

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

**ДОСЛІДЖЕННЯ МЕТОДІВ РОЗРОБЛЕННЯ ІГРОВОГО СВІТУ**

(тема)

Виконав:

студент 2 курсу, групи ІНФМ-21-1

Борисенко Є. А.

(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика

(повна назва освітньої програми)

Керівник доц. Творошенко І.С.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

\_\_\_\_\_

(підпис)

Кобилін О.А.

(прізвище, ініціали)

2022 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)Кафедра Інформатики  
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_» \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентів Борисенку Єгору Андрійовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження методів розроблення ігрового світу

затверджена наказом по університету від 9 листопада 2022 року № 1469Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 22 листопада 2022 р.3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека SDL, мова програмування C++, середовище розробки NetBeans IDE.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Проаналізувати сучасний стан питання що розглядається.2. Виявити особливості методів для розроблення ігрового світу.3. Здійснити вибір інструментальних засобів для програмної реалізації відеогри на основі обраних методів.4. Програмно реалізувати відеогру на основі обраних методів.5. Протестувати розроблений застосунок та провести аналіз результатів.6. Виявити перспективи подальшої роботи.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність дослідження, об'єкт та мета дослідження, постановка задачі дослідження, вихідні дані дослідження, етапи розроблення, результат тестування, висновки, перспективи подальших досліджень, апробація результатів роботи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	09.11.2022	
2	Аналіз завдання, підбір літератури	09.11.22-10.11.22	
3	Аналіз літератури з досліджуваної проблеми	11.11.22-12.11.22	
4	Аналіз методів розроблення ігрового світу	13.11.22-14.11.22	
5	Розробка методу	15.11.22-16.11.22	
6	Програмна реалізація	17.11.22-18.11.22	
7	Оформлення пояснювальної записки	19.11.22-20.11.22	
8	Перевірка на плагіат	21.11.2022	
9	Рецензування	22.11.2022	
10	Підготовка презентації та доповіді	27.11.2022	
11	Занесення роботи в електронний архів	28.11.2022	
12	Попередній захист кваліфікаційної роботи	30.11.2022	

Дата видачі завдання 9 листопада 2022 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Творошенко І.С.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 79 с., 58 рис., 111 джерел.

ВІДЕОГРА, ГРАВЕЦЬ, ТЕКСТУРА, АНІМАЦІЯ, СПРАЙТ, 2D, 3D, КАМЕРА, КОМП'ЮТЕР, КОНСОЛЬ, ІНДІ, ОБРОБКА, ПІКСЕЛЬ, ВОКСЕЛЬ, ПОЛІГОН, МЕШ, ДЕТАЛІЗАЦІЯ, ШЕЙДЕР, СЕТИНГ, ГРАФІКА.

Об'єктом дослідження є відеоігрова індустрія.

Метою дослідження є реалізація та вивчення методів створення ігрових світів різного рівню складності та наповнення.

У ході аналізу сучасного стану відеоігрової індустрії розбито методи на дев'ять основних класів, зроблено огляд розвитку та сучасних технологій, що використовуються для створення ігрових світів. На основі наукової літератури додатково сформовано важливі аспекти при створенні відеоігор.

Детально розглянуто методи розмірності простору, методи встановлення камери або точки зору, методи анімації та методи взаємодії розробників з гравцями.

У результаті дослідження здійснена програмна реалізація простої 2D гри.

VIDEOGAME, PLAYER, TEXTURE, ANIMATION, SPRITE, 2D, 3D, CAMERA, COMPUTER, CONSOLE, INDY, PROCESSING, PIXEL, VOXEL, POLYGON, MESH, DETALIZATION, SHADER, SETTING, GRAPHICS.

The object of research is the video game industry.

The purpose of the research is the implementation and study of methods of creating game worlds of various levels of complexity and content.

During the analysis of the current state of the video game industry, methods were divided into nine main classes; an overview of the development and modern technologies used to create game worlds was made. On the basis of scientific literature, important aspects in the creation of video games were additionally formed.

Spatial dimensioning techniques, camera or point of view setup techniques, animation techniques, and developer-player interaction techniques are discussed in detail.

As a result of the research, a software implementation of a simple 2D game was made.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	7
Вступ.....	8
1 Аналіз існуючих методів для розроблення ігрового світу.....	11
1.1 Класифікація існуючих методів для розроблення ігрового світу .....	11
1.1.1 Методи розмірності простору .....	11
1.1.2 Методи постанови камери.....	12
1.1.3 Методи анімації.....	12
1.1.4 Методи взаємодії розробників з гравцем.....	13
1.1.5 Методи взаємодії гри з гравцем.....	14
1.1.6 Методи жанрів .....	14
1.1.7 Методи подачі історії.....	15
1.1.8 Методи вводу .....	16
1.1.9 Методи оплати та монетизації .....	16
1.2 Аналіз сучасних підходів для розроблення ігрового світу .....	17
1.3 Аналіз літературних джерел щодо апробації результатів застосування методів для розроблення ігрового світу .....	21
1.4 Постановка задачі дослідження .....	24
2 Дослідження вибраних методів розроблення ігрового світу.....	26
2.1 Розмірність простору світу гри .....	26
2.1.1 2D .....	26
2.1.2 2.5D або псевдо 3D.....	27
2.1.3 3D .....	29
2.1.4 4D .....	33
2.2 Постанова камери .....	36
2.2.1 Від 1-ї особи.....	36
2.2.2 Від 2-ї особи.....	37
2.2.3 Від 3-ї особи.....	38
2.2.4 Ізометрія .....	40

	6
2.3 Анімація.....	41
2.3.1 Спрайтова анімація .....	41
2.3.2 Векторна анімація .....	42
2.3.3 Скелетна анімація.....	42
2.3.4 Stop-motion анімація .....	45
2.4 Взаємодія розробників з гравцем.....	45
2.4.1 Спільнота.....	45
2.4.2 Щоденники розробників.....	47
2.4.3 Досягнення .....	47
2.4.4 Секрети та пасхалки.....	49
2.4.5 Події.....	49
2.4.6 Додатковий контент та оновлення .....	50
3 Порівняння вибраних методів розроблення ігрового світу.....	51
3.1 Вибір інструментальних засобів для реалізації вибраних методів.....	51
3.2 Етапи програмної реалізації вибраних методів для розроблення ігрового світу .....	53
3.3 Тестування програмної реалізації комп'ютерної гри на основі обраних методів розроблення ігрового світу .....	56
3.4 Перспективи подальшої роботи .....	67
Висновки .....	68
Перелік джерел посилання .....	69

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

2D – двовимірний

2.5D – дво з половиною вимірний

3D – тривимірний

4D – чотиривимірний

ГБ – гігабайт

ІТ – інформаційні технології

DLC – Downloadable Content (контент завантаження)

DMA – Dynamic Memory Allocation (динамічний розподіл пам'яті)

HDD – Hard Disk Drive (жорсткий магнітний диск)

HDMI – High Definition Multimedia Interface (інтерфейс для мультимедіа високої чіткості)

HDR – High Dynamic Range (загальна назва технологій роботи із зображеннями і відео, діапазон яскравості яких перевищує можливості стандартних технологій відображення зображень.)

IDE – Integrated Development Environment (інтегроване середовище розробки)

NFT – Non-Fungible Token (невзаємозамінний токен)

RPG – Role-Playing Game (рольова відеогра)

SDL – Simple DirectMedia Layer (мультимедійна бібліотека)

SSD – Solid-State Drive (твердотілий накопичувач)

VR – Virtual Reality (віртуальна реальність)

## ВСТУП

Індустрія створення відеоігор одна з наймолодших та одна з найбільш стрімко зростаючих сфер сьогодення, попри те що існує застаріла концепція, що ігри це лише дитячі забавки – це вже давно не так, оскільки більшість із сучасних відеоігор мають вікові обмеження від 18 років [1].

Починаючи з 50-х років 20-го століття, коли були створені перші відеоігри, які в той час були досить примітивними, але за рахунок новизни вважалися серйозними науковими проектами [2, 3]. За наступні два десятиріччя вони вийшли з наукових лабораторій та наукових виставок і на початку 70-х років стали новим феноменом [4]. Ця епоха називається золотим віком відеоігор.

Розвиток індустрії відеоігор дуже тісно пов'язаний з можливостями та доступністю електронно-обчислювальних машин. Так, зі створенням домашніх консолей комп'ютерні ігри мігрували з гральних залів у квартири та приватні домогосподарства [4]. На початку 90-х років, із ростом доступності персональних комп'ютерів та їх позиціонуванням на ринку як більш універсальних пристроїв ігри для ПК стали масовими [4].

Впродовж наступних років і до теперішнього часу великі відеоігри ставали більш структурно складними та наближалися до реального світу за деталізацією та реалізмом.

Оскільки комп'ютерні ігри є інтерактивними розвагами, їх обирають в першу чергу щоб переключитися від реального світу та розважитися. Дослідники виділяють чотири основні та вісім похідних категорій гравців, які обирають ігри в залежності від різних факторів що впливають на мотивацію гравців [5].

Перша основна категорія – накопичувачі або кар'єристи. Їхньою мотивацією є накопичення сили, засобів, об'єктів та інших ресурсів і благ, що надаються грою. Іншими словами, стимулом для даної категорії є прогрес і зростання в процесі гри.

Накопичувачі поділяються на усвідомлених – планувальників та неусвідомлених – опортуністів. Перші скрупульозно продумують всі свої дії і ходи, другі – намагаються не просто скрізь побачити нові можливості, але і миттєво в них вчепитися.

Друга основна категорія – кілери. Головною мотивацією для них служить перевага над іншими учасниками, наявність влади у грі та відчуття насолоди після перемоги над іншими гравцями. Всіма силами прагнуть до неї. Усвідомлений кілер вважається політиком, він використовує будь-які внутрішньо ігрові засоби для досягнення своїх цілей. Неусвідомлений кілер стає гриффером або тролем, тобто гравцем, що виміщає свою злість та біль на оточуючих гравцях, також це може робитися для псування ігрового досвіду для інших гравців.

Третя основна категорія – дослідники. Вони намагаються якомога краще дізнатися ігрову реальність і її секрети, битви і активні дії для них другорядні. Усвідомлений дослідник – це вчений, він ретельно обдумує свої дії, мислить логічно, вивчає ігрові механіки, знаходить способи отримання ігрових переваг та варіанти їх застосування. Так у нього формується цілісна картина знання. Несвідомий дослідник – хакер, його дослідження полягає у способах заламати гру, наприклад знайти помилки або дірки у світі гри.

Четверта основна категорія – соціальники або комунікатори, вони прагнуть до спілкування з іншими учасниками, соціальної взаємодії і взаєморозуміння. Усвідомлені соціальники називаються громадськими діячами, а неусвідомлені – друзями.

Перші відрізняються тим, що застосовують свої соціальні зв'язки для отримання максимальної вигоди від партнерів по грі, а другі перебувають у постійному пошуку того, з ким можна просто гарно провести час у грі.

Хоча основна задача більшості відеоігор є надання можливості переключитися та розважитися, сучасні комп'ютерні ігри також дозволяють вивчити щось нове, створити відчуття інтерактивного музею або попередити розвиток деяких захворювань, які пов'язані з роботою мозку.

Також існують ігри які потребують активних фізичних дій.

Актуальність дослідження полягає в тому, що впровадження методів розроблення ігрових світів для їх створення для гравців є одним зі способів проведення дозвілля, вивчення чогось нового та при обмеженому застосуванні ефективним методом профілактики для покращення здоров'я.

# 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ДЛЯ РОЗРОБЛЕННЯ ІГРОВОГО СВІТУ

## 1.1 Класифікація існуючих методів для розроблення ігрового світу

Оскільки створення ігрових світів є доволі комплексним питанням, то методи для їх розроблення можна розділити на декілька категорій.

### 1.1.1 Методи розмірності простору

Загально вважається, що людство живе у тривимірному просторі, а в деякій літературі до цього додається ще часовий вимір [6, 7].

Для ігор часовий вимір не завжди є можливим з технічної точки зору, оскільки для зберігання попередніх станів системи необхідна значна кількість ресурсів, а саме – пам'яті, тому, чим більше світ гри має об'єктів, тим важче його обробляти, не кажучи про те, щоб зберігати результати попередніх обробок.

Математично можливо прорахувати світ будь-якої розмірності простору, хоч одновимірний, хоч двадцятивимірний, але це не має сенсу, через те, що людина загалом сприймає навколишній світ та будь-яку інформацію навколо себе у першу чергу через аудіовізуальні сигнали.

Більш того, екрани будь-якого розміру здатні відображати лише двовимірне зображення, хоча вже існують технології здатні до відображення тривимірного зображення через голограми, вони не є доступними для широкої громадськості.

Виходячи з цього, зображати можна лише тривимірні світи та світи, розмірність яких менша.

Найбільш розповсюдженими є дво- та тривимірні ігрові світи.

До таких методів входять:

- 2D;
- 2.5D або псевдо 3D;
- 3D;
- 4D.

### 1.1.2 Методи постанови камери

Категорія методів постанови камери теж є однією з найбільш фундаментальних для розробки ігрового світу. Вона визначає, з якої точки зору відбувається гра, та які технічні рішення необхідно впровадити. Від постанови камери залежить наскільки складною може бути розробка та які обмеження можна отримати в процесі.

Камеру можна встановити від 1-ї, 2-ї та 3-ї особи.

### 1.1.3 Методи анімації

Гра не може бути статичною, якщо не відбуваються якісь зміни на екрані, то це і відеогрою назвати не можна.

Анімація – це найбільш наочне зображення, які дії відбуваються, а також це спосіб передачі величезної кількості інформації, за долі секунди гравець може зрозуміти, що відбувається та прийняти рішення, що йому потрібно робити [8]. Більш досвідчені гравці здатні приймати рішення значно швидше, розуміючи патерни та очікуючи певних речей, у порівнянні з людьми, які приєдналися до цієї спільноти недавно.

Анімувати персонажів та оточення можна за допомогою наступних методів:

- спрайтова анімація;

- векторна анімація;
- скелетна анімація;
- stop-motion анімація.

#### 1.1.4 Методи взаємодії розробників з гравцем

Ігрова індустрія трохи відрізняється від інших продуктів в сфері ІТ та сфері розваг тим, що зв'язок між розробниками та користувачами не такий сильний як в ІТ, та не такий слабкий, як мають інші галузі сфери розваг. Це щось середнє, коли немає прямої консультації між замовником та виконавцем, та це й не щось само в собі як, наприклад, кінофільми чи література.

Комунікація може йти різними шляхами, у тому числі, завдячуючи розвитку Інтернету, можливості для зворотного зв'язку значно збільшилися, вони перейшли до соціальних мереж та спеціальних платформ. Унаслідок цього, у розробників з'явилося більше можливостей виправити помилки та загалом покращити свої продукти.

Розробники можуть взаємодіяти зі своєю аудиторією через такі підходи:

- через спільноту;
- публікуючи щоденники розробки;
- досягнення;
- заховані секрети, які називають пасхальними яйцями або пасхалками;
- події;
- додатковий контент та оновлення.

### 1.1.5 Методи взаємодії гри з гравцем

Будь-яка гра перш за все цікава тим, що вона може запропонувати гравцю. Це може бути зовсім новий досвід, переосмислення чогось старого з нового ракурсу або власне те саме старе, що грає на струнах ностальгії.

У цілому до таких методів входять внутрішні можливості гри по обробці ігрового світу, наприклад, штучний інтелект не ігрових персонажів та фізика оточення.

До даних методів відносяться:

- The Director;
- Nemesis;
- A-LIFE;
- метод адаптації;
- Close Call;
- метод залежності можливостей персонажа від його вигляду;
- зворотній зв'язок;
- Chemical engine;
- система карми;
- Permadeath.

### 1.1.6 Методи жанрів

Жанр відеоігор – це категорія ігор, пов'язаних схожими ігровими характеристиками [9, 10].

Жанри відеоігор, зазвичай, визначаються не обстановкою, не історією гри, не її середовищем, а тим, як гравець взаємодіє з грою. Жанри можуть охоплювати широкий спектр ігор, що призводить до ще більш специфічних класифікацій, які називаються піджанрами.

Загалом жанр можна охарактеризувати як серцевину гри, набір певних дій що є основою, їх ще називають центральними механіками [9, 10]. Центральних механік може бути декілька, через це гру можна відносити відразу до декількох жанрів.

До даних підходів відносяться:

- платформер;
- стратегія;
- RPG;
- симулятор;
- шутер;
- Adventure;
- Action;
- хоррор;
- файтинг;
- головоломка та інші.

#### 1.1.7 Методи подачі історії

Сучасні ігрові світи з плином часу стають все більш структурно складними, вони вже давно вийшли за межі простих геометричних форм, які якимось переміщуються по екрану, зараз це складні застосунки, які можуть нести глибокі сенси та навчати багатьом речам на кшталт математики, світової історії, психології та багато чому іншому [11, 12].

Прикладами побудови сюжету для ранніх ігор були література та кінематограф, відеоігрова індустрія багато з них перейняла, але, оскільки ігри це інтерактивні розваги, то і методи подачі сюжету в них відрізняються. Вони еволюціонували, стали більш гнучкими, майже недоступними для тих же літератури та кінематографа.

Подавати історію можна лінійно, нелінійно, заховано та через оточення і навколишній світ.

#### 1.1.8 Методи вводу

Як і в будь-якій програмі в ігровому світі користувач повинен якимось чином передавати інформацію про свої дії до віртуального середовища, для цього існує багато способів, винайдених за 50 років існування індустрії [13–15].

Вводити інформацію можна за допомогою наступних методів:

- клавіатура та комп'ютерна миша;
- смартфон;
- геймпад;
- мікрофон;
- елементи віртуальної реальності;
- сенсорні контролери;
- комп'ютерний руль;
- Amiibo;
- нейрошолом та інші.

#### 1.1.9 Методи оплати та монетизації

Як і в будь-якій іншій сфері для створення будь-чого потрібні ресурси, відеоігри не є винятком. Великі компанії навчилися не тільки отримувати дохід від створеного продукту для утримання робітників та розвитку, але й збирати надприбутки. Наприклад, річний дохід Activision Blizzard за 2021 рік склав 8,803 мільярда доларів, що на 8,87% більше ніж у 2020 році [16]. Такі цифри досить наочно демонструють, що і у відеоігровій галузі можна побудувати успішний бізнес зі значними грошовими обертами.

Продати та додатково монетизувати гру можна наступними методами:

- рітейл;
- цифрова дистрибуція;
- DLC;
- підписка;
- Season pass;
- Battle pass;
- мікротранзакції;
- лутбокси та інші.

## 1.2 Аналіз сучасних підходів для розроблення ігрового світу

Галузь розробки відеоігор дуже тісно пов'язана з технологічним прогресом та розвитком у більш широкій сфері інформаційних технологій. За останні п'ять років відбулася значна еволюція в підходах до розроблення ігрових світів.

По-перше, починаючи з осені 2018 року компанія Nvidia випустила новий тип домашніх графічних карт побудованих на мікроархітектурі Turing, ця інновація дозволяє обчислювати специфічні операції з плаваючою точкою з трасування променів у реальному часі [17].

Інший важливий крок був зроблений компаніями виробниками процесорів – спочатку AMD а згодом Intel, вони покращили технічний процес та архітектури своїх продуктів, це призвело до збільшення продуктивності, як за рахунок більшої потужності процесорів для обробки, так і завдяки розширенню шин передачі даних [18, 19]. Окрім цього, важливим елементом еволюції стало також покращення роботи з пам'яттю, оперативна пам'ять стала швидшою та більш об'ємною, постійна пам'ять також збільшилася та набула більш широке використання користувачами SSD дисків, які прийшли на заміну HDD [20, 21].

Також доволі масовим стало використання HDR, технології що покращує передачу кольору [22]. Після персональних комп'ютерів ці нововведення прийшли на ринок персональних консолей. У листопаді 2020 року з випуском консолей Xbox Series X/S, а також PlayStation 5 почалося дев'яте покоління ігрових приставок [23, 24].

По-друге, отримавши зменшення технічних обмежень, розробники відеоігор почали реалізовувати переваги нових пристроїв. Спочатку була дуже масова хвиля впровадження підтримки для моніторів та телевізорів високої роздільної здатності, від 2 мільйонів пікселів до 10 мільйонів пікселів, при такому об'ємі інформації сталося дві речі:

- загальне збільшення розміру гри;
- значне навантаження на шини та кабелі зв'язку, які передавали дані з графічного пристрою до пристрою відображення.

Розміри найбільших ігор збільшилися з 20 ГБ до 100-110 ГБ, а деякі погано оптимізовані ігри перейшли навіть за 230 ГБ, це пояснюється високою деталізацією текстур та кат-сцен. Інша проблема була тимчасовою, в кращому випадку можна було передати до 30 кадрів на секунду, з приходом нових стандартів DisplayPort 2.0 та HDMI 2.1 вдалося передавати в 4-8 разів більше кадрів [25, 26], у залежності від роздільної здатності для персональних комп'ютерів та ігрових приставок відповідно.

У той же час, ринок індустрії зростав шаленими темпами, в першу чергу, це відбулося за рахунок значного збільшення долі мобільного сегменту, хоча комп'ютерний та консольний сегменти теж стабільно розширювалися, але не так стрімко як мобільний.

Іншим поштовхом до розвитку ринку, як не дивно, стала пандемія короно вірусу, яка почалася на початку 2020 року. На рисунку 1.1 зображено об'єм ринку відеоігрової індустрії станом на липень 2022 року.

На рисунку 1.1 видно, що мобільний сегмент перевершує сегмент персональних комп'ютерів та консолей разом узятих, це, в свою чергу, пов'язано з дуже великою розповсюдженістю смартфонів та їх доступністю у порівнянні з іншими ігровими приладами.

## Глобальний ринок відеоігр 2022

За сегментами

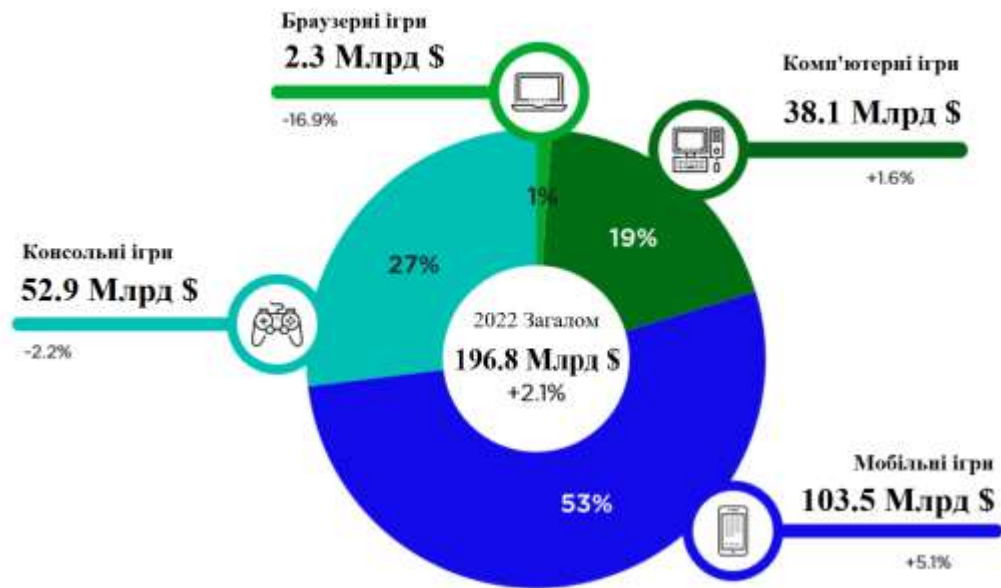


Рисунок 1.1 – Об'єм ринку відеоігрової індустрії

На даний час існують і деякі інші технології які в майбутньому можуть стати великими сегментами ринку, але всі вони поки що не мають широкої аудиторії. До таких технологій входять віртуальна та доповнена реальність, доволі успішними проєктами яких є «Half-Life: Alyx» та «Pokémon Go» відповідно [27, 28].

Ще одним новим підходом є хмарний геймінг, поки що його ключові проблеми полягають в тому, що користувачу необхідно мати максимально швидкий Інтернет без перебоїв, щоб отримати доступ до серверів, де відбувається обробка [29].

Деякими проєктами, що інвестували у розвиток такого підходу є NVIDIA GeForce NOW, Amazon Luna та Google Stadia [30–32]. Через малу аудиторію гравців Google Stadia змушений припинити свою діяльність, основною причиною такого рішення є збитковість.

Абсолютно новими підходами є використання метавсесвітів та технології Blockchain для NFT [33].

У розробці та видавництві індустрія розділилася на дві основні течії, одна з яких – це великі корпорації, а інша – незалежні розробники, їх також називають інді-розробниками.

Розміри команд у корпораціях починаються з сотень людей та можуть доходити до кількох тисяч, з іншого боку у інді-розробників за весь процес розробки та публікації може відповідати одна людина, найбільші колективи незалежних розробників можуть доходити до 100-150 людей [9, 10].

Розміри команд не завжди визначають розмір та бюджет самої гри. Проекти корпорацій характеризуються невеликим бажанням експериментувати, високою залежністю від інвесторів та акціонерів, спробами зробити гру більш інклюзивною і, за рахунок цього, розширити аудиторію, дуже часто – великими бюджетами.

Великі відеоігри, на проходження яких необхідно витратити від 50 і більше годин називають AAA проектами [14]. Раніше літери характеризували коштовність гри, але ще на початку 2010-х років реальні бюджети почали сильно виходити за межі встановлених буквених позначень, на сьогодні це використовується, як маркетингова стратегія, та несе певну характеристику того, що очікують гравці.

Інді-ігри, з іншого боку, можуть сильно відрізнитися одна від одної, також вони ніяк не прив'язані до конкретних бюджетів, хоча вони, зазвичай, невеликі, існують і винятки. Наприклад, орієнтовна вартість AAA гри «Grand Theft Auto V» становила приблизно 265 мільйонів доларів, в той час як інді-гра «Star Citizen» на основі краудфандінгу зібрала більше, ніж 500 мільйонів доларів, більш того, збір коштів ще продовжується [34, 35].

Маленьким іграм через різні причини, найчастіше, невелику аудиторію, важко задати тренд в індустрії, тому, зазвичай, тренди задають великі проекти. До них відноситься «The Legend of Zelda: Breath of the Wild», головним трендом цієї відеоігри для багатьох інших ігор стала модель дослідження світу гри та експерименти із взаємодіями [36].

«God of War» 2018 року показала приклад динамічного бойового процесу та повністю безшовні переходи між кат-сценами та ігровим процесом [37], ігри від Naughty Dog дивували високою кінематографічністю та вмінням тримати гравця в напрузі [38], «Red Dead Redemption 2» вирізнялася своєю надзвичайною деталізованістю [39] та фото реалістичністю, «Fortnite» став прикладом для багатокористувацьких ігор, зробивши багато небачених нововведень [40].

Прогрес також відбувся в програмному забезпеченні для створення відеоігор – ігрових рушіях. У новому ігровому рушії Unreal Engine 5 було додано технології, які значно полегшують та пришвидшують створення ігрових світів. Наприклад, за допомогою системи віртуалізованої геометрії Nanite оптимізується обробка високої кількості об'єктів. Інша корисна технологія – Lumen полегшує роботу розробників в плані роботи з освітленням, роблячи його динамічним [41].

### 1.3 Аналіз літературних джерел щодо апробації результатів застосування методів для розроблення ігрового світу

Відеоігри є популярним видом дозвілля для людей усіх вікових категорій у всьому світі. Незважаючи на деяку негативну репутацію та емпіричні дослідження, які висвітлюють більш негативні аспекти відеоігор, вони також мають і терапевтичний ефект.

До корисних ефектів від гри у комп'ютерні ігри входять [42]:

- когнітивна реабілітація;
- фізіотерапія та трудотерапія;
- розвиток соціальних та комунікативних навичок серед людей із вадами навчання;
- регуляція стресу, тривоги та емоцій;
- фізична активність з використанням фітнес-ігор.

Для того, щоб розуміти, що таке відеоігри та вміти створювати їх, необхідно знати їхню історію розвитку та філософію розробки.

Сід Меєр – один із надзвичайно досвідчених розробників та піонер в індустрії висловлює свою думку, згідно з якою відеогра має бути «низкою цікавих рішень», він також ділиться своїм поглядом на історію індустрії, психологію геймерів і захоплююче розуміння творчого процесу, включно з його десятьма правилами гарного ігрового дизайну [43].

Відеоігри стають все більш центральною частиною культурного життя, впливаючи на різні аспекти повсякденного життя, спільноту та формування ідентичності.

Сучасні відеоігри можуть давати важливе уявлення про сучасну природу цифрової культури та культуру участі, моделі споживання та формування ідентичності, пізню модерність і сучасну політичну раціональність [44].

Вже давно пройшли часи, коли ігри створювалися ентузіастами та мали невеликі об'єми робіт. З розвитком технологій та підходів обсяги робіт збільшилися і складні ігри стали потребувати більших команд, як наслідок, з'явилися проблеми, які характерні і для інших сфер ІТ-індустрії.

Такими проблемами стали виконання замалих термінів розробки, недостатнє фінансування та вигорання працівників [45, 46].

Сама розробка починається з концепції та дизайну, хороший дизайн гри досягається, коли вона розглядається з якомога більшої кількості точок зору, існує понад 100 наборів запитань, які потрібно поставити, коли відеогра створюється, грається та змінюється, допоки процес дизайну не фіналізується та матиме кінцеві форми [47], хоча, інколи, випущена гра може не мати фінальної концепції, як наслідок, для гравця це буде здаватися набором погано зібраних ідей, і скоріш за все така гра не буде успішною.

Так само, як і фінальність концепції та дизайну, важливий і попередній досвід розробників.

Джордан Мекнер ділиться своїм підходом на прикладі однієї з власних ігор – першої «Prince of Persia», незважаючи на те, що грі вже виповнилося більше 30 років, методи та підходи описані при її створенні актуальні й до нині, особливо для початківців [48].

Якою б структурно простою або складною не була б гра, для залучення та зацікавлення гравців комп'ютерна гра повинна мати ефекти занурення та потоку. Оскільки мозок людини сприймає віртуальний світ так само, як і реальний, то загалом, ефекту занурення можна досягти, створивши світ гри наближений до реального.

З ефектом потоку набагато складніше, адже не існує універсального потоку для всіх гравців. Гравці різняться за навичками та досвідом, що цікаво одній людині може бути нецікаво для іншої. Загалом правила для формування потоку полягають в тому, що навички геймера повинні бути пропорційні викликам, з якими він стикається у грі. При надто простих викликах для досвідченого гравця йому буде нудно, недосвідчений гравець при складних викликах буде відчувати тривогу та навіть стрес, в обох випадках ймовірність того, що гравець покине гру є високою [49, 50].

Графік залежності навичок від викликів для досягнення ефекту потоку зображено на рисунку 1.2.

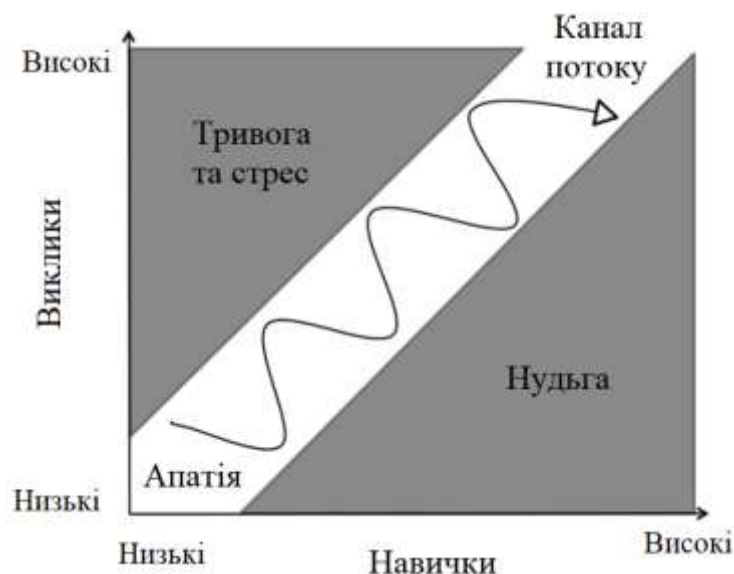


Рисунок 1.2 – Ефект потоку

Інша важлива річ, яка може покращити занурення або імерсивність, а також збільшити емоційний відгук у геймера – звуковий супровід. Так само, як і у кінематографі сцени супроводжуються звуками для підсилення ефекту, так і для відеоігор працює такий самий підхід. Наприклад, основний сюжет гри «Assassin's Creed IV: Black Flag» розгортається в Золоту добу піратства, в Карибському морі на декількох островах на початку XVIII століття, якщо головний герой збере текст пісень, то він зможе наказати своїй команді заспівати шанті під час морських подорожей [51].

Необхідно розуміти, як ІТ-індустрія буде розвиватися в найближчому майбутньому, багато серйозних проєктів в розробці займають від двох до семи років, а за цей час відбудеться багато змін і деякі елементи стануть неактуальними.

Найбільш реалістичним прогнозом на даний час виглядає більше залучення систем штучного інтелекту для розробки та функціонування відеоігор [52].

Іншою перспективною розробкою, яка дозволить значно покращити ігрові світи може бути квантові комп'ютери.

#### 1.4 Постановка задачі дослідження

Спираючись на результати попереднього аналізу актуальність дослідження полягає в тому, що впровадження методів розроблення ігрових світів для їх створення для гравців є одним зі способів проведення дозвілля, вивчення чогось нового та при обмеженому застосуванні є ефективним методом профілактики для покращення здоров'я.

Об'єктом дослідження є відеоігрова індустрія.

Метою дослідження є реалізація та вивчення методів створення ігрових світів різного рівню складності та наповнення.

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати сучасний стан питання що розглядається;
- проаналізувати літературні джерела щодо апробації результатів застосування методів для розроблення ігрового світу;
- виявити особливості методів для розроблення ігрового світу;
- здійснити вибір інструментальних засобів для програмної реалізації відеогри на основі обраних методів;
- програмно реалізувати відеогру на основі обраних методів;
- протестувати розроблений застосунок та провести аналіз результатів;
- виявити перспективи подальшої роботи.

## 2 ДОСЛІДЖЕННЯ ВИБРАНИХ МЕТОДІВ РОЗРОБЛЕННЯ ІГРОВОГО СВІТУ

### 2.1 Розмірність простору світу гри

#### 2.1.1 2D

Одним із найпростіших методів для створення відеогри [53] підхід 2D. Для цього необхідні лише два типи графічних об'єктів – текстура та спрайт.

Текстура – це растрове зображення, яке відтворює візуальні властивості будь-яких поверхонь або об'єктів.

Спрайт – представляє собою комп'ютерний об'єкт з нанесеною поверхнього текстурою.

У залежності від складності гри вона може мати лише програмну реалізацію відображення для графічних об'єктів, реалізацію взаємодії з об'єктами через натискання на них та фізичні властивості притаманні реальному світу. Наприклад, для того щоб симулювати гравітацію у грі необхідно встановити коллайдери.

Коллайдер – невидимий елемент у відеогрі, який відстежує зіткнення або колізії, це потрібно для того щоб персонаж не провалювався через землю або не проходив через стіни, що, в свою чергу, руйнувало б задум розробників. Для побудови елементарної гри достатньо встановити будь-яке зображення як текстуру заднього фону, щоб вона контрастувала з іншими об'єктами зображеними на екрані, та створити персонажа з оточенням методом піксельної графіки (рис. 2.1).



Рисунок 2.1 – Гра «T-Rex Game»

### 2.1.2 2.5D або псевдо 3D

Даний метод є наступним кроком розвитку методу 2D, різниця полягає у створенні ефекту об'єму та наближенні до тривимірного простору [54]. Більш того, 2.5D ігри потребують більше ресурсів для функціонування та, в більшості випадків, створюються за допомогою ігрових рушіїв. Усі внутрішньо ігрові ресурси називають асетами [9, 10].

Також, для додаткової обробки деяких графічних елементів можуть використовуватися шейдери [9, 10].

Ігровий асет – цифровий об'єкт, який переважно складається з однотипних даних, яка представляє частину ігрового контенту і має деякі властивості, є неподільною сутністю.

До асетів відносяться всі дані, що використовуються та оброблюються відеогрою:

- геометричні моделі;
- текстури;
- окремі звуки та музичні треки;
- тексти діалогів;
- анімаційні дані;
- моделі поведінки ігрового штучного інтелекту та інше.

Сукупність всіх асетів представляє собою контент гри, тоді як ігровий рушій є сукупністю всіх механізмів, що обробляють контент.

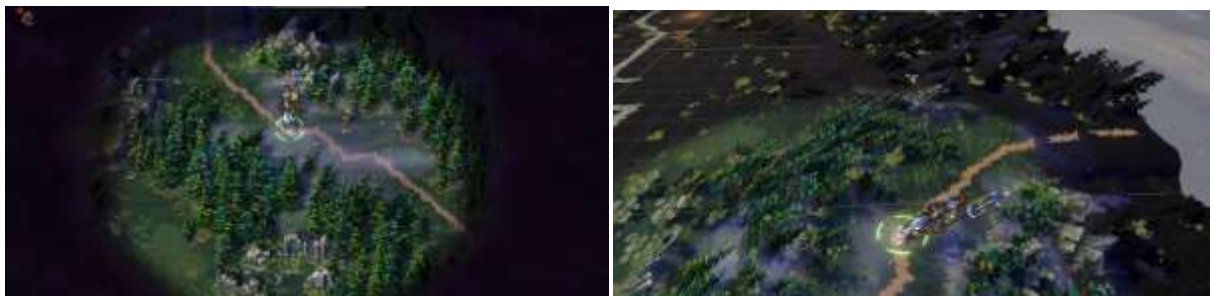
Шейдер – це програма призначена для виконання графічним процесором, представляє собою набір інструкцій по обробці певних ефектів. Наприклад, ефекти затінення, віддзеркалення, розсіювання світла, зміщення поверхні та інші.

Один зі способів створення 2.5D гри є використання 2D асетів і накладання їх шар за шаром для створення паралакс-ефекту (рис. 2.2).



Рисунок 2.2 – 2.5D з паралакс-ефектом, гра «Ori and the Blind Forest»

Другий підхід – використання тривимірного простору з розміщенням спрайтів типу Billboard, спрайт постійно повернутий до камери (рис. 2.3).



а)

б)

Рисунок 2.3 – Розміщенням спрайтів типу Billboard у тривимірному просторі, гра «Songs of Conquest»: а) те що бачить геймер; б) вид збоку при розробці

Ще одним варіантом побудови 2.5D є створення гри у 3D оточенні з обмеженням гравця лише за двома осями (рис. 2.4).



Рисунок 2.4 – Об’ємні предмети оточення та персонаж, гра «Trine 4: The Nightmare Prince»

### 2.1.3 3D

Для побудови ігрового світу у 3D змінюється концепція. Оскільки тепер є три осі для руху, відповідно гравець має більше ступенів свободи як і в управлінні персонажем, так і в управлінні камерою.

Тривимірні відеоігри створюються на основі 3D моделей [9, 14], моделі в свою чергу складаються з будівельних елементів, які називають полігонами. Полігони, або багатокутники, представляють собою плоскі трикутники, чотирикутники, п’ятикутники та інші багатокутники різної форми, які в комбінації створюють полігональну сітку, так званий меш. Чим більше полігонів міститься у моделі тим більше ресурсів необхідно комп’ютеру для обробки, але за рахунок збільшення кількості полігонів збільшується і деталізація.

Різницю між високо деталізованою та низько деталізованою полігональною моделлю зображено на рисунку 2.5.

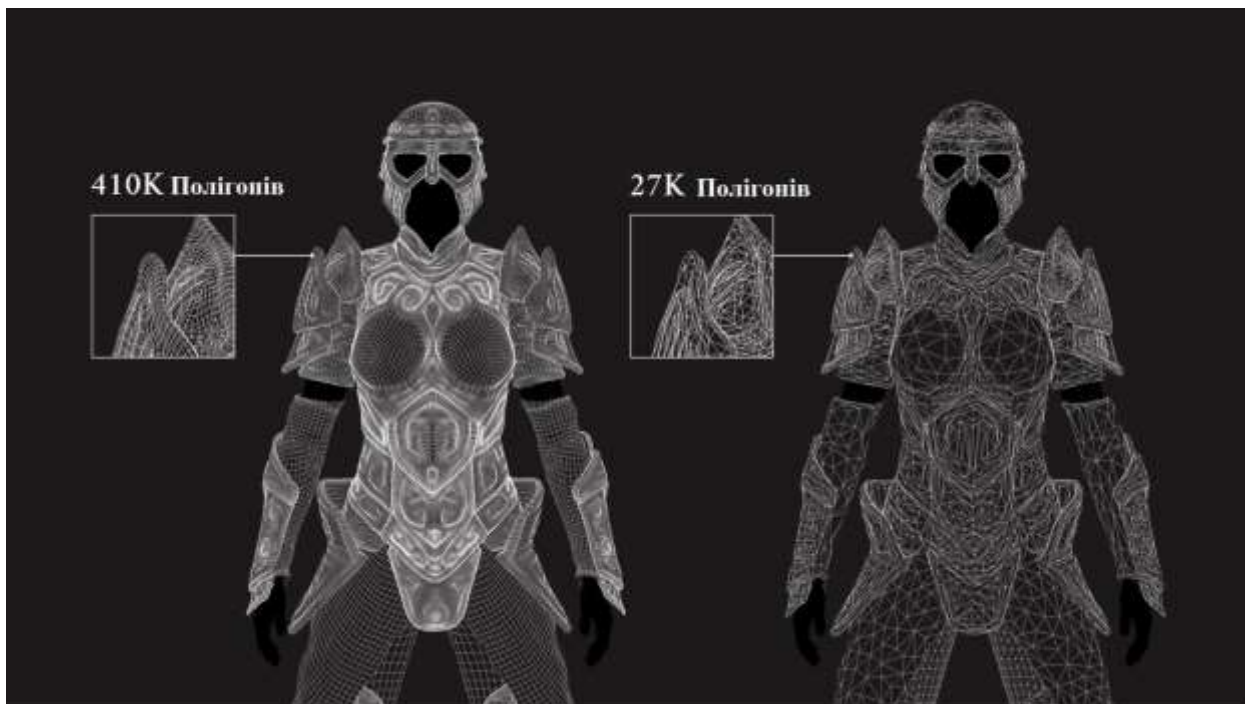


Рисунок 2.5 – Різниця між високою деталізацією броні у 410 000 полігонів та низькою деталізацією у 27 000

Для того щоб модель набула кольору та ілюзії рельєфу накладаються текстури (рис. 2.6). Якість текстурованої поверхні визначається текселями – кількістю пікселів на мінімальну одиницю текстури [55].



Рисунок 2.6 – Модель броні з встановленою текстурою на персонажі

При накладанні текстур на полігональні об'єкти може використовуватися рельєфне текстурування [56, 57], воно потрібно для додання більш реалістичного й насиченого вигляду поверхні об'єктів, наприклад, пори та зморшки на шкірі.

Кількість полігонів в моделі та якість текстур – взаємопов'язані речі. При низькій деталізації полігональної сітки але високій якості текстур модель буде виглядати плоскою, при високій деталізації полігональної сітки та низькій якості текстур – модель буде виглядати нереалістичною.

Процес обробки полігонів та текстур потребує різних ресурсів. Деталізована полігональна сітка потребує великої потужності графічного процесору, в той час як текстури залежать більшою мірою від постійної та оперативної пам'яті [9, 10, 14].

З плином часу, технології та платформи розвиваються, надаючи все більше можливостей та прибираючи певні обмеження, порівняння вигляду та деталізації одного й того ж персонажу для різних поколінь консолей показано на рисунках 2.7, 2.8.



а)



б)

Рисунок 2.7 – Порівняння деталізації персонажа, Кратос, серія «God of War»:

а) «God of War II» 2007 року випуску; б) «God of War» 2018 року випуску



а)



б)

Рисунок 2.8 – Порівняння деталізації мешу персонажа, Кратос, серія ігор «God of War»: а) загалом 5 700 полігонів, 1 200 з них для голови, 5 текстур;  
 б) загалом 80 000 полігонів, 32 000 з них для голови, 140 текстур

Для голови використовується більша кількість полігонів ніж для інших частин тіла, це зумовлено тим, що обличчя має велику кількість м'язів, і для реалістичного відображення міміки або не підсвідомих скорочень м'язів такі зони як область очей, ніс та губи потребують значної деталізації [58].

Для створення 3D моделей використовуються такі середовища як 3DS Max, Maya, Zbrush, Blender 3D та інші [59]. Якщо необхідно перенести реальний об'єкт у віртуальний простір використовуються технології фотограмметрії та 3D сканування [60, 61].

Також існує інший принцип створення 3D світу – за допомогою вокселів [62].

Воксель – аналог пікселя для тривимірного простору. Тривимірний світ побудований за допомогою вокселів потребує значно більше ресурсів для обробки ніж аналогічний двовимірний. Відеоігри використовують вокселі для створення моделей руйнувань та ландшафту який складно створити на основі карти висот, наприклад печери.

На даний час створити гру повністю на основі вокселів з деталізацією наближеною до реального світу технічно неможливо. Попри це, воксельні ігри часто надають можливість будувати та змінювати світ гри, хоча й з доволі примітивним відображенням (рис. 2.9).



Рисунок 2.9 – Побудова колізею у «Minecraft»

#### 2.1.4 4D

Для створення 4D гри необхідно додати до тривимірної гри ще один вимір, або часовий, або просторовий.

У випадку якщо необхідно мати певні маніпуляції із часом, то треба зберігати всі попередні стани світу гри [63].

Оскільки 3D відеоігри самі по собі вже є структурно складними, вони мають велику кількість ресурсів, тому зберігати можна лише останні декілька секунд ігрового процесу (рис. 2.10).



Рисунок 2.10 – Перемотка часу назад для уникнення падіння,  
гра «Prince of Persia: The Forgotten Sands»

Створення відеогри з чотиривимірним простором потребує модифікацію підходу 3D. В основу ідеї створення ігор простору більшої розмірності покладено науково-фантастичний роман Едвіна Ебботта «Флетландія», у ньому йдеться про історію квадрата який переносять у тривимірний світ [64]. Для того щоб отримати двовимірне зображення маючи тривимірний об'єкт треба його розрізати та відобразити проекцію, для відображення тривимірного зображення необхідно мати розріз та проекцію чотиривимірного об'єкту.

Оскільки людство живе у тривимірному просторі нам важко уявити простір більшої розмірності, але математично та технічно це можливо. Для побудови зображення у 3D використовуються пласкі структури, багатокутники різної форми, які називають полігонами. Відповідно, для 4D треба використовувати об'ємні фігури – багатогранники. Будівельними блоками для 4D моделі є тетраедр, паралелепіпед, додекаедр та інші багатогранники, які за аналогією можна назвати поліхідронами. Для наочності обрано гру «Miegakure», як приклад [65].

Проблема полягає у тому, що створювати 4D моделі так само як і 3D моделі неможливо. Вирішити її можна за допомогою процедурної генерації на основі 3D мешу (рис. 2.11).

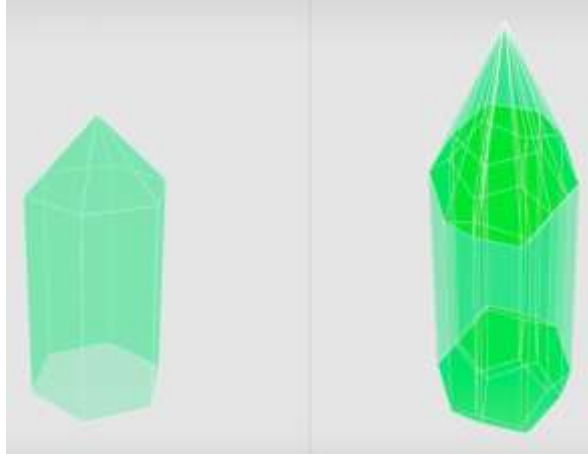


Рисунок 2.11 – 3D та 4D моделі кристалу

На рисунку 2.12 показано що бачить гравець, а саме – відображення через 2D проекцію 3D зрізу 4D об'єкту. Здається що деякі об'єкти ніби парять у повітрі, хоча насправді це розріз 4D об'єкту, який встановлено на землю.



Рисунок 2.12 – Застосування однієї моделі кристалу, за рахунок різної орієнтації та 3D розрізів у різних місцях складається враження різних об'єктів

Позиціонування об'єктів у просторі та переміщення персонажів у грі відбувається так само як і в інших іграх меншої розмірності простору – через систему координат (рис. 2.13).

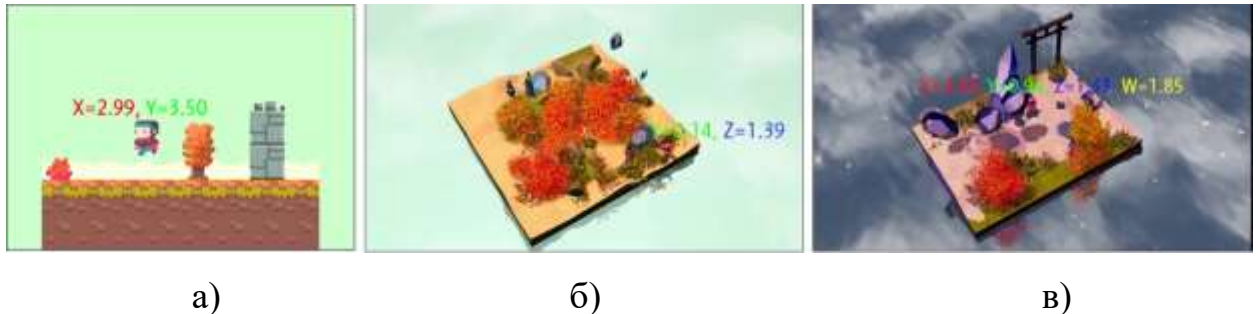


Рисунок 2.13 – Координати персонажа у просторі:

а) два виміри; б) три виміри; в) чотири виміри

## 2.2 Постанова камери

### 2.2.1 Від 1-ї особи

Постанова камери від першої особи – це те, до чого люди звикли у повсякденному житті, бачити на власні очі, в іграх – через очі персонажа яким керує гравець.

У відеоіграх такий підхід дозволяє зекономити на певних нюансах, таких як повне моделювання голови протагоніста та зменшення кількості анімацій яких не видно [9, 14], інколи такий підхід ще називають камера з руками.

Те що бачить гравець при встановленні камери від 1-ї особи та те як це виглядає зі сторони показано на рисунку 2.14.



а)



б)

Рисунок 2.14 – Сцена з різних ракурсів, гра «Wolfenstein II: The New Colossus»: а) вид від 1-ї особи; б) вид збоку

### 2.2.2 Від 2-ї особи

Геймер бачить світ відеогри так само – з очей, але не того персонажа яким керує. Через те що гравець не має чіткого відчуття де знаходиться камера він може почуватися трохи дезорієнтованим.

На рисунку 2.15 зображено встановлення камери від 2-ї особи, геймер керує помаранчевою машиною.



Рисунок 2.15 – Вид від 2-ї особи, гра «Driver: San Francisco»

Для того щоб цей підхід працював камера повинна постійно слідувати за гравцем, адже, в іншому випадку, він вийде з поля зору камери та перестане орієнтуватися у просторі [66–96].

### 2.2.3 Від 3-ї особи

У реальному світі найближчою аналогією до виду від третьої особи буде ситуація, коли людина рухається дивлячись в екран на якому показується зображення з відеокамери, в полі зору якої людина знаходиться (рис. 2.16).



Рисунок 2.16 – Вид від 3-ї особи у реальному житті: а) установка, вивід зображення на VR окуляри; б) поле зору камери

Відеоігри від 3-ї особи можуть мати фіксовану або вільну камеру [9, 10, 14]. Сама камера встановлюється недалеко від персонажа яким керує гравець, якщо камера вільна, то вона може рухатися за сферо подібною траєкторією.

Фіксована камера використовується коли за задумом розробників треба обмежити гравця для досягнення певного ефекту або для посилення сприйняття сцени (рис. 2.17).



Рисунок 2.17 – Фіксована камера візуально перетворює 3D гру на 2.5D,  
гра «NieR: Automata»

Майже всі ігри у 2D та 2.5D мають фіксовану камеру від 3-ї особи. Вільна камера використовується коли світ гри є структурно складним і геймеру треба розуміти свою орієнтацію в просторі.

Функціонально вид від 3-ї особи дозволяє мати точки зору недоступні для виду від 1-ї особи, наприклад для погляду за кути залишаючись непоміченим для ворогів (рис. 2.18).



Рисунок 2.18 – Камера розташована вище за героя, тому гравець споглядає ворогів, а вони його не помічають, гра «Ghost of Tsushima»

#### 2.2.4 Ізометрія

Саме поняття ізометрії в комп'ютерних іграх відноситься до того чи іншого виду паралельної проекції тривимірних об'єктів на двовимірну площину з особливим розташуванням камери [9, 10]. Кут огляду зміщений так, що дозволяє створювати ефект тривимірності і дозволяє показати деякі деталі оточення, які не видно при виді зверху або збоку.

Функціонально, поле зору гравця значно збільшується у порівнянні з видом від 3-ї особи і він може бачити більше подій гри, наприклад, те що відбувається за спиною персонажа.

При застосуванні цього підходу гравець бачить гру від 3-ї особи, але камера встановлюється на значно більшій відстані від персонажа яким керує гравець (рис. 2.19).



Рисунок 2.19 – Ізометрична постановка камери, гра «Tunic»

Також, ізометрична проекція дозволяє наповнити гру великою кількістю секретів та захованих предметів [97, 98]. Оскільки не видно, що знаходяться за такими об'єктами як стіни або ліс, за ними можна сховати секретні кімнати або якісь речі, які допомагають гравцю проходити гру.

## 2.3 Анімація

### 2.3.1 Спрайтова анімація

Спрайтова анімація використовує принципи традиційної анімації, а саме – намальовані один за одним кадри, які при швидкому відтворенні складаються у відео, але замість кадрів усієї сцени використовується набір спрайтів для анімації конкретного елемента гри (рис. 2.20) [99].



Рисунок 2.20 – Набір спрайтів для анімації бігу персонажа з гри «Mega Man»

Для створення наборів спрайтів для анімації спочатку створюються ключові спрайти, на їх основі створюються проміжні спрайти. Чим більше проміжних спрайтів буде у анімації, тим плавніше вона буде. Можна використати метод інтерполяції для створення проміжних спрайтів.

Спрайтова анімація більш поширена у використанні для створення 2D ігор, але у 3D її теж використовують. Вона використовується не для анімації персонажів чи об'єктів, а для анімації ефектів. Наприклад, ефекту горіння вогню (рис. 2.21) [100].



Рисунок 2.21 – Рендер полум'я на основі спрайтів у 3D просторі

### 2.3.2 Векторна анімація

Векторна анімація використовує ті ж принципи що і спрайтова, відмінність полягає у тому, що для анімації використовуються векторні зображення, а не растрові, які лежать в основі спрайтів [101].

Векторне зображення – це зображення яке описується математичною формулою, завдяки цьому воно не пікселізується та зберігає свою якість незалежно від трансформації (рис. 2.22).



Рисунок 2.22 – Різниця між растровим і векторним зображенням та їх відповідні формати

### 2.3.3 Скелетна анімація

Для створення скелетної анімації треба спочатку створити полігональну сітку, при анімації сітка буде деформуватися створюючи рухи. Після створення полігональної сітки, її також називають мешем, встановлюються кістки скелету процесом монтажу [102], rigging (рис. 2.23).

Після монтажу скелета треба прив'язати меш до кісток. Такий процес називають скінінгом, він полягає у встановленні коефіцієнтів залежності для полігонів від конкретної кістки [103]. Ваговий коефіцієнт визначає ступінь деформації мешу пересуванням кістки. Кістки мають ієрархію – переміщення батьківської кістки змінить положення дочірньої.



Рисунок 2.23 – Різні варіанти зображення кісток, кістки для обличчя відсутні

Скінінг 3D моделі зображено на рисунку 2.24, колір визначає коефіцієнт залежності, від 0 до 100%, від синього до червоного кольору відповідно.

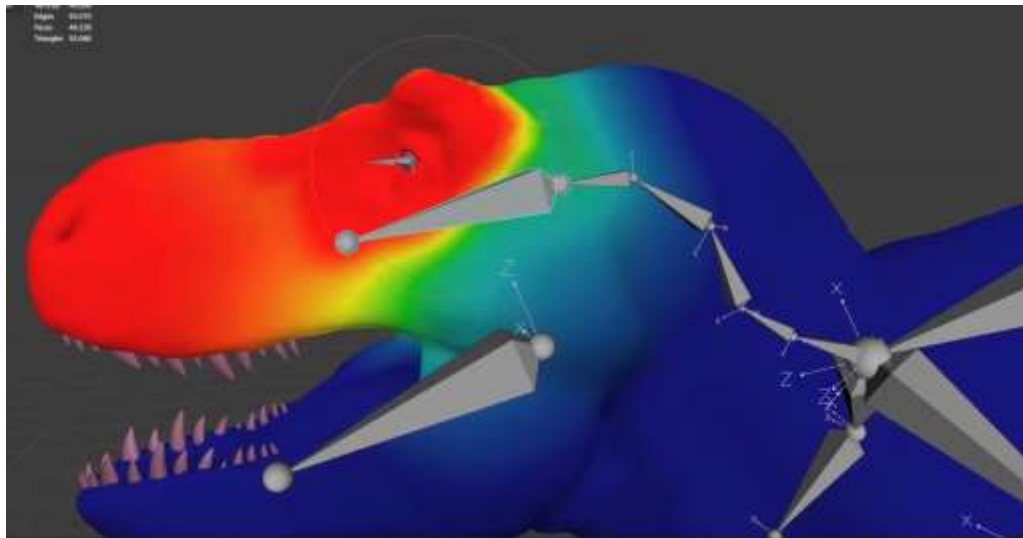


Рисунок 2.24 – Встановлення коефіцієнтів для верхньої щелепи динозавра

Скелетну анімацію можна створювати вручну розставляючи кістки за ключовими кадрами та створюючи проміжні, чи за допомогою автоматизованих методів.

Одним із таких методів є процес захоплення руху, анімації на його основі дуже реалістична [104]. Для нього використовуються живі актори та спеціальне обладнання (рис. 2.25).



Рисунок 2.25 – Захоплення руху для гри «The Last of Us Part II»

Іншим способом, який значно пришвидшує процес створення скелетної анімації є використання штучного інтелекту [105].

Скелетну анімацію також можна використовувати для анімації у 2D іграх, перед тим як почати анімацію об'єкт анімації необхідно розділити на частини та створити полігональну сітку.

На рисунку 2.26 зображено персонажа у зібраному та розібраному вигляді.

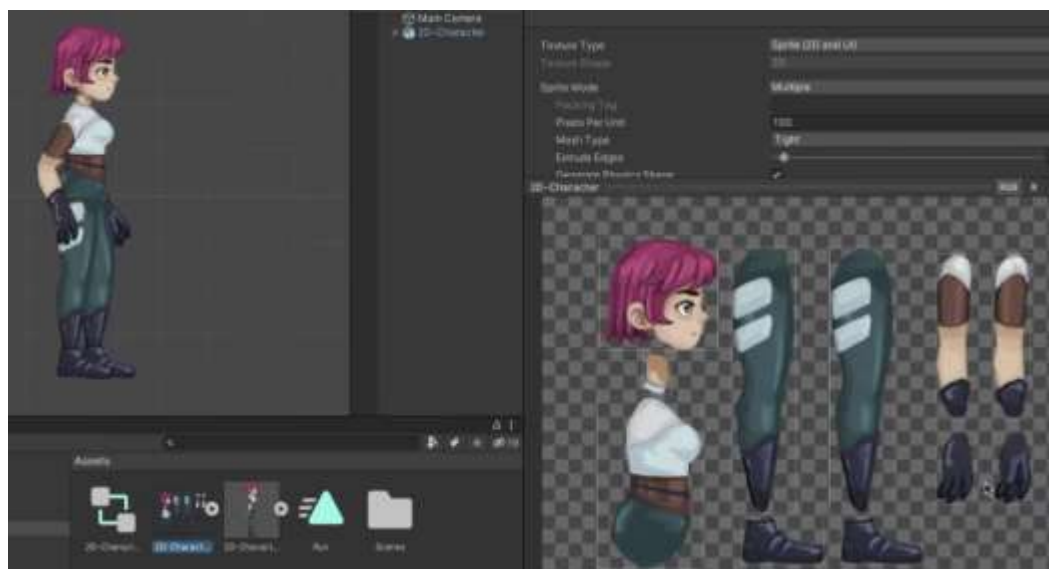


Рисунок 2.26 – 2D персонаж у зібраному вигляді та у вигляді компонентів

### 2.3.4 Stop-motion анімація

Використання stop-motion анімації більш звичне для кінематографу і в іграх використовується рідко через достатньо великий об'єм робіт. Технічно вона дуже схожа на традиційну анімацію, але кожен кадр не малюється а фотографується, потім сцена мінімально змінюється і робиться новий кадр. Її можна створити використовуючи ляльок, пластилін, папір, конструктор LEGO та будь-які інші предмети.

На рисунку 2.27 зображено фрагмент з відеогри, яку повністю анімовано, включно з внутрішньо ігровими відео, за допомогою stop-motion анімації.

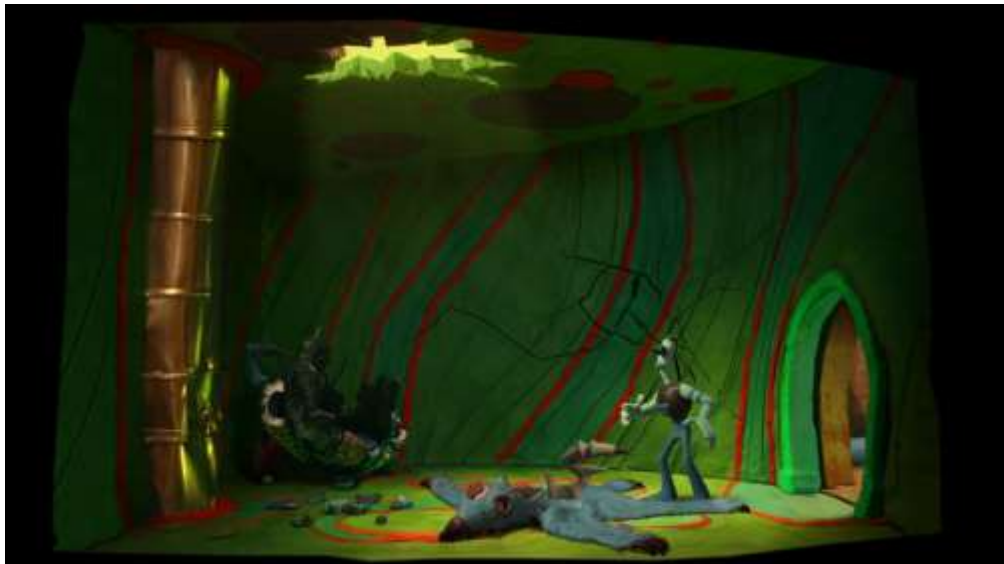


Рисунок 2.27 – Пластилінова stop-motion анімація, гра «Armikrog»

## 2.4 Взаємодія розробників з гравцем

### 2.4.1 Спільнота

У кожної більш-менш вдалої гри з'являється декілька десятків тисяч і більше шанувальників, які можуть створювати різний контент на основі сюжету або світу гри [9, 14, 15].

Таким контентом може бути:

- картини з персонажами або пейзажами із гри;
- музикальні твори які присвячені якомусь елементу гри;
- текстові та ілюстровані оповідання що розширюють світ гри, але не є офіційними;
- неофіційні ігри з деякими персонажами та елементами із гри;
- модифікації для гри.

Розробники намагаються прописувати обмеження для створення контенту у ліцензійних контрактах, але навіть судові процеси не зупиняють людей від створення нового контенту для улюбленої гри [106]. Найбільш поширені обмеження стосуються використання оригінальних елементів інтелектуальної власності розробників та їх подальшої монетизації.

На рисунку 2.28 зображено порівняння між оригінальною версією гри від розробників та модифікованою версією від шанувальників.



Рисунок 2.28 – Оригінальна гра справа та модифікована версія зліва, гра «GTA 5»

### 2.4.2 Щоденники розробників

Розробники можуть публікувати результати своєї роботи час від часу, показуючи та описуючи деталі та контент що був змінений чи доданий. За допомогою цього методу вони потім можуть отримати зворотній зв'язок та покращити деякі елементи гри (рис. 2.29).

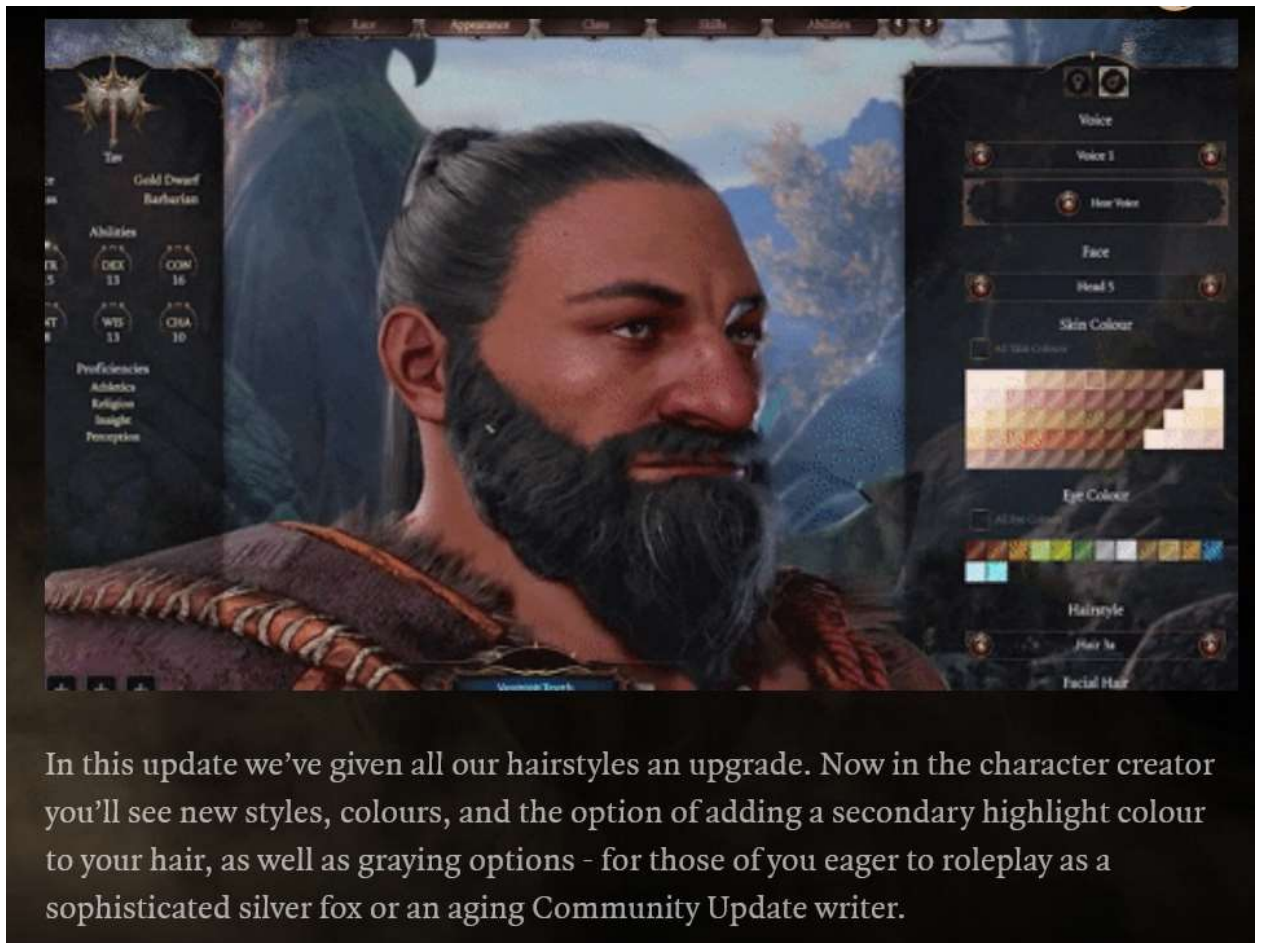


Рисунок 2.29 – Фрагмент з щоденнику розробників, гра «Baldur's Gate III»

### 2.4.3 Досягнення

Досягнення представляють собою певні дії які необхідно виконати гравцю, їх можна умовно поділити на три категорії.

Перша категорія – це досягнення за проходження сюжету гри, дуже часто розробники можуть їх сховати щоб не розповідати заздалегідь гравцю неочікувані повороти у сюжеті гри.

Друга категорія – досягнення які видаються за пошук певних предметів або якоїсь їх кількості.

Третя категорія – досягнення за специфічні дії гравця. Наприклад, пройти рівень гри без отримання ушкоджень або за визначений час.

На рисунку 2.30 зображено момент отримання досягнення.



Рисунок 2.30 – Момент отримання досягнення,  
гра «Uncharted 4: A Thief's End»

Також, впровадження досягнень є способом розробників пожартувати над якоюсь ситуацією або самими собою, як зображено на рисунку 2.30. Для того щоб отримати досягнення «Страх сцени» треба у доволі динамічний момент, коли гра спонукає бігти вперед, припинити керувати персонажем і нічого не робити 30 секунд. Жарт полягає в тому, що у розробників під час живої демонстрації гри на виставці Е3 вимкнувся контролер саме на цьому моменті [107].

#### 2.4.4 Секрети та пасхалки

Метод ховання деяких елементів гри щоб не всі гравці їх знайшли зовсім не новий. Перше пасхальне яйце або пасхалка, було додано до гри «Adventure» розробником, який працював на Atari. Він заховав своє ім'я, це пов'язано з тим, що раніше компанії не публікували імена розробників [108].

Зараз заховані елементи не завжди такі примітивні – це можуть бути посилення на історичні події, елементи культури, в тому числі сучасної, політичні події та інше.

На рисунку 2.31 зображено момент знаходження пасхалки у грі.



Рисунок 2.31 – Секретний вихід з рівня, посилення на Мері Попінс, гра «Hitman 2»

#### 2.4.5 Події

Події, або івенти, можуть застосовуватися розробниками для розширення аудиторії гри, або у випадку довгого існування гри – для активізації гравців щоб вони повернулися у гру [109].

Події можуть бути тематичними під якісь свята, наприклад Різдво чи день Святого Валентина, або доволі практичними при зміні якогось великого сегменту гри (рис. 2.32).



а)



б)

Рисунок 2.32 – Гра «Fortnite»:

а) звичайний ігровий процес; б) ігровий процес під час івенту

#### 2.4.6 Додатковий контент та оновлення

Оскільки індустрія відеоігор доволі динамічна, у порівнянні з кінематографом, та більше схожа на ІТ-індустрію, їй потрібно оновлюватися, впроваджувати нові технології та виправляти минулі помилки.

Розробники ігрових світів стараються покращити досвід гравця прибираючи прикрі помилки або додаючи новий контент. Деякі ігри дуже сильно змінюються після років оновлення та додавання нового контенту у порівнянні із початковою версією (рис. 2.33).



а)



б)

Рисунок 2.33 – Інтерфейс космічного корабля, гра «No Man's Sky»:

а) вигляд інтерфейсу на релізі у 2016 році; б) вигляд інтерфейсу у 2022 році

## 3 ПОРІВНЯННЯ ВИБРАНИХ МЕТОДІВ РОЗРОБЛЕННЯ ІГРОВОГО СВІТУ

### 3.1 Вибір інструментальних засобів для реалізації вибраних методів

Для реалізації програмного застосунку, а саме комп'ютерної гри, обрано мову програмування C++, використано бібліотеку SDL та середовище розробки NetBeans IDE.

Мова програмування C++ є об'єктно-орієнтованою мовою програмування та включає такі поняття, як класи, успадкування, поліморфізм, абстракція даних та інкапсуляція, які дозволяють повторне використання коду та роблять програму ще надійнішою. Більш того, таким чином можливо вирішувати проблеми реального світу, розглядаючи дані як об'єкт.

C++ створено як скомпільовану мову, це означає що вона перекладається на машинну мову, яку може зрозуміти безпосередньо система, що робить комп'ютерну програму високоефективною. Для відеоігор також можна використовувати інтерпретовані мови програмування, але використання скомпільованих мов програмування значно підвищує продуктивність, використовуючи менше ресурсів. Також, швидкодія скомпільованих мов значно вище за інтерпретовані, для великих та складних ігор це є важливим елементом для оптимізації.

Інша перевага C++ надає програмісту повний контроль над керуванням пам'яттю. Це можна розглядати і як актив, і як зобов'язання, оскільки це збільшує відповідальність програміста за керуванням пам'яттю. Ця концепція реалізована за допомогою DMA на основі вказівників.

Ще одною сильною стороною C++ є можливість до легкого масштабування застосунків, для відеоігор це важливо коли гра є великою та містить багато елементів.

Також C++ дозволяє маніпуляцію на достатньо низькому рівні, наприклад, для визначення кількості ядер процесору та навантаження на них для досягнення більшої продуктивності, і як результат, оптимізації роботи застосунку.

SDL – це безкоштовна кросплатформенна бібліотека для розробки програмного забезпечення, призначена для надання апаратного рівня абстракції для мультимедійних апаратних компонентів комп'ютера. Розробники програмного забезпечення можуть використовувати його для створення високопродуктивних комп'ютерних ігор та інших мультимедійних програм, які можуть працювати на багатьох операційних системах, таких як Android, iOS, Linux, macOS та Windows.

SDL може керувати графікою, аудіо, пристроями введення, дисковою підсистемою, потоками, завантаженням спільних об'єктів, мережею та таймерами. Для 3D-графіки він може працювати з контекстом OpenGL, Vulkan, Metal або Direct3D11, старіша версія Direct3D9 також підтримується. Поширеною помилкою є те, що SDL – це ігровий рушій, хоча це не так.

Дану бібліотеку обрано через те, що вона підходить для безпосереднього створення ігор або опосередковано використовується рушіями, побудованими на її основі. Також вона не містить фізичного рушія для симуляції елементів реального світу, наприклад гравітації.

NetBeans IDE представляє собою модульне середовище розробки, де модулями виступають такі мови програмування як Java, Python, PHP, JavaScript, C, C++ та інші, і, за необхідності, можна встановити необхідний мовний пакет. Також, інтерфейс NetBeans IDE є інтуїтивно зрозумілим та не такий навантажений у порівнянні з Microsoft Visual Studio. Саме середовище розробки краще оптимізовано ніж конкуренти. Компілятори та інтерпретатори для відповідних мов програмування встановлюються так само модульно як і мовні пакети.

### 3.2 Етапи програмної реалізації вибраних методів для розроблення ігрового світу

Для розуміння об'ємів роботи програмної реалізації спочатку треба визначити які методи будуть використані при розробці ігрового світу, та наскільки структурно складною буде гра.

Платформою для цільової розробки відеогри обрано Microsoft Windows 10 Version 22H2.

Методом розмірності простору обрано 2D.

Методом постанови точки зору гравця обрано фіксований вид від 3-ї особи з видом згори. Технічно, сам комп'ютерний об'єкт камери відсутній, він потрібний для ігор які створені у 3D просторі.

Методом для анімації обрано метод спрайтової анімації.

Будь-яка відеогра має набір текстур з яких складається світ гри. Щоб визначити текстури які підходять або не підходять для створюваної гри необхідно задати сетинг.

Сетинг – це сукупність різнопланових елементів, які однозначно ідентифікують світ, де відбуваються події певного художнього твору, відеогри або настільної рольової гри. Елементами сетингу є відомості з географії, історії, часто також міфології. Зазвичай у відеоіграх сетинг задається у вигаданому всесвіті, однак можливе й цілком реальне місце у цілком реальному історичному періоді.

Сетингом гри обрано пустелю та руїни старого міста у пустелі.

На рисунку 3.1 зображено повний набір текстур що використано у програмній реалізації відеогри. Більша частина із них отримані із вільних джерел розповсюдження графічних ассетів, деякі з них створені вручну.

Для створення світу гри обрано концепцію рівнів. Це означає що кожна локація це окрема мапа, і гравець не може вільно переміщуватися між рівнями як йому заманеться.

Такий підхід визначає що рівні повинні відрізнятися один від одного.



У даній програмній реалізації обробка рівня відбувається з використанням можливостей мови програмування C++, а саме динамічна робота з пам'яттю.

По-перше, при запуску гри відбувається ініціалізація внутрішніх ресурсів програми. Після цього до функції обробки передається файл в якому розмічено структуру рівня, зображено на рисунку 3.2. Потім за допомогою перетворення символічних рядків у цілі числа створюється матриця, яка заповнюється відповідними параметрами залежно від розмітки рівня.

Рівень та всі його елементи обробляються динамічно, це зроблено для того щоб не засмічувати та не перевантажувати оперативну пам'ять. Обробляється лише той рівень на якому знаходиться гравець.

По-друге, вся графічна частина гри обробляється за допомогою бібліотеки SDL. Вона відображає вікно гри, та усі елементи в ньому, також ця бібліотека використовується для створення анімацій. Для того щоб проініціалізована матриця відобразилася відповідно до розмітки рівня встановлюються координати початкового пікселю, на основі яких буде створено графічну структуру рівня.

По-третє, SDL також використовується для обробки подій, зокрема, натискання на різні клавіші комп'ютерної миші та відповідні реакції зі сторони гри.

Сама гра працює у безкінечному циклі, і навіть якщо гравець програв він відразу може почати спочатку. Умовою виходу є або закриття гри як будь-якої іншої програми або натискання на спеціальну панель у самій грі.

Також створено систему збереження прогресу. Якщо гравець не закінчив гру, а просто вийшов з неї, то буде створено файл збереження у якому буде записано прогрес гравця враховуючи останній пройдений рівень, система збереження прогресу теж є динамічною.

У випадку коли гравець пройшов частину рівня – прогрес не буде збережено.

Якщо гравець витратить всі свої життя то гру буде програно і файл збереження буде видалено, це означає що гравцю потрібно буде починати гру з самого початку, з першого рівня.

Умовою для виграшу гравця є закінчення усіх рівнів. Для того щоб йому вдалося це зробити у грі присутні різні ресурси, на кшталт, монет та додаткових життів. Також гравець повинен використовувати підказки щоб зрозуміти де знаходяться вороги, якщо гравець обирає місце де є ворог він втрачає одне життя.

### 3.3 Тестування програмної реалізації комп'ютерної гри на основі обраних методів розроблення ігрового світу

Тестування є не менш важливим ніж процес створення, оскільки гра що була розроблена не є структурно складною, об'єм тестування у рази менший у порівнянні з великими іграми. Усі проблеми які виникли на етапі компіляції програми, такі як помилки компіляції або витік пам'яті, було виправлено.

Для запуску гри необхідно запустити її через exe-файл.

На рисунку 3.3 показано як виглядає перший рівень гри після оброблення первинних даних та встановлення текстур.



Рисунок 3.3 – Перший рівень з текстурями

Закриті клітинки відображені як покриті шаром піску, гравець може відкривати лише клітинки суміжні з відкритими, тобто відкрити закриту клітинку на іншій стороні рівня не вийде, якщо гравець не відкрив попередні.

Стартовим полем є відкрита клітинка.

Для того щоб гравець не займався нудною роботою з відкриття кожної клітинки окремо кожен раз натискаючи ліву клавішу миші створено рекурсивну функцію відкриття порожніх клітинок (рис. 3.4).



Рисунок 3.4 – Відкриття пустих клітинок в один клік

У ході тестування виявилось що таке відкриття клітинок прибирає відчуття дослідження. Експериментально встановлено, що глибина виклику рекурсивної функції щоб гра не грала сама у себе та щоб у гравця не було сильно нудних одноманітних дій є параметр який дорівнює 5 (рис. 3.5).



Рисунок 3.5 – Часткове відкриття клітинок

Цифри у клітинках виступають підказками для гравця скільки небезпек є у сусідніх клітинках, гравець повинен обходити небезпеки та не відкривати клітинки, які він вважає будуть небезпечними. Якщо він обере клітинку в якій є ворог чи пастка він втратить одне життя (рис. 3.6).



Рисунок 3.6 – Знайдено два ворога

Якщо гравець потрапив на клітинку з небезпекою вона залишається активною, тому, якщо гравець натисне на клітинку з відкритою небезпекою ще декілька разів він втратить відповідну кількість життів. При ситуації коли усі життя вичерпано гра закінчиться та геймеру доведеться починати спочатку (рис. 3.7).



Рисунок 3.7 – Смерть від одного ворога

Для попередження випадкових натискань на клітинки в яких є вороги чи пастки гравець може встановити попереджувальний знак, який заблокує відкриття певної клітинки. Кількість та дальність встановлення знаків є необмеженою, але їх можна встановлювати лише на ті клітинки, які імітують пісок. Попереджувальний знак можна зняти за необхідності.

На рисунку 3.8 зображено встановлення попереджувальних знаків, вони блокують відкриття клітинок.

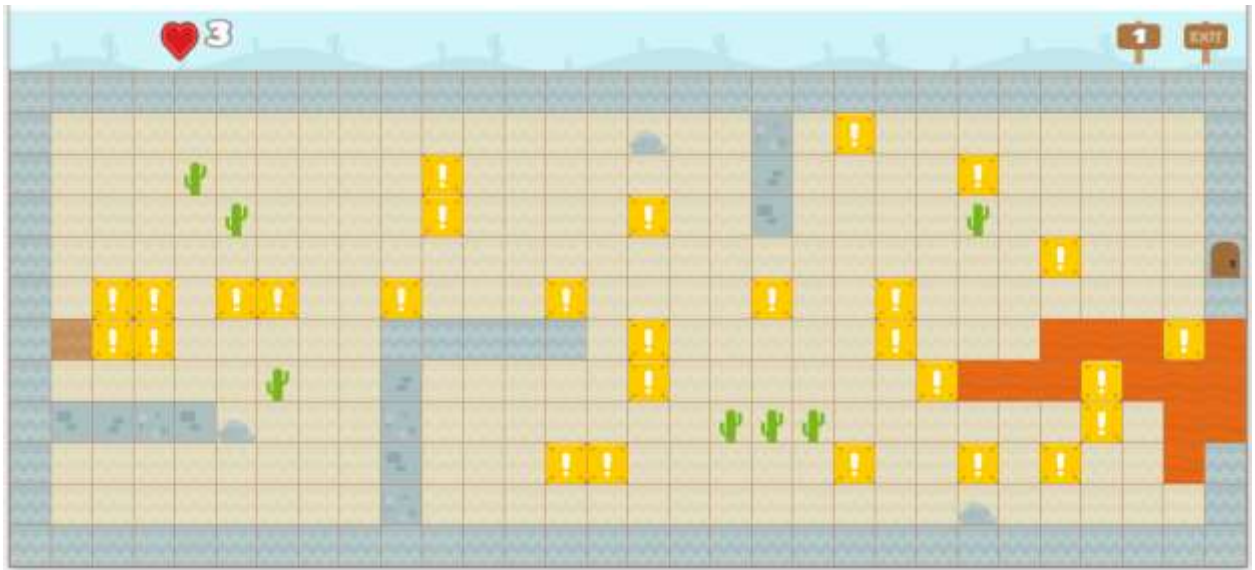


Рисунок 3.8 – Попереджувальні знаки

Для легшого проходження гри необхідно збирати ресурси. Одним з таких ресурсів є монети.

Вони варіюються за номіналом, де бронзова монета дорівнює 10 умовним одиницям, срібна – 30, золота – 50, також існують і інші види ресурсів які збільшують кількість грошей у грі.

На початку гри інтерфейс який показує скільки гравець має грошей приховано, він з'являється лише тоді коли геймер збере якийсь грошовий ресурс (рис. 3.9)

Інколи гравцю можуть зустрітися додаткові життя у вигляді серця або фляги з водою, після натискання на такий ресурс кількість життів гравця збільшиться (рис. 3.10).

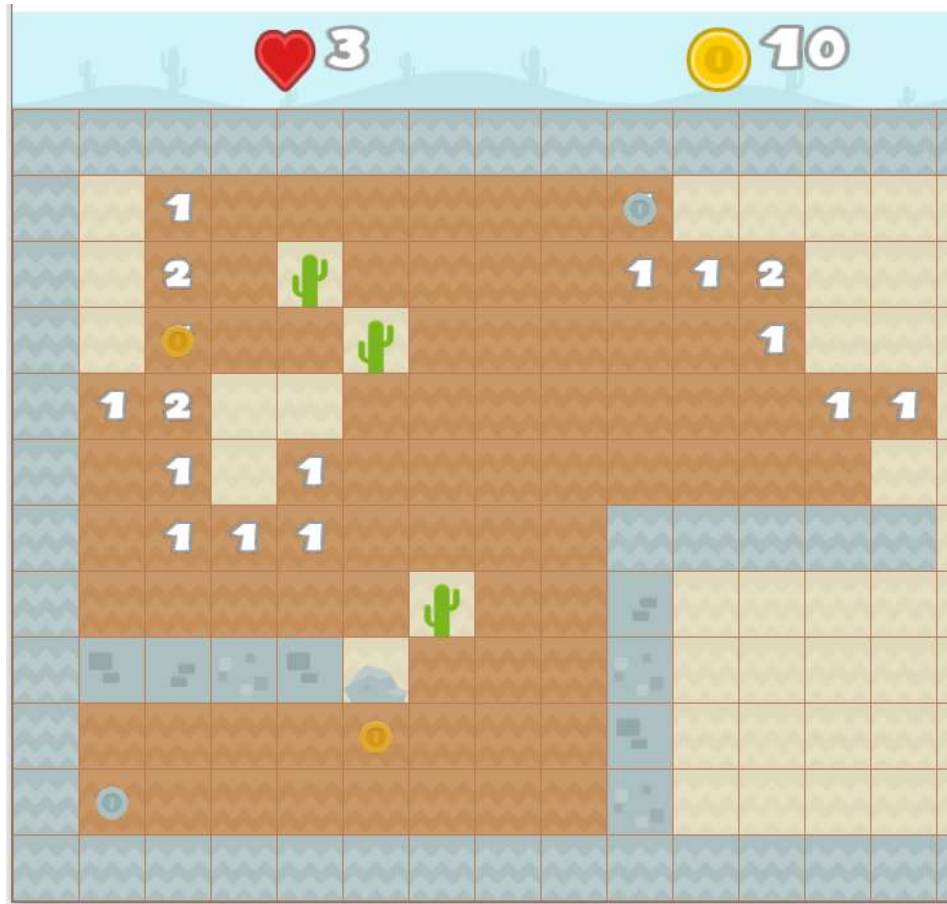


Рисунок 3.9 – Грошовий інтерфейс після підняття бронзової монети



а)



б)

Рисунок 3.10 – Збільшення кількості життів гравця: а) ресурс у грі;  
б) плюс одне життя

Лава також представляє небезпеку для гравця, якщо він на неї натисне то втратить одразу 3 життя (рис. 3.11). Як і з іншими небезпеками, гравець може натискати декілька разів, поки не втратить усі життя.



Рисунок 3.11 – Результат натискання на лаву

Усі небезпеки які є у грі анімовано за допомогою спрайтової анімації. Кількість спрайтів для усіх об'єктів анімації дорівнює двом. Відрізняються лише часові межі відтворення анімації та кількість повторень викликів однієї анімації.

Небезпеки, вороги та пастки, по рівню розставляє комп'ютер на основі псевдо випадкової генерації відповідно до розмітки рівня.

Для того, щоб у гравця був початок та кінець рівня вороги не розміщуються на початковій клітинці та біля виходу.

На рисунках 3.12 – 3.14 показано що генерація ворогів працює [110]. Коли рівень пройдено, і відкриті усі порожні клітинки, з'являються всі вороги та можна відкрити закриті двері у правій частині екрану.



Рисунок 3.12 – Генерація ворогів №1



Рисунок 3.13 – Генерація ворогів №2



Рисунок 3.14 – Генерація ворогів №3 та відчинений вихід з рівня

Якщо на рівні є додаткові зачинені двері то з'являється частина інтерфейсу з ключами (рис. 3.15).



Рисунок 3.15 – Нова частина інтерфейсу та зачинені жовті двері на 2-у рівні

Також рівень може бути більший за розмір екрану, для навігації по рівню у верхній його частині розміщені інтерактивні стрілочки. Один натиск переміщує поле зору гравця на 15 клітинок в заданому напрямку (рис. 3.16).

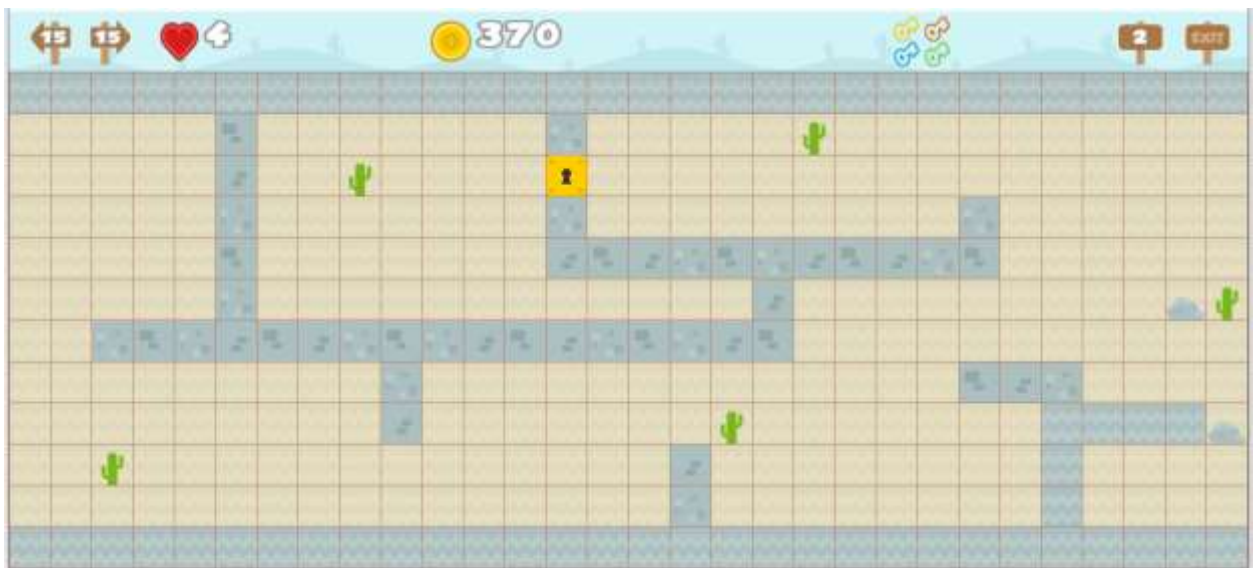


Рисунок 3.16 – Навігація по рівню

При проходженні першого рівня створюється файл збереження прогресу гравця. Процедури створення, зчитування та видалення файлу збереження працюють відповідно до задуму.

Для того щоб відчинити додаткові зачинені двері на рівні необхідно знайти ключ відповідного кольору та підібрати його (рис. 3.17). Після того як гравець натискає на ключ він залишається в нього до кінця рівня. Ключі між рівнями не переходять та не зберігаються.



а)



б)

Рисунок 3.17 – Ключ від жовтої двері: а) знайдений на рівні;  
б) піднятий ключ в інвентарі

Під час дослідження рівня гравець також може знайти динаміт, він інколи потрібен для того щоб прокласти шлях. Хоча гравець не обмежений у його використанні, він може використати його щоб просто відкрити клітинки або знищити ворогів.

Динаміт має радіус дії та при застосуванні знищує наступні речі:

- деякі стіни;
- камені;
- кактуси;
- ворогів та пастки;
- ресурси які можна підібрати, в тому числі ключі від додаткових зачинених дверей.

На рисунках 3.18 – 3.21 показано взаємодію гравця із ресурсом типу «динаміт» та його графічну реалізацію.



Рисунок 3.18 – Динаміт на рівні



Рисунок 3.19 – Відображення динаміту в інтерфейсі після підняття



Рисунок 3.20 – Зона дії та анімація динаміту



Рисунок 3.21 – Наслідок застосування динаміту, підказки щодо ворогів перераховуються автоматично

При досягненні певних сум на грошовому рахунку відкривається внутрішньо ігровий магазин, у якому гравець може докупити життів та динаміту щоб полегшити проходження гри (рис. 3.22).



Рисунок 3.22 – Внутрішньо ігровий магазин

На рисунку 3.23 зображено дієздатність магазину та значне збільшення ресурсів гравця, для збільшення рахунку використано інструменти розробника, при стандартній грі така маніпуляція неможлива.



Рисунок 3.23 – Робота внутрішнього магазину

### 3.4 Перспективи подальшої роботи

Враховуючи сумісність розробленої відеогри лише з Microsoft Windows одна з найбільших перспектив для подальшої роботи – створення версії для мобільних пристроїв та портування гри на консолі. Більш того, гру можна опублікувати та вивести в комерційну площину.

Іншим напрямком для розвитку може бути додавання контенту, наприклад звуку та більшої кількості рівнів. Таким чином, гра буде більш об'ємною але й довше буде цікавою для гравців.

Для легшої взаємодії з кодом програмну частину можна дещо рефактурувати, скоротивши деякі конструкції.

Також гра може виступати концепцією прототипом для створення більш складної гри, наприклад, у 3D.

## ВИСНОВКИ

Метою дослідження була реалізація та вивчення методів створення ігрових світів різного рівню складності та наповнення. Для розробників відеоігор необхідно знати, як можна створювати комп'ютерні ігри, щоб оптимізувати робочий процес.

Для демонстрації деяких основних методів створено невелику 2D гру на основі спрайтів та текстур з використанням спрайтової анімації. Також гра має фіксовану точку зору від 3-ї особи з видом згори. Платформою гри є Microsoft Windows 10 Version 22H2.

У ході тестування підтверджено роботу усього функціоналу гри відповідно до задуму.

Найкращий результат при проходженні гри чесно, якого вдалося досягнути, є 4-й рівень з 8-ми, це вказує, що складність гри є високою.

На основі визначеної постановки задачі, обґрунтовано вибір інструментів для програмної реалізації та створено комп'ютерну гру використовуючи мову програмування C++ та бібліотеку роботи з мультимедіа SDL. Середовищем для розробки програмного коду був NetBeans IDE.

Результати роботи апробовано у вигляді тез доповідей під час XXXVII Міжнародної науково-практичної конференції «Modern ways of solving the latest problems in science» [111].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Gamer Demographics: Facts and Stats About the Most Popular Hobby in the World. URL: <https://dataprot.net/statistics/gamer-demographics/> (дата звернення 19.09.2022).
2. Bertie the Brain Still Lives. URL: <https://www.popularmechanics.com/technology/gadgets/a23660/bertie-the-brain/> (дата звернення: 19.09.2022).
3. The First Video Game? URL: <https://www.bnl.gov/about/history/firstvideo.php> (дата звернення 19.09.2022).
4. The «Real» golden age of gaming: 1970s/1980s vs. 2010s/2020s. URL: <https://gamesforfun.com/the-real-golden-age-of-gaming-1970s-1980s-vs-2010s-2020s/> (дата звернення 19.09.2022).
5. Bartle's Player Types for Gamification. URL: <https://www.interaction-design.org/literature/article/bartle-s-player-types-for-gamification> (дата звернення 19.09.2022).
6. Hints of the 4th dimension have been detected by physicists. URL: <https://bigthink.com/technology-innovation/hints-of-the-4th-dimension-have-been-detected-by-physicists/> (дата звернення 19.09.2022).
7. What is time? The mysterious essence of the fourth dimension. URL: <https://www.newscientist.com/article/mg25433910-500-what-is-time-the-mysterious-essence-of-the-fourth-dimension/> (дата звернення 19.09.2022).
8. Reaction Time and Game Design. URL: <https://www.retrogamedeconstructionzone.com/2020/05/reaction-time-and-game-design.html> (дата звернення 19.09.2022).
9. Aleem, S., Capretz, L. F., & Ahmed, F. (2016). Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*, 4(1), 6.

10. Bethke, E. (2003). Game development and production. Wordware Publishing, Inc.
11. The cumulative morality of Far Cry 3. URL: <https://thinkchristian.net/the-cumulative-morality-of-far-cry-3> (дата звернення 19.09.2022).
12. Assassin's Creed Origins' Discovery Tour lets the beauty of Egypt shine. URL: <https://www.polygon.com/2018/2/14/17008318/assassins-creed-origins-discovery-tour-impressions> (дата звернення 19.09.2022).
13. Pascarella, L., Palomba, F., Di Penta, M., & Bacchelli, A. (2018, May). How is video game development different from software development in open source? In 2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR) (pp. 392-402). IEEE.
14. Freedman, E. (2018). Engineering Queerness in the Game Development Pipeline. *Game Studies*, 18(3).
15. Rosyid, H. A., Palmerlee, M., & Chen, K. (2018). Deploying learning materials to game content for serious education game development: A case study. *Entertainment computing*, 26, 1-9.
16. У 2021 році Activision Blizzard заробила на мікротранзакціях \$5,1 млрд. URL: <https://mezha.media/2022/02/07/activision-blizzard/> (дата звернення 19.09.2022).
17. Nvidia's Turing Architecture Explored: Inside the GeForce RTX 2080. URL: <https://www.tomshardware.com/reviews/nvidia-turing-gpu-architecture-explored,5801.html> (дата звернення 20.09.2022).
18. Intel 12th-Gen Alder Lake Pricing, Benchmarks, Specs and All We Know. URL: <https://www.tomshardware.com/news/intel-alder-lake-specifications-price-benchmarks-release-date> (дата звернення 20.09.2022).
19. AMD Cezanne benchmarks. URL: <https://www.notebookcheck.net/AMD-Cezanne-benchmarks-Ryzen-5000H-45-watt-APUs-with-Zen-3.516895.0.html> (дата звернення 20.09.2022).

20. DDR4 versus DDR5: Is it time to upgrade your RAM? URL: <https://www.reviewed.com/laptops/features/ddr4-vs-ddr5> (дата звернення 20.09.2022).

21. Shipments of hard and solid state disk drives worldwide from 2015 to 2021. URL: <https://www.statista.com/statistics/285474/hdds-and-ssds-in-pcs-global-shipments-2012-2017/> (дата звернення 20.09.2022).

22. What is HDR gaming? URL: <https://www.trustedreviews.com/explainer/what-is-hdr-gaming-2946693> (дата звернення 20.09.2022).

23. Xbox Series X review – One Year On. URL: <https://www.gamesradar.com/xbox-series-x-review/> (дата звернення 20.09.2022).

24. PS5 review - One Year On. URL: <https://www.gamesradar.com/ps5-review/> (дата звернення 20.09.2022).

25. DisplayPort 2.0: everything you need to know. URL: <https://www.digitaltrends.com/computing/displayport-2-everything-you-need-to-know/> (дата звернення 21.09.2022).

26. HDMI 2.1: features, specs and news about the latest HDMI standard. URL: <https://www.whathifi.com/advice/what-hdmi-21-everything-you-need-to-know> (дата звернення 21.09.2022).

27. Half-Life: Alyx tech analysis - a VR masterpiece that must be experienced. URL: <https://www.eurogamer.net/digitalfoundry-2020-half-life-alyx-tech-analysis> (дата звернення 24.09.2022).

28. Pokemon Go Review. URL: <https://www.ign.com/articles/2016/07/13/pokemon-go-review> (дата звернення 24.09.2022).

29. Cloud Gaming Explained. URL: <https://shadow.tech/blog/gaming/cloud-gaming-explained> (дата звернення 24.09.2022).

30. NVIDIA GeForce NOW. URL: <https://www.nvidia.com/en-eu/geforce-now/> (дата звернення 24.09.2022).

31. Amazon Luna. URL: <https://www.amazon.com/luna/landing-page> (дата звернення 24.09.2022).

32. Google Stadia. URL: <https://stadia.google.com/> (дата звернення 24.09.2022).

33. Metaverse Gaming is the Talk of the Town! URL: <https://www.analyticsinsight.net/metaverse-gaming-is-the-talk-of-the-town-know-more/> (дата звернення 24.09.2022).

34. GTA 5 rumoured to be the most expensive video game ever made. URL: <https://www.independent.co.uk/games/gta-5-rumoured-to-be-the-most-expensive-video-game-ever-made-8809581.html> (дата звернення 25.09.2022).

35. Star Citizen has now raised more than half a billion dollars in crowdfunding. URL: <https://www.eurogamer.net/star-citizen-has-now-raised-more-than-half-a-billion-dollars-in-crowdfunding> (дата звернення 25.09.2022).

36. The Legend of Zelda: Breath of the Wild review. URL: <https://www.theverge.com/2017/3/2/14787082/the-legend-of-zelda-breath-of-the-wild-review> (дата звернення 25.09.2022).

37. God of War Review. URL: <https://www.ign.com/articles/2018/04/12/god-of-war-review> (дата звернення 25.09.2022).

38. 10 Best Naughty Dog Games, According To IMDb. URL: <https://screenrant.com/naughty-dog-best-games-imdb/> (дата звернення 25.09.2022).

39. 101 Amazing Little Details in Red Dead Redemption 2. URL: <https://www.ign.com/articles/2018/11/12/79-amazing-little-details-in-red-dead-redemption-2> (дата звернення 25.09.2022).

40. Fortnite chapter 2. URL: <https://www.epicgames.com/fortnite/en-US/chapter2> (дата звернення 25.09.2022).

41. Unreal Engine 5. URL: <https://www.unrealengine.com/en-US/unreal-engine-5> (дата звернення 25.09.2022).

42. Griffiths, M. D. (2018). The therapeutic and health benefits of playing video games.
43. Meier, S. (2020). *Sid Meier's Memoir!: A Life in Computer Games*. WW Norton & Company.
44. Daniel, M., & Garry, C. (2018). *Video games as culture: considering the role and importance of video games in contemporary society*. Routledge.
45. Schreier, J. (2017). *Blood, sweat, and pixels: The triumphant, turbulent stories behind how video games are made*. New York: Harper.
46. Schreier, J. (2021). *Press reset: ruin and recovery in the video game industry*. Grand Central Publishing.
47. Schell, J. (2008). *The Art of Game Design: A book of lenses*. CRC press.
48. Mechner, J., & Novgorodoff, D. (2011). *The Making of Prince of Persia: Journals, 1985-1993* (p. 330). ICG Testing.
49. Tanskanen, S. (2018). Player immersion in video game: Designing an immersive game project.
50. Michailidis, L., Balaguer-Ballester, E., & He, X. (2018). Flow and immersion in video games: The aftermath of a conceptual challenge. *Frontiers in psychology*, 9, 1682.
51. Klimmt, C., Possler, D., May, N., Auge, H., Wanjek, L., & Wolf, A. L. (2019). Effects of soundtrack music on the video game experience. *Media Psychology*, 22(5), 689-713.
52. Fields, T. (2022). *Game Development 2042: The Future of Game Design, Development, and Publishing*. CRC Press.
53. Khalifa, A., de Mesentier Silva, F., & Togelius, J. (2019, August). Level design patterns in 2D games. In 2019 *IEEE Conference on Games (CoG)* (pp. 1-8). IEEE.
54. What Are 2.5D Games? How They Differ From 2D and 3D Games. URL: <https://www.makeuseof.com/what-are-2-5d-games-2d-3d/> (дата звернення 10.10.2022).

55. Akenine-Mo, T., Haines, E., & Hoffman, N. (2018). Real-time rendering.
56. Fake vs True Displacement - Part 1/2. URL: <https://www.artstation.com/blogs/andreasmischok/jzWz/fake-vs-true-displacement-part-12> (дата звернення 12.10.2022).
57. Fake vs True Displacement - Part 2/2. URL: <https://www.artstation.com/blogs/andreasmischok/WYVG/fake-vs-true-displacement-part-22> (дата звернення 12.10.2022).
58. The Polygonal Evolution of 5 Iconic PlayStation Characters. URL: <https://blog.playstation.com/2019/12/16/the-polygonal-evolution-of-5-iconic-playstation-characters/> (дата звернення 12.10.2022).
59. Top 10: Best 3D Modeling Software of 2022. URL: <https://all3dp.com/1/best-free-3d-modeling-software-3d-cad-3d-design-software/> (дата звернення 12.10.2022).
60. Курило, І. Є. (2019). Методи 3d сканування для задач ігрової індустрії.
61. Чан, А. Л. В., & Романюк, О. Н. (2020). *Аналіз пристроїв 3D-сканування* (Doctoral dissertation, ВНТУ).
62. Nässén, O., & Leiborn, E. (2019). Real-time Terrain Deformation with Isosurface Algorithms.
63. Hanson, C. (2018). *Game time: Understanding temporality in video games*. Indiana University Press.
64. Abbott, E. A. (2018). *Flatland: A Romance of Many Dimensions (Illustrated)*. Youcanprint.
65. Cavallo, M. (2021, May). Higher Dimensional Graphics: Conceiving Worlds in Four Spatial Dimensions and Beyond. In *Computer Graphics Forum* (Vol. 40, No. 2, pp. 51-63).
66. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Intelligent classification of biophysical system states using fuzzy interval logic, *Telecommunications and Radio Engineering*, 78(14), pp. 1303-1315.

67. Tvoroshenko Irina, Ahmad M. Ayaz, Mustafa Syed Khalid, Lyashenko Vyacheslav, and Alharbi Adel R. (2020) Modification of Models Intensive Development Ontologies by Fuzzy Logic, *International Journal of Emerging Trends in Engineering Research*, 8(3), pp. 939-944.

68. Кучеренко, Е.И., Филатов, В.А., Творошенко, И.С., & Байдан, Р.Н. (2005). Интеллектуальные технологии в задачах принятия решений технологических комплексов на основе нечеткой интервальной логики. *Восточно-Европейский журнал передовых технологий*, (2), 92-96.

69. Lyashenko V., Mustafa S.K., Tvoroshenko I., and Ahmad M.A. (2020) Methods of Using Fuzzy Interval Logic During Processing of Space States of Complex Biophysical Objects, *International Journal of Emerging Trends in Engineering Research*, 8(2), pp. 372-377.

70. Творошенко, І. С., Мгеброва, В. Р., & Білий, В. В. (2016). Практичні аспекти створення вихідної інформації для проведення геоінформаційного аналізу у сфері управління нерухомістю.

71. Daradkeh Y.I., and Tvoroshenko I. (2020) Technologies for Making Reliable Decisions on a Variety of Effective Factors using Fuzzy Logic, *International Journal of Advanced Computer Science and Applications*, 11(5), pp. 43-50.

72. Tvoroshenko I.S., and Gorokhovatsky V.O. (2020) Effective tuning of membership function parameters in fuzzy systems based on multi-valued interval logic, *Telecommunications and Radio Engineering*, 79(2), pp. 149-163.

73. Творошенко, І. С., Шевченко, А. Р. (2018) Удосконалення просторової мережі навчальних закладів міста Сєвєродонецька на основі геоінформаційного аналізу. *Системи обробки інформації*, (1), 46-52.

74. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

75. Tvoroshenko I.S., and Kramarenko O.O. (2019) Software determination of the optimal route by geoinformation technologies, *Radio Electronics Computer Science Control*, 3, pp. 131-142.

76. Asaad Ma. Babker, Abd Elgadir A. Altoum, Irina Tvoroshenko, and Vyacheslav Lyashenko (2019) Information Technologies of the Processing of the Spaces of the States of a Complex Biophysical Object in the Intellectual Medical System HEALTH, *International Journal of Advanced Trends in Computer Science and Engineering*, 8(6), pp. 3221-3227.

77. Gorokhovatskyi, V. A., Rusakova, N., & Tvoroshenko, I. S. (2020). The application of image analysis methods and predicate logic in applied problems of magnetic monitoring. *Telecommunications and Radio Engineering*, 79(20).

78. Творошенко, І. С., & Табашник, В. А. (2018). Розробка просторової моделі геоінформаційної підтримки людей з обмеженими можливостями, що пересуваються на інвалідних колясках, у місті Харків. *Збірник наукових праць Харківського національного університету Повітряних Сил*, (1), 122-128.

79. Tvoroshenko, I., & Tkachenko, D. (2020). Mechanisms of image classification based on descriptors of local features.

80. Tvoroshenko, I., & Babochkin, O. (2021). Object identification method based on image keypoint descriptors.

81. Кучеренко, Є. І., Творошенко, І. С., Анопрієнко, Т. В. (2016) Моделювання та оцінювання станів складних об'єктів із застосуванням формальної логіки. *Системи обробки інформації*, (2), 76-82.

82. Творошенко, І. С. (2010). Анализ процессов принятия решений в интеллектуальных системах. *Системи обробки інформації*, (2), 248-253.

83. Творошенко, І. С. (2018). Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем. *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland*, 118-121.

84. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems, *Telecommunications and Radio Engineering*, 78(20), pp. 1857-1868.

85. Tvoroshenko, I., & Kukharchuk, V. (2021). Current state of development of applications for recognition of faces in the image and frames of video captures.
86. Кучеренко, Е. И., Творошенко, И. С. (2010) Прикладные аспекты моделирования нечетких процессов в сложных системах. *Збірник наукових праць Харківського університету Повітряних сил*, (1), 127-131.
87. Tvoroshenko, I., & Dziubenko, M. (2020). Modern methods of analysis of the movement scheme using video detection of vehicles.
88. Tvoroshenko, I. (2019). Development of models of spatial analysis of status of interactive processes of complex systems.
89. Tvoroshenko I., and Zarivchatskyi R. (2020) Analysis of existing methods for searching object in the video stream, Abstracts of VI International Scientific and Practical Conference «About the problems of science and practice, tasks and ways to solve them» (October 26-30, 2020). Milan, Italy, pp. 500-505.
90. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.
91. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.
92. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ. 124 с.
93. Ahmad M.A., Gorokhovatskyi V., Tvoroshenko I., Vlasenko N., and Mustafa S.K. (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Parameters for the Structural Description, *International Journal of Engineering Trends and Technology*, 69(10), pp. 186-192.
94. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. *Сучасні інформаційні системи*, 6 (3), С. 5–12.

95. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання. Міжнародний науковий симпозіум «Інтелектуальні рішення-С». Обчислювальний інтелект (результати, проблеми, перспективи). Теорія прийняття рішень: праці міжн. наук. симпозіуму (Вересень 29, 2021). Київ – Ужгород, С. 44-45.

96. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73 (3), pp. 6069–6084.

97. Smoke, mirrors, and scrolling textures: Behind the scenes of TUNIC. URL: <https://blog.unity.com/games/smoke-mirrors-and-scrolling-textures-behind-the-scenes-of-tunic> (дата звернення 03.11.2022).

98. Tunic Hides Its Secrets in Plain Sight. URL: <https://www.superjumpmagazine.com/tunic-hides-its-secrets-in-plain-sight/> (дата звернення 03.11.2022).

99. Serpa, Y. R., & Rodrigues, M. A. F. (2019, October). Towards machine-learning assisted asset generation for games: a study on pixel art sprite sheets. In *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (pp. 182-191). IEEE.

100. Depth-Aware video frame INterpolation (DAIN). URL: <https://github.com/baowenbo/DAIN> (дата звернення 05.11.2022).

101. Van der Spuy, R. (2010). *AdvancED Game Design with Flash*. Apress.

102. Arshad, M. R., Yoon, K. H., Manaf, A. A. A., & Ghazali, M. A. M. (2019). Physical rigging procedures based on character type and design in 3D animation. *International Journal of Recent Technology and Engineering*, 8(3), 4138-47.

103. Wang, H., Güler, R. A., Kokkinos, I., Papandreou, G., & Zafeiriou, S. (2020, August). Blsm: A bone-level skinned model of the human mesh. In *European Conference on Computer Vision* (pp. 1-17). Springer, Cham.

104. Sharma, S., Verma, S., Kumar, M., & Sharma, L. (2019, February). Use of motion capture in 3D animation: motion capture systems, challenges, and recent trends. In *2019 international conference on machine learning, big data, cloud and parallel computing (comitcon)* (pp. 289-294). IEEE.
105. DeepMotion: AI Driven Motion for Games. URL: <https://80.lv/articles/deepmotion-ai-driven-motion-for-games> (дата звернення 09.11.2022).
106. When Fandom Clashes with IP Law. URL: <https://hbr.org/2019/07/when-fandom-clashes-with-ip-law> (дата звернення 13.11.2022).
107. Uncharted 4 Stage Fright trophy pokes fun at a notorious E3 fail. URL: <https://www.gamesradar.com/uncharted-4-stage-fright-trophy-pokes-fun-at-a-notorious-e3-fail/> (дата звернення 13.11.2022).
108. The True Story Behind The Original Video Game 'Easter Egg' That Inspired 'Ready Player One'. URL: <https://www.forbes.com/sites/sethporges/2017/12/20/the-true-story-behind-the-original-video-game-easter-egg-that-inspired-ready-player-one/?sh=46175c852976> (дата звернення 13.11.2022).
109. Guitart, A., Chen, P. P., Bertens, P., & Periañez, Á. (2018, April). Forecasting player behavioral data and simulating in-game events. In *Future of Information and Communication Conference* (pp. 274-293). Springer, Cham.
110. Tvoroshenko, I., & Almakaieva, A. (2020). Application of procedural generation of game content using software algorithms.
111. Борисенко Є. (2022) Огляд методів розроблення ігрового світу, *Abstracts of XXXVII International Scientific and Practical Conference «Modern ways of solving the latest problems in science» (September 20 – 23, 2022). Varna, Bulgaria*, pp. 435-437.