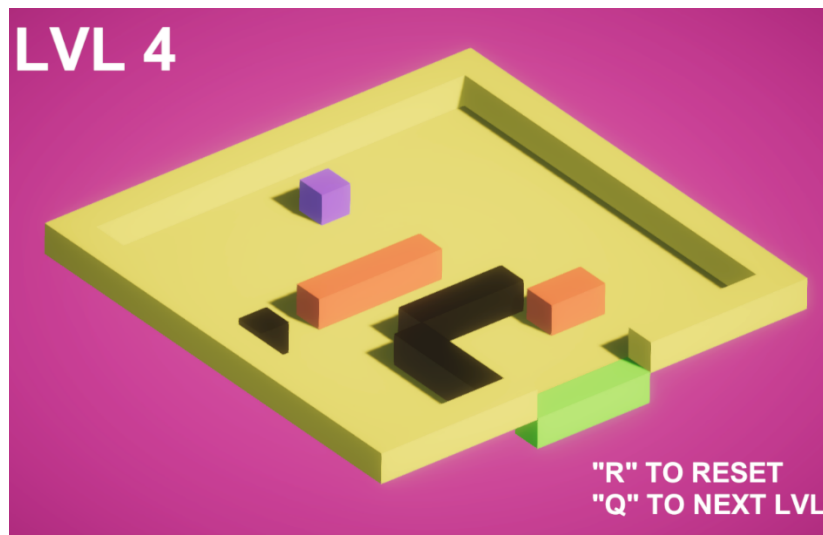


Рис. 4.19 – Четвертий рівень гри

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

**ІАЛЦ.467800.003 ПЗ**

Арк.

74

Четвертий рівень (рис. 4.19) не обладнаний новими механіками. Пройти цей рівень можливо шляхом взаємодії об'єктів, в даному випадку гравець повинен допомогти собі помаранчевими кубами, пересовуючи фіолетовий куб по осі X.

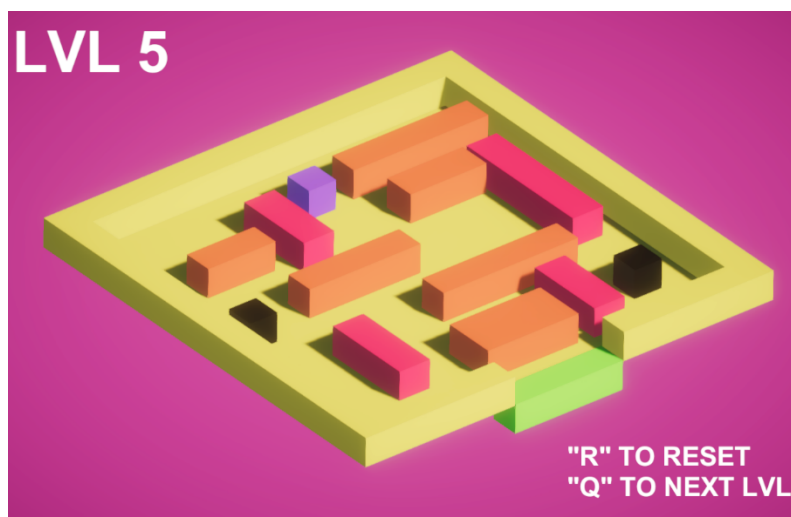


Рис. 4.20 – П'ятий рівень гри

П'ятий рівень (рис. 4.20) наповнений великою кількістю об'єктів-перешкод, які рухаються по осям Z та X, та нерухомими чорними кубами.

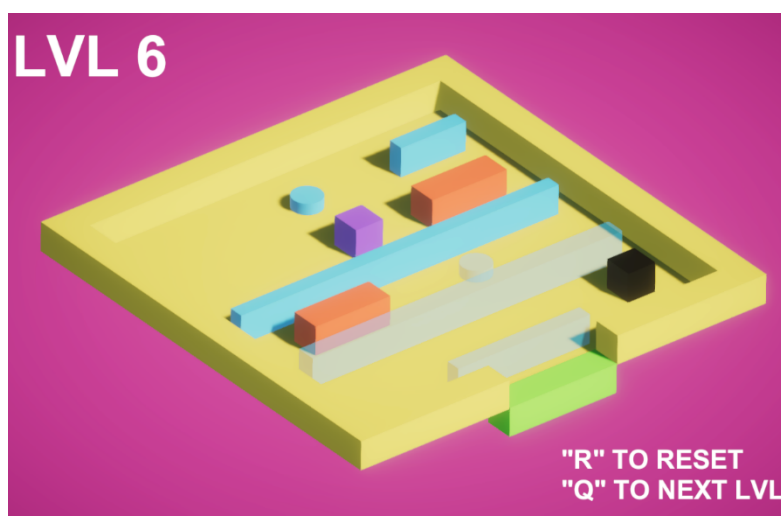


Рис. 4.21 – Шостий рівень гри

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

75

Шостий рівень, який зображено на рисунку 4.21, має нову механіку, в якій гравцю потрібно взаємодіяти з кнопками, щоб міняти сині об'єкти з прозорих на непрозорі та навпаки.

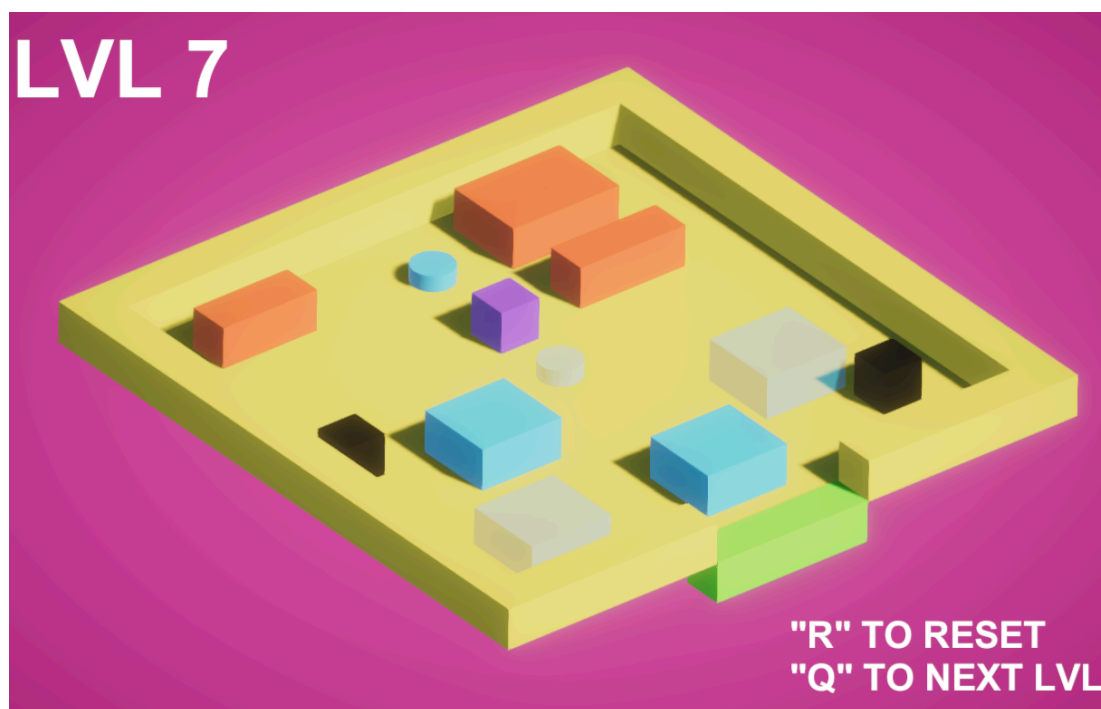


Рис. 4.22 – Сьомий рівень гри

Останній сьомий рівень, який зображено на рисунку 4.22, є другим варіантом використання нової механіки, яку було додано в 6 рівні. Гравцю тут прийдеться взаємодіяти з кнопками та помаранчевими кубами, адже в деяких моментах кнопка буде слугувати перешкодою.

Зм.	Арк.	№ докум.	Підпис	Дата

**ІАЛЦ.467800.003 ПЗ**

Арк.

76

## ВИСНОВКИ ДО РОЗДІЛУ 4

В четвертому розділі було створено музикальний проєкт для гри, а саме саундтрек. Ми реалізували основну ідею, використовуючи платформу Ableton Live, та внутрішні плагіни.

У даному розділі було описано весь проєкт, всю пророблену роботу, проілюстровано всі рівні (7) гри та варіанти їх проходження, а також інструкція по користуванню. Збільшити кількість рівнів, або взагалі різнообразити нашу гру не так і складно, за допомогою зручних інструментів та скрипту, який було написано, маючи невелику кількість часу ми можемо реалізувати будь-яку ідею.

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	Арк.
						77
<b><i>Зм.</i></b>	<b><i>Арк.</i></b>	<b><i>№ докум.</i></b>	<b><i>Підпис</i></b>	<b><i>Дата</i></b>		

## ВИСНОВКИ

Даний дипломний проєкт присвячений розробці комп'ютерної 3D гри з саунд-продюсуванням.

Для створення гри та музикального проєкту було виконано аналіз існуючих платформ (ігрових рушіїв і музикальних секвенсорів) та середовищ розробки, вивчивши всі переваги та недоліки, було визначено платформи з якими відбувалася робота.

Для створення ігрових механік та реалізації ідеї було виконано огляд існуючих відомих рішень, що відносяться до даної.

В ході роботи був виконаний загальний опис предметної області, була визначена основна ідея гри, були створені основні елементи та механіки, а також сформована задача музикального проєкту, та реалізована за допомогою платформи Ableton Live.

Реалізована комп'ютерна 3D гра має не складний інтерфейс, велику кількість різних елементів, та механік. Вимоги до апаратної системи користувача є не дуже вимогливими, завдяки простій концепції. Також важливо зазначити, завдяки правильному вибору платформи для створення гри, в нашому випадку – Unity, при необхідності, проєкт можна реалізувати на різних операційних системах та платформах.

Ознайомитись з кодом проєкту можна за наступним посиланням:  
<https://github.com/kionlx/diploma.git>

					<b><i>ІАЛЦ.467800.003 ПЗ</i></b>	Арк.
						78
<b><i>Зм.</i></b>	<b><i>Арк.</i></b>	<b><i>№ докум.</i></b>	<b><i>Підпис</i></b>	<b><i>Дата</i></b>		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Комп'ютерні ігри [Електронний ресурс] – Режим доступу до ресурсу: <https://mirnovogo.ru/kompyuternye-igry/>.
2. Дослідження Social Psychiatry and Psychiatric Epidemiology [Електронний ресурс] – Режим доступу до ресурсу: <https://www.springer.com/journal/127>.
3. Зображення Nimatron 1.2 [Електронний ресурс] – Режим доступу до ресурсу: <https://upload.wikimedia.org/wikipedia/commons/c/c0/Nimatron-picture.png>.
4. Зоображення 1.3 Computer Space [Електронний ресурс] – Режим доступу до ресурсу: <https://i.pinimg.com/originals/a7/d1/70/a7d170ea621d836047b73e42988cfl1d5.jpg>.
5. Зображення 1.4 Pong [Електронний ресурс] – Режим доступу до ресурсу: <https://i.ytimg.com/vi/TrezFjGF-Kg/maxresdefault.jpg>.
6. Зображення 1.5 Space Invaders [Електронний ресурс] – Режим доступу до ресурсу: [https://thumbs-prod.si-cdn.com/5EXIaf6WkalC33v9TqwQMA7k13o=/800x600/filters:no\\_upscale\(\)/https://public-media.si-cdn.com/filer/7f/4c/7f4c2d11-089f-4d86-8052-5cf95295be7e/file-20180531-69508-1oenzpj.png](https://thumbs-prod.si-cdn.com/5EXIaf6WkalC33v9TqwQMA7k13o=/800x600/filters:no_upscale()/https://public-media.si-cdn.com/filer/7f/4c/7f4c2d11-089f-4d86-8052-5cf95295be7e/file-20180531-69508-1oenzpj.png).
7. Зображення 1.6 Odyssey [Електронний ресурс] – Режим доступу до ресурсу: <https://ids.si.edu/ids/deliveryService?id=NMAH-2006-11760&max=1000>.
8. Зображення 1.7 Commodore 64 і ZX Spectrum [Електронний ресурс] – Режим доступу до ресурсу: [https://hsto.org/getpro/habr/post\\_images/c3b/3a9/329/c3b3a9329f065ee07612c955444cfbdd.jpg](https://hsto.org/getpro/habr/post_images/c3b/3a9/329/c3b3a9329f065ee07612c955444cfbdd.jpg).

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

**ІАЛЦ.467800.003 ПЗ**

Арк.

79

9. Зображення 1.8 Nintendo [Електронний ресурс] – Режим доступу до ресурсу: [https://cdn02.nintendo-europe.com/media/images/01\\_website\\_elements/hardware\\_1/hardware\\_2016\\_NintendoClassicMiniNintendoEntertainmentSystem\\_support\\_carousel\\_image.png](https://cdn02.nintendo-europe.com/media/images/01_website_elements/hardware_1/hardware_2016_NintendoClassicMiniNintendoEntertainmentSystem_support_carousel_image.png).
10. Зображення 1.9 Quake [Електронний ресурс] – Режим доступу до ресурсу: <https://3dnews.ru/assets/external/illustrations/2019/05/27/988189/02.jpg>.
- 11.Зображення 1.10 World Of Warcraft [Електронний ресурс] – Режим доступу до ресурсу: [https://bnetcmsus-a.akamaihd.net/cms/blog\\_header/xy/XYV31KQ28UP21572634907064.jpg](https://bnetcmsus-a.akamaihd.net/cms/blog_header/xy/XYV31KQ28UP21572634907064.jpg)
12. Зображення 1.11 Maze War [Електронний ресурс] – Режим доступу до ресурсу: [https://upload.wikimedia.org/wikipedia/commons/f/f4/Maze\\_war.jpg](https://upload.wikimedia.org/wikipedia/commons/f/f4/Maze_war.jpg).
13. Аналіз найдорожчих ігор [Електронний ресурс] – Режим доступу до ресурсу: <https://itc.ua/articles/igry-na-100-mln-samye-dorogie-videoigry-v-istorii/>.
- 14.Зображення 1.13 Minecraft [Електронний ресурс] – Режим доступу до ресурсу:  
[https://static.wikia.nocookie.net/minecraft\\_gamepedia/images/1/1c/Survival\\_1.16.png/revision/latest/scale-to-width-down/375?cb=20201006022808](https://static.wikia.nocookie.net/minecraft_gamepedia/images/1/1c/Survival_1.16.png/revision/latest/scale-to-width-down/375?cb=20201006022808).
15. Зображення 1.14 mafia 2 [Електронний ресурс] – Режим доступу до ресурсу: [https://cdn.vox-cdn.com/thumbor/aRW-IUUNpdVtouCI5QkmrSs43m0=/62x0:737x450/1200x800/filters:focal\(62x0:737x450\)/cdn.vox-cdn.com/uploads/chorus\\_image/image/1952279/mafia2.06.lg.0.jpg](https://cdn.vox-cdn.com/thumbor/aRW-IUUNpdVtouCI5QkmrSs43m0=/62x0:737x450/1200x800/filters:focal(62x0:737x450)/cdn.vox-cdn.com/uploads/chorus_image/image/1952279/mafia2.06.lg.0.jpg).
- 16.Зображення 1.15 Call of Duty [Електронний ресурс] – Режим доступу до ресурсу: <https://i.ytimg.com/vi/Sq1HCNMjJMo/maxresdefault.jpg>.

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

**ІАЛЦ.467800.003 ПЗ**

Арк.

80

- 17.Зображення 1.16 Assasins’s Creed [Електронний ресурс] – Режим доступу до ресурсу: <https://i.ytimg.com/vi/3AgcKKjxDME/maxresdefault.jpg>.
18. Зображення 1.18 Mortal Combat [Електронний ресурс] – Режим доступу до ресурсу: <https://i.ytimg.com/vi/yGNZhDEssKE/maxresdefault.jpg>.
19. Зображення 1.19 Sims [Електронний ресурс] – Режим доступу до ресурсу: [https://game-tips.ru/game\\_fps\\_test/games/The%20Sims%204/screenshots2.jpg?x76266](https://game-tips.ru/game_fps_test/games/The%20Sims%204/screenshots2.jpg?x76266).
20. Зображення 1.20 microsoft flight simulator [Електронний ресурс] – Режим доступу до ресурсу: [https://i.ytimg.com/vi/rqD\\_GG9tYz0/maxresdefault.jpg](https://i.ytimg.com/vi/rqD_GG9tYz0/maxresdefault.jpg).
- 21.Зображення 1.21 FIFA [Електронний ресурс] – Режим доступу до ресурсу: <https://i.ytimg.com/vi/QJX0vXDP4g/maxresdefault.jpg>.
22. Зображення 2.2 [Електронний ресурс] – Режим доступу до ресурсу: [https://api.pixonic.com/uploads/ckeditor/pictures/250/content\\_Без\\_названия\\_4\\_.jpg](https://api.pixonic.com/uploads/ckeditor/pictures/250/content_Без_названия_4_.jpg).
- 23.Зображення 2.3 [Електронний ресурс] – Режим доступу до ресурсу: [http://www.ozone3d.net/public/jegx/201108/cryengine3\\_sandbox.jpg](http://www.ozone3d.net/public/jegx/201108/cryengine3_sandbox.jpg).
- 24.Зображення 2.4 [Електронний ресурс] – Режим доступу до ресурсу: [https://cms-assets.tutsplus.com/uploads/users/438/posts/26554/final\\_image/FinalProduct.jpg](https://cms-assets.tutsplus.com/uploads/users/438/posts/26554/final_image/FinalProduct.jpg).
25. Зображення 2.5 [Електронний ресурс] – Режим доступу до ресурсу: <https://prosound.ixbt.com/news/2018/december/07/cubaise-10-0.jpg>.
26. C# документація [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/>.

<b>Зм.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>

**ІАЛЦ.467800.003 ПЗ**

**Арк.**

**81**

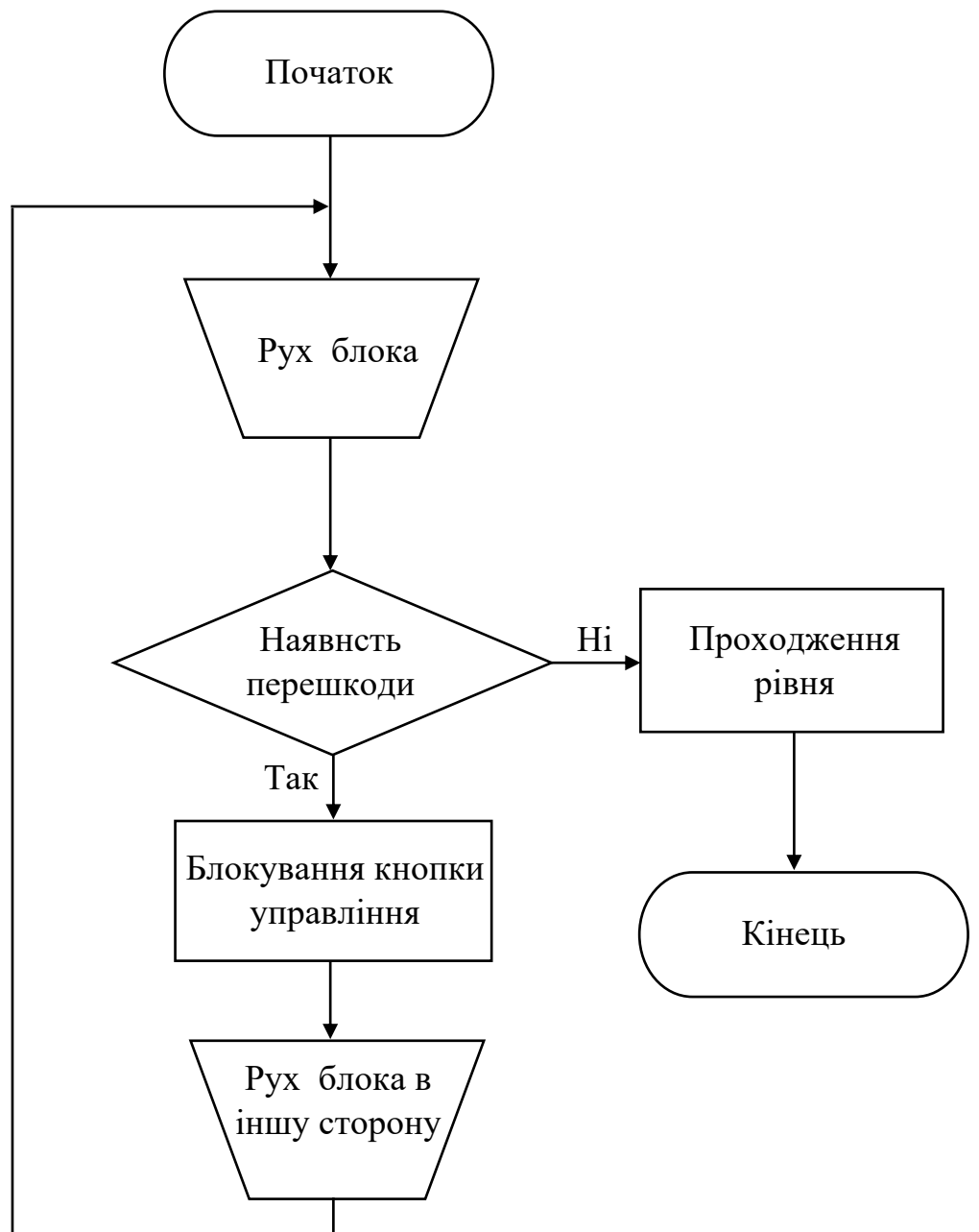
27. Visual Studio документація [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>.
28. Social Psychiatry and Psychiatric Epidemiology [Електронний ресурс] – Режим доступу до ресурсу: <https://www.springer.com/journal/127>.
29. Документація UnrealEngine [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unrealengine.com/en-US/index.html>.
30. Документація Unity [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unity3d.com/ru/530/Manual/>.
31. Історія розвитку комп'ютерних ігор [Електронний ресурс] – Режим доступу до ресурсу: [https://ru.wikipedia.org/wiki/История\\_компьютерных\\_игр](https://ru.wikipedia.org/wiki/История_компьютерных_игр).
32. How to choose a Game Engine [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-for-your-game>.
33. Video Game Genres [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hp.com/us-en/shop/tech-takes/video-game-genres>.
34. Difference od 2D and 3D games [Електронний ресурс] – Режим доступу до ресурсу: <https://www.makeuseof.com/2d-games-vs-3d-gamesdifferences/>.
35. Call of Duty [Електронний ресурс] – Режим доступу до ресурсу: [https://ru.wikipedia.org/wiki/Call\\_of\\_Duty\\_\(серия\\_игр\)#Популярность\\_и\\_продажи](https://ru.wikipedia.org/wiki/Call_of_Duty_(серия_игр)#Популярность_и_продажи).
36. Класифікація комп'ютерних ігор – [https://vlab.wikia.org/ru/wiki/Жанры\\_компьютерных\\_игр](https://vlab.wikia.org/ru/wiki/Жанры_компьютерных_игр).
37. Дослідження ігор в навчанні [Електронний ресурс] – Режим доступу до ресурсу: <http://www.edutainme.ru/post/7-games-research/>.

					<b>ІАЛЦ.467800.003 ПЗ</b>	Арк.
						82
Зм.	Арк.	№ докум.	Підпис	Дата		

38. Документація CryEngine [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.cryengine.com/display/SDKDOC2/Home>.
39. Music Production Software [Електронний ресурс] – Режим доступу до ресурсу: <https://rotorvideos.com/blog/guide-best-music-productionsoftware>.
40. Документація Ableton Live [Електронний ресурс] – Режим доступу до ресурсу: <http://www.incunabula.ru/blogs/makemusic/stuff/ALDocs.pdf>.
41. Введення в С# [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/sharp/tutorial/1.1.php>.
42. С# документація [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/>.
43. Game Engines [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gamedesigning.org/career/video-game-engines/>.

					<i><b>ІАЛЦ.467800.003 ПЗ</b></i>	Арк.
						83
<i><b>Зм.</b></i>	<i><b>Арк.</b></i>	<i><b>№ докум.</b></i>	<i><b>Підпис</b></i>	<i><b>Дата</b></i>		

**Додаток 1**  
**До дипломного проєкту**  
**на тему: «Комп'ютерна 3D гра**  
**з саунд-продюсуванням »**



Зм	Арк.	№ докум	Підпис	Дата
Розроб.		Людковський А.О		
Первір.		Кочура Ю.П.		
Н.Контр.		Сімоненко В.П		
Затверд.				

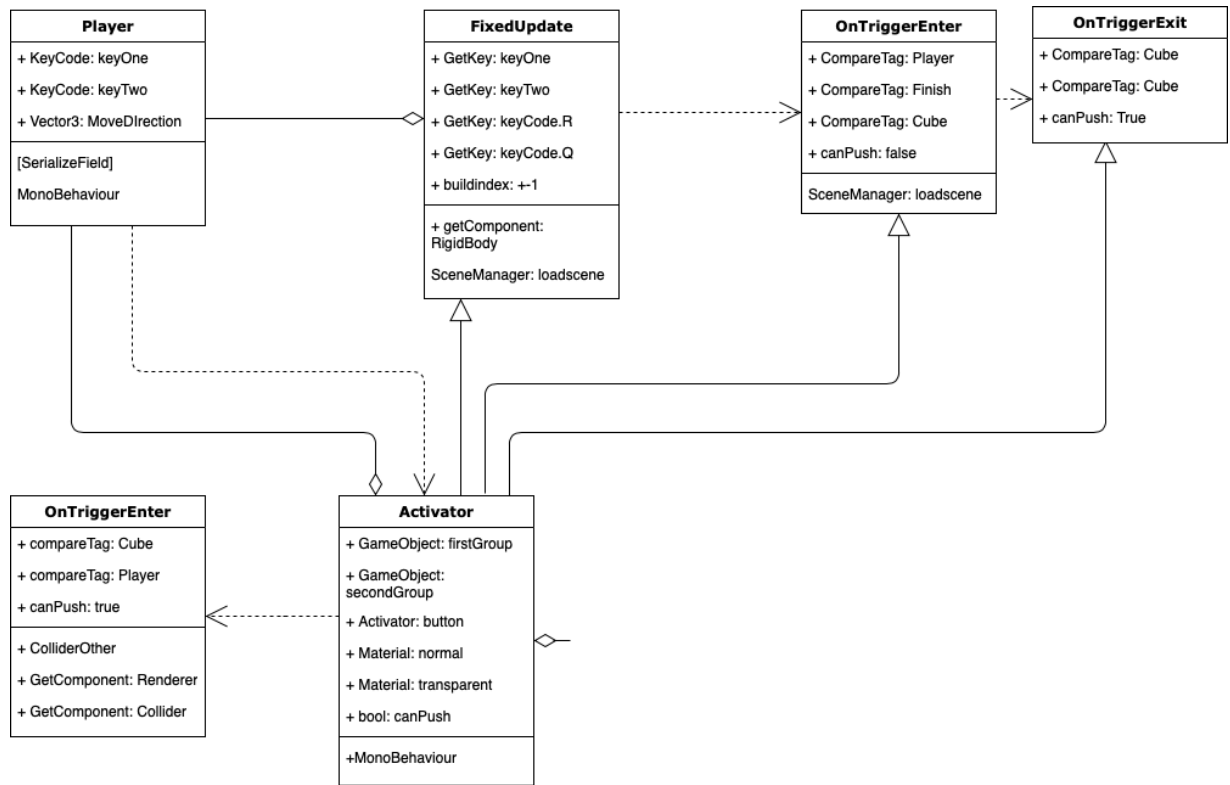
**ІАЛЦ.467800.004 Д1**

Комп'ютерна 3D гра з саунд-продюсуванням  
Принципова схема

Літ.	Арк.	Аркушіє
	1	1

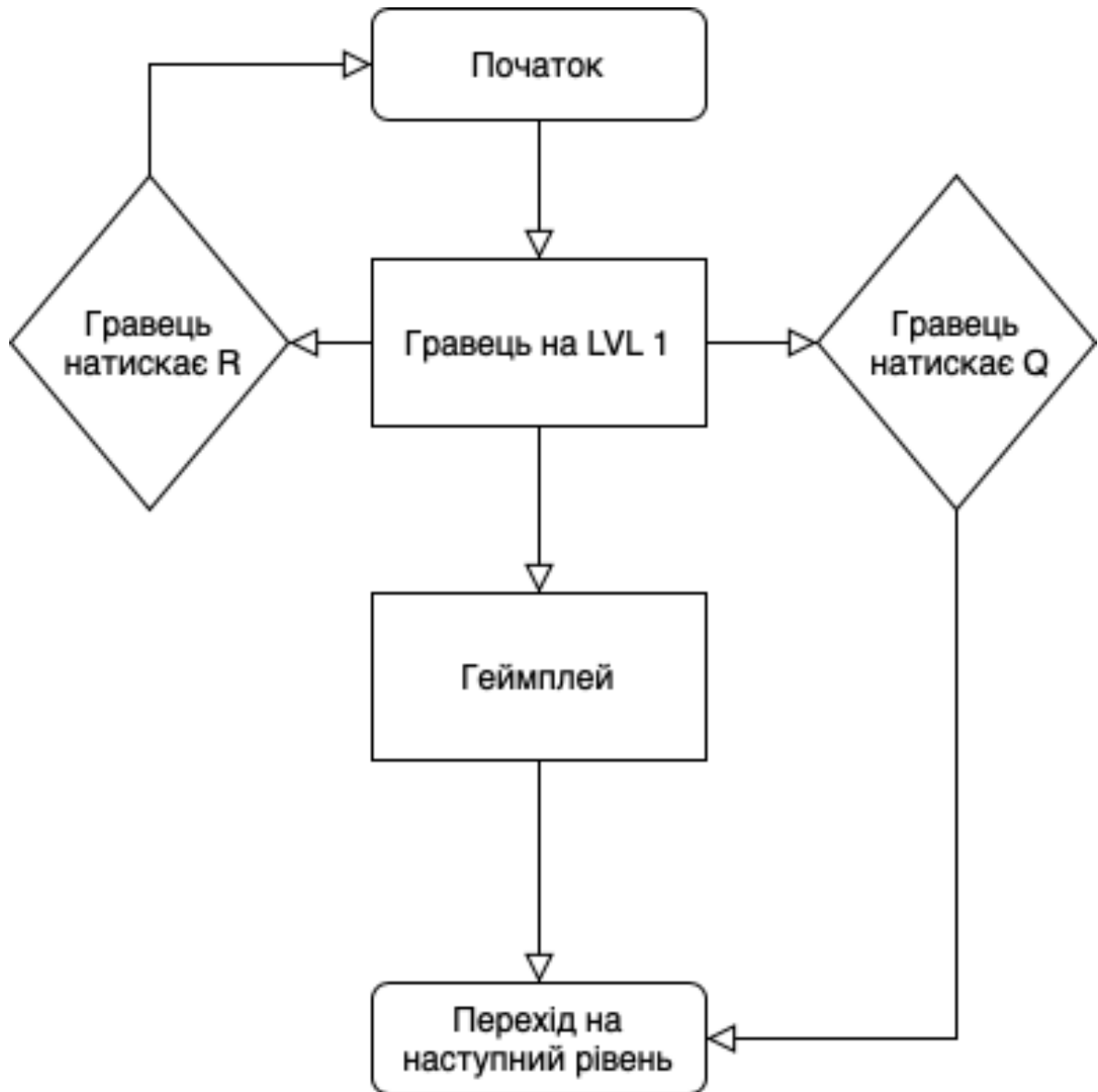
КПІ ім. Ігоря Сікорського  
ФІОТ, ІВ-73

**Додаток 2**  
**До дипломного проєкту**  
**на тему: «Комп'ютерна 3D гра**  
**з саунд-продюсуванням »**



					<b>ІАЛЦ.467800.005 Д2</b>		
<b>Зм</b>	<b>Арк.</b>	<b>№ докум</b>	<b>Підпис</b>	<b>Дата</b>			
<b>Розроб.</b>	Людновський А.О				<b>Лім.</b>	<b>Арк.</b>	<b>Аркушіє</b>
<b>Первір.</b>	Кочура Ю.П.					1	1
<b>Н.Контр.</b>	Сімоненко В.П				<b>Комп'ютерна 3D гра з саунд-продюсуванням</b> <b>Функціональна схема</b> <b>КПІ ім. Ігоря Сікорського</b> <b>ФІОТ, ІВ-73</b>		
<b>Затверд.</b>							

**Додаток 3**  
**До дипломного проєкту**  
**на тему: «Комп'ютерна 3D гра**  
**з саунд-продюсуванням »**



Зм	Арк.	№ докум	Підпис	Дата
Розроб.		Людковський А.О		
Первір.		Кочура Ю.П.		
Н.Контр.		Сімоненко В.П		
Затверд.				

**ІАЛЦ.467800.006 Д2**

Комп'ютерна 3D гра з саунд-продюсуванням  
Структурна схема

Літ.	Арк.	Аркушіє
	1	1

КПІ ім. Ігоря Сікорського  
ФІОТ, ІВ-73

