

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет**  
**імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**  
Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_ Ю. П. Кондратенко  
«\_\_\_» \_\_\_\_\_ 2022 р.

**БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

**ФОРМУВАННЯ 3D-МОДЕЛІ ДЛЯ ІГРОВОГО  
ЗАСТОСУНКУ ЗАСОБАМИ ШТУЧНОГО ІНТЕЛЕКТУ**

Спеціальність 122 «Комп'ютерні науки»

**122 – БКР – 401.21810125**

*Виконав студент 4-го курсу, групи 401*

\_\_\_\_\_ *І. В. Чепара*

«13» червня 2022 р.

*Керівник: д-р техн. наук, проф.*

\_\_\_\_\_ *О. П. Гожий*

«13» червня 2022 р.

**Миколаїв – 2022**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет ім. Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

Рівень вищої освіти **бакалавр**  
Спеціальність **122 «Комп'ютерні науки»**  
*(шифр і назва)*  
Галузь знань **12 «Інформаційні технології»**  
*(шифр і назва)*

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук, проф.  
\_\_\_\_\_ Ю. П. Кондратенко  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**З А В Д А Н Н Я**  
**на виконання кваліфікаційної роботи**

Видано студенту групи 401 факультету комп'ютерних наук Чепарі Івану Васильовичу

1. Тема кваліфікаційної роботи «Формування 3D-моделі для ігрового застосунку засобами штучного інтелекту».

Керівник роботи Гожий Олександр Петрович, д-р техн. наук, проф.

Затв. наказом Ректора ЧНУ ім. Петра Могили від « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р. № \_\_\_\_

2. Строк представлення кваліфікаційної роботи студентом « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

3. Вхідні (початкові) дані до роботи: Формування 3D-моделі для ігрового застосунку на основі 2D зображення засобами штучного інтелекту

Очікуваний результат: Ігровий застосунок на ігровому движку Unity та програма для побудови 3D моделі засобами штучного інтелекту.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- Аналіз актуальності репрезентації гравця у ігровому всесвіті
- Визначення впливу машинного навчання на розвиток відеоігор
- Огляд існуючих методів вирішення технічної задачі;

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «формування у майбутніх фахівців відповідних знань щодо особливостей впливу несприятливих виробничих чинників на комп'ютеризованих робочих місцях»

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	О. В. Макарова	

Керівник роботи д-р техн. наук, проф. Гожий О. П.

(наук. ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Завдання прийнято до виконання Чепара І.В.

(прізвище та ініціали)

\_\_\_\_\_ (підпис)

Дата видачі завдання « 26 » \_\_\_\_\_ листопада \_\_\_\_\_ 2022 р.

**КАЛЕНДАРНИЙ ПЛАН**  
**виконання кваліфікаційної роботи**

Тема: Формування 3D-моделі для ігрового застосунку засобами штучного інтелекту

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заяви на затвердження теми та керівників БКР	26.10.2022	26.10.2022	
2	Отримання завдання на виконання БКР	22.11.2022	22.11.2022	
3	Складання календарного плану роботи на весь період виконання БКР	08.12.2022	09.12.2022	
4	Отримання завдання на переддипломну практику	21.05.2022	21.05.2022	
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	23.05.2022	04.06.2022	
6	Розробка звіту з переддипломної практики	04.06.2022	06.06.2022	
7	Розробка програмного застосунку для формування 3D-моделі засобами штучного інтелекту	06.06.2022	11.06.2022	
8	Розробка ігрового застосунку за допомогою ігрового движка	12.06.2022	16.06.2022	
9	Попередній захист БКР на засіданні комісії кафедри	30.05.2022	31.05.2022	
10	Доробка та остаточне оформлення БКР	02.06.2022	20.06.2022	
11	Подання БКР рецензенту	21.06.2022	21.06.2022	
12	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	22.06.2022	25.06.2022	
13	Захист БКР перед екзаменаційною комісією (ЕК)	27.06.2022	29.06.2022	

Розробив студент Чепара І.В. \_\_\_\_\_  
(прізвище та ініціали) (підпис)

Керівник роботи д-р техн. наук. проф. Гожий О.П. \_\_\_\_\_  
(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2022 р.

## АНОТАЦІЯ

**бакалаврської кваліфікаційної роботи студента групи 401 ЧНУ ім. Петра  
Могили  
Чепари Івана Васильовича**

**Тема: «Формування 3D-моделі для ігрового застосунку засобами штучного  
інтелекту»**

**Об'єкт роботи** – методи побудови 3D моделі засобами штучного інтелекту.

**Предмет роботи** – програма для генерації 3D моделі за допомогою штучного інтелекту на основі зовнішніх даних людини, використовуючи 2D зображення.

**Метою** бакалаврської кваліфікаційної роботи є репрезентація гравця у відеогрі за допомогою створення його аватару, який має з ним зовнішню схожість.

Робота складається з фахового розділу і спеціальної частини з охорони праці. Пояснювальна записка складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі представлені аналіз розвитку відеоігор з метою встановлення появи потреби у «зануренні» гравців та аналіз самого занурення з метою дослідження чому і як воно впливає на ігровий досвід.

У другому розділі було проаналізовано вплив машинного навчання на відеоігри, які технології машинного навчання використовуються, які задачі вони вирішують та було досліджено вплив машинного навчання на 3D моделювання, які алгоритми використовуються, проблеми 3D моделювання та машинного навчання.

У третьому розділі були проаналізовані найпопулярніші технології для розробки відеоігор та моделювання 3D. За допомогою порівняння сильних та слабких сторін кожної технології, було прийнято рішення обрати Unity як ігровий движок та Blender як 3D редактор.

У четвертому розділі було розроблено програму для формування 3D моделі по 2D зображенню засобами штучного інтелекту, проводилось корегування 3D моделі у Blender, було розроблено ігровий застосунок за допомогою Unity.

Результатом роботи був ігровий застосунок на базі ігрового движка Unity та програма для створення 3D моделі за допомогою штучного інтелекту.

Бакалаврська кваліфікаційна робота викладена на 92 сторінках, вона містить 4 розділи, 48 ілюстрацій, 0 схеми та 38 джерела у переліку посилань.

Ключові слова: відеогра, штучний інтелект, 3D моделювання, Unity, Blender, C#, занурення, нейронні мережі.

## **ABSTRACT**

**of bachelor's degree qualification work of a student of group 401 BSNU named  
after Petro Mohyla  
Chepara Ivan Vasilyevich**

**Topic: "Formation of a 3D model for gaming applications by means of artificial  
intelligence"**

The object of work - methods of building a 3D model using artificial intelligence.

The subject of the work is a program for generating a 3D model using artificial intelligence based on human external data, using 2D images.

The purpose of the bachelor's thesis is to represent the player in the video game by creating his avatar, which has an external resemblance to him.

The work consists of a professional section and a special part on labor protection. The explanatory note consists of an introduction, four chapters, conclusions and appendices.

The first section presents an analysis of the development of video games in order to establish the need for "immersion" of players and analysis of the immersion in order to explore why and how it affects the gaming experience.

The second section analyzed the impact of machine learning on video games, what machine learning technologies are used, what problems they solve and investigated the impact of machine learning on 3D modeling, which algorithms are used, problems of 3D modeling and machine learning.

The third section analyzes the most popular technologies for video game development and 3D modeling. By comparing the strengths and weaknesses of each technology, it was decided to choose Unity as the game engine and Blender as the 3D editor.

In the fourth section, a program was developed for the formation of a 3D model based on 2D images by means of artificial intelligence, the 3D model was adjusted in Blender, and a game application was developed using Unity.

The result was a game application based on the Unity game engine and a program for creating 3D models using artificial intelligence.

The bachelor's thesis is set out on 92 pages, it contains 4 sections, 48 illustrations, 0 diagrams and 38 sources in the list of references.

Keywords: video game, artificial intelligence, 3D modeling, Unity, Blender, C #, immersion, neural network

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ .....	4
ВСТУП.....	6
1 АНАЛІЗ АКТУАЛЬНОСТІ ПРОБЛЕМИ РЕПРЕЗЕНТАЦІЇ ГРАВЦЯ ЧЕРЕЗ ІГРОВУ 3D-МОДЕЛЬ У ВІДЕОІГРАХ.....	8
1.1 Аналіз потреби занурення у відеоіграх .....	8
1.2 Аналіз розвитку занурення у відеоіграх та роль ігрової моделі .....	13
1.3 Існуючі аналоги репрезентуючих моделей .....	18
1.4 Постановка задачі .....	21
Висновки до розділу 1 .....	21
2 АНАЛІЗ ВПЛИВУ ШТУЧНОГО ІНТЕЛЕКТУ НА ВІДЕОІГРИ .....	23
2.1 Роль і технології машинного навчання у відеоіграх.....	23
2.2 Вплив машинного навчання на 3D моделювання .....	34
Висновок до розділу 2 .....	38
3 ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМИ РЕПРЕЗЕНТАЦІЇ ГРАВЦЯ .....	39
3.1 Технології для створення ігрового додатку .....	39
3.2 Технології для побудови 3D-моделі .....	48
3.3 Технології машинного навчання .....	55
Висновки до розділу 3.....	58
4 СТРУКТУРА ТА РЕАЛІЗАЦІЯ ПРОГРАМИ ДЛЯ ФОРМУВАННЯ 3D МОДЕЛІ.....	59
4.1 Структура програми для формування 3D моделі .....	59
4.2 Реалізація програми для формування 3D моделі.....	64

4.3 Підготовка моделі та її імпорт в ігровий застосунок .....	69
<b>5 ОХОРОНА ПРАЦІ .....</b>	<b>78</b>
5.1 Міжнародні та державні норми з питань безпеки праці користувачів інформаційних технологій .....	79
5.2 Виробничі фактори, які можуть спричинити професійні захворювання в ІТ-індустрії, та методи й засоби їх усунення .....	81
5.3 Травми та професійні захворювання на підприємствах та робочих місцях галузі .....	84
Висновок до розділу 5 .....	89
<b>ВИСНОВКИ .....</b>	<b>90</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>91</b>



## ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

NPC (Non-Playable Characters) – це персонажі гри, відмінні від основного гравця.

AI (Artificial Intelligence) – властивість інтелектуальних систем виконувати творчі функції, які традиційно вважаються прерогативою людини

Геймер (Gamer) – користувач ігрового застосунку

Занурення (Immersion) – залученість та зацікавленість гравця у створеному всесвіті

Машинне навчання (ML) — це тип штучного інтелекту (AI), який дозволяє програмним застосункам стати більш точними в прогнозуванні результатів, не будучи явно запрограмованим на це.

Згортка нейронна мережа (CNN) — це нейронна мережа, яка має один або кілька згорткових шарів і використовується в основному для обробки зображень, класифікації, сегментації, а також для інших автоматично корельованих даних.

# Пояснювальна записка

до кваліфікаційної роботи

на тему:

**«Формування 3D-моделі для ігрового застосунку  
засобами штучного інтелекту»**

Спеціальність 122 «Комп'ютерні науки»

**122 – БКР – 401.21810125**

*Виконав студент 4-го курсу, групи 401*

***І.В. Чепара***

*(підпис, ініціали та прізвище)*

«13» червня 2022 р.

*Керівник: д-р техн. наук, проф.*

*(наук. ступінь, вчене звання)*

***О.П. Гожий***

*(підпис, ініціали та прізвище)*

«13» червня 2022 р.

**Миколаїв – 2022**

## ВСТУП

Ігровий застосунок (англ. Game Application, англ. Game App) або відеогра (англ. Video Game) – це програма, яка реалізує деякий інтерактивний ігровий процес або геймплей (англ. Gameplay), який призначений для того, щоб зацікавити або розважити користувача, який іменується гравцем (англ. Gamer). Це визначення загалом описує як і перші створені ігрові застосунки 50-60х років 20 століття, розробка яких велась зусиллями однієї людини або команди з декількох осіб, так і сучасні великобюджетні Triple-A (AAA) проекти, команди розробників яких може налічувати тисячі людей з різноманітніших сфер ІТ.

Але у сучасному світі гравці потребують не тільки ігровий процес, а також деякі елементи, які стали невід'ємною частиною, у тому чи іншому вигляді, практично всіх сучасних ігрових застосунків.

До основних елементів сучасної відеогри відносять:

- сетинг, що представляє собою сукупність складових вигаданого художнього світу, які однозначно визначають унікальність створеного середовища, де відбуваються основні події ігрового застосунку.
- геймплей є основною частиною ігрового застосунку, так як усі супутні елементи мають важливе значення, але саме геймплей становить суть відеогри. Він визначає чим саме буде займатись гравець протягом всієї ігрової сесії. Геймплей складається з механік, які дозволяють виконувати певні дії та ігрові умовності, які обмежують гравця. Усі інші елементи базуються над інтерактивним ігровим процесом да доповнюють його.
- сюжет, який є невід'ємною складовою будь-якої гри у тому чи іншому виді та у тій чи іншій кількості, в залежності від структури самої відеогри та погляду розробників на розвиток цієї гри, являє собою послідовність подій, які взаємозв'язані у обраному художньому всесвіті та покладені в основу замислу даної гри, які описують передісторію виміру та мотивацію головного або головних героїв та їх існуючих та майбутніх вчинків.

Усі перелічені елементи ігрового застосунку, окрім базового ігрового процесу, еволюціонували з потреб гравців у більш осмислених проектах, які окрім механічного процесу, можуть надати більш комплексне проведення часу за допомогою додаткових елементів самого проекту. З цієї причини, ігрові застосунки перетворилися на повноцінні витвори мистецтва, про що свідчить додання відеоігор до переліку видів мистецтв урядом Сполучених Штатів Америки у 2011 році.

Так як сучасні відеоігри вважаються одним із видів мистецтв, вони повинні відповідати головним характеристикам даного напрямлення. Актуальність обраної теми проявляється у потребі більш занурити гравців у створений художній всесвіт, а саме у репрезентації гравця у відеоігри методом реалізації його аватару, який буде представляти його особистість та мати схожий зовнішній вигляд. Ця потреба полягає в тому, щоб гравець відчув себе частиною створеного світу, він сам, або його аватар, який його представляє, був безпосереднім учасником подій, що відбуваються.

**Об'єктом розробки** є методи побудови 3D моделі засобами штучного інтелекту.

**Предметом роботи** є програма для генерації 3D моделі за допомогою штучного інтелекту на основі зовнішніх даних людини, використовуючи 2D зображення.

**Мета** комплексної дипломної роботи полягає у репрезентації гравця у відеоігри за допомогою створення його аватару, який має з ним зовнішню схожість.

# 1 АНАЛІЗ АКТУАЛЬНОСТІ ПРОБЛЕМИ РЕПРЕЗЕНТАЦІЇ ГРАВЦЯ ЧЕРЕЗ ІГРОВУ 3D-МОДЕЛЬ У ВІДЕОІГРАХ

## 1.1 Аналіз потреби занурення у відеоіграх

Для аналізу потреби занурення у відеоіграх потрібно дослідити як вони розвивались с впливом часу, щоб зрозуміти, чому в іграх не вистачало тільки ігрового процесу для розваження гравця, які додаткові елементи були створені для вирішення цієї проблеми і коли з'явилася потреба у зануренні у відеоіграх.

У 1952 році один британський професор А.С. Дуглас створив один з перших ігрових застосунків «ОХО», в основному відома як хрестики-нулики, як частину своєї докторської дисертації в Кембриджському університеті. А в 1958 році Вільям Хігінботем створив «Теніс для двох» на великому аналоговому комп'ютері та підключив екран осцилографа до щорічного Дня відвідувача Брукхейвенської національної лабораторії в Аптоні, штат Нью-Йорк. [1]

У 1962 році Стів Рассел з Массачусетського технологічного інституту винайшов «Spacewar!» (див. Рисунок 1.1), яка зараз вважається однією з перших масових комп'ютерних відеоігор для космічного бою для PDP-1, який тоді використовувався в основному в університетах. Це була перша відеогра, в яку можна було грати на кількох комп'ютерах.



Рисунок 1.1 – Інтерфейс «Spacewar!»

У 1967 році розробники компанії Sanders Associates Inc. на чолі з Ральфом Баєром винайшли прототип багатокористувацької системи відеоігор, у яку можна було б грати на телевізорі. Його називали «коричневою коробкою». Байєр, якого іноді називають батьком відеоігор, ліцензував свій пристрій компанії Magnavox, яка продала систему споживачам у 1972 році як Odyssey, першу домашню ігрову консоль. Протягом наступних кількох років примітивна консоль Odyssey стала комерційно «помирати», тобто завдавати збитків. Однак одна з 28 ігор Odyssey надихнула Atari Pong, першу аркадну відеогру компанії, випущену в 1972 році. У 1975 році Atari випустила домашню версію Pong, яка була настільки ж успішною, як і його аркадний аналог. Magnavox і Sanders Associates зрештою подадуть до суду на Atari за порушення авторських прав. Atari поступився і став ліцензіатом Odyssey. Протягом наступних 20 років Magnavox виграла понад 100 мільйонів доларів у судах щодо авторських прав на Odyssey та його патенти на відеоігри. У 1977 році Atari випустила Atari 2600 (також відому як Video Computer System), домашню консоль з джойстиком та змінними ігровими картриджами, які могли грати в різнокольорові ігри, фактично запустивши друге покоління відеоігрових консолей.

[2]

Наприкінці 1970-х і початку 1980-х років індустрія відеоігор мала кілька помітних етапів, зокрема:

- 1) аркадна гра Space Invaders була випущена в 1978 році
- 2) Activision була заснована в 1979 році як перший сторонній розробник ігор (який розробляє програмне забезпечення без створення консолей або аркадних кабінетів).
- 3) сталося знайомство з надзвичайно популярним Pac-Man у Сполучених Штатах Америки
- 4) створення Donkey Kong компанією Nintendo, яка представила світу персонажа Mario
- 5) Microsoft випустила свою першу гру Flight Simulator

У 1983 році індустрія відеоігор у Північній Америці пережила серйозний «крах» через низку факторів, включаючи перенасичений ринок ігрових консолей, конкуренцію з боку комп'ютерних ігор і надлишок розкручених і низькоякісних ігор, таких як сумнозвісна E.T., гра Atari, заснована на однойменному фільмі і часто вважається найгіршою грою, яку коли-небудь створювали.

Криза, що тривала кілька років, призвела до банкрутства кількох компаній, що випускають домашні комп'ютери та ігрові приставки. Домашня індустрія відеоігор почала відновлюватися в 1985 році, коли Nintendo Entertainment System (NES), яка називається Famicom в Японії, прийшла до Сполучених Штатів. NES покращила 8-бітну графіку, кольори, звук та ігровий процес порівняно з попередніми консолями.

Nintendo, японська компанія, яка починала як виробник гральних карт у 1889 році, випустила ряд важливих франшиз відеоігор, які існують і сьогодні, таких як Super Mario Bros., The Legend of Zelda (див. Рисунок 1.2) і Metroid. Крім того, Nintendo ввела різні правила щодо сторонніх ігор, розроблених для її системи, допомагаючи боротися з поспіхом, низькоякісним програмним забезпеченням. Сторонні розробники випустили багато інших довготривалих франшиз, таких як Capcom's Mega Man, Konami's Castlevania, Square's Final Fantasy і Enix's Dragon Quest (Square і Enix пізніше об'єднуються, щоб утворити Square Enix в 2003 році).



Рисунок 1.2 – Інтерфейс «The Legend of Zelda»

У 1989 році Nintendo знову зробила хвилю, популяризуючи портативні ігри, випустивши свій 8-розрядний пристрій для відеоігор Game Boy і гру Tetris, яка часто входить у комплект. Протягом наступних 25 років Nintendo випустить ряд успішних наступників Game Boy, включаючи кольоровий Game Boy у 1998 році, Nintendo DS у 2004 році та Nintendo 3DS у 2011 році. Також у 1989 році Sega випустила свою 16-розрядну консоль Genesis у Північній Америці як наступник своєї Sega Master System 1986 року, яка не змогла адекватно конкурувати з NES. Завдяки своїй технологічній перевазі над NES, розумному маркетингу та випуску в 1991 році гри Sonic the Hedgehog, Genesis досяг значних успіхів у порівнянні зі своїм старшим суперником. У 1991 році Nintendo випустила свою 16-розрядну консоль Super NES в Північній Америці, розпочавши першу справжню «консольну війну». На початку-середині 1990-х було випущено безліч популярних ігор на обох консолях, включаючи нові франшизи, такі як Street Fighter II і Mortal Kombat, файтинг, який зображував кров і кров у версії гри Genesis. У відповідь на насильницькі ігри (а також на слухання Конгресу про насильницькі відеоігри) Sega створила раду з оцінки відеоігор у 1993 році, щоб надати описове маркування для кожної гри, що продається на домашній консолі Sega. Пізніше було створено комітет з оцінок програмного забезпечення для розваг, який досі використовується для оцінювання відеоігор на основі вмісту. У середині 1990-х років відеоігри вийшли на великий екран із виходом фільму Super Mario Bros. у 1993 році, а потім — Street Fighter і Mortal Kombat протягом наступних двох років. З тих пір було випущено безліч фільмів, заснованих на відеоіграх. Маючи набагато більшу бібліотеку ігор, нижчу ціну та успішний маркетинг, Genesis до цього часу випередив SNES у Північній Америці. Але Sega не змогла досягти подібного успіху в Японії.

Завдяки стрибку в комп'ютерних технологіях, п'яте покоління відеоігор започаткувало тривимірну еру ігор. У 1995 році Sega випустила в Північній Америці свою систему Saturn, першу 32-розрядну консоль, яка грала в ігри на компакт-дисках, а не на картриджах, на п'ять місяців раніше запланованого. Цей



крок мав перевершити перший набіг Sony у відеоігри, Playstation, які продавалися на 100 доларів менше, ніж Saturn, коли він був запущений пізніше того ж року. Наступного року Nintendo випустила свою 64-розрядну систему на основі картриджів Nintendo 64. Хоча Sega і Nintendo кожна випустила свою частку високорейтингових 3D-ігор від бренду, таких як Virtua Fighter на Saturn і Super Mario 64 на Nintendo 64, відомі компанії з відеоігор не могли конкурувати з сильними третими компаніями Sony. партійна підтримка, яка допомогла Playstation отримати численні ексклюзивні ігри. Простіше кажучи: Sony домінувала на ринку відеоігор і продовжуватиме це робити в наступному поколінні. Фактично, Playstation 2, випущена в 2000 році і здатна грати в оригінальні ігри Playstation, стала б найбільш продаваною ігровою консоллю всіх часів. Playstation 2, яка була першою консоллю, яка використовувала DVD, зіткнулася з Sega Dreamcast (випущена в 1999 році), Nintendo Gamecube (2001) і Xbox від Microsoft (2001). Dreamcast, який, на думку багатьох, випередив свій час і одна з найкращих консолей, коли-небудь створених з кількох причин, включаючи можливість онлайн-ігор, стала комерційним провалом, який поклав край консольним зусиллям Sega. Sega вимкнула систему в 2001 році, ставши відтепер сторонньою компанією, що займається програмним забезпеченням.

У 2005 і 2006 роках Xbox 360 від Microsoft, Playstation 3 від Sony і Wii від Nintendo започаткували сучасну епоху ігор високої чіткості. Хоча Playstation 3 — єдина на той час система для відтворення Blu-ray — сама по собі була успішною, Sony вперше зіткнулася з жорсткою конкуренцією з боку своїх конкурентів. Xbox 360, який мав подібні графічні можливості до Playstation 3, отримав похвалу за свою екосистему онлайн-ігор і в 2007 році отримав набагато більше нагород критиків гри, ніж інші платформи; у ньому також була представлена Microsoft Kinect, найсучасніша система захоплення руху, яка пропонувала інший спосіб грати у відеоігри (хоча Kinect ніколи не був популярним серед основних гравців або розробників ігор). І незважаючи на те, що технологічно поступається двом іншим системам, Wii перевершила свою конкуренцію в продажах. Його чутливі до

руху пульта дистанційного керування зробили ігри активнішими, ніж будь-коли раніше, допомагаючи привернути увагу значно більшої частини широкої громадськості, включаючи людей у будинках престарілих. Ближче до кінця десятиліття і початку наступного відеоігри поширилися на платформи соціальних мереж, як-от Facebook, і мобільні пристрої, такі як iPhone, охоплюючи більш звичайну ігрову аудиторію. Як повідомляється, Rovio, компанія, яка створила гру для мобільних пристроїв Angry Birds (а пізніше й мультфільм Angry Birds), заробила колосальні 200 мільйонів доларів у 2012 році. У 2011 році Skylanders: Spyro's Adventure представила відеоігри у фізичний світ. [3] Гра вимагала від гравців розміщувати пластикові іграшкові фігурки (продаються окремо) на аксесуар, який зчитує NFC-мітки іграшок, щоб залучити персонажів у гру. У наступні кілька років ми побачимо кілька сиквелів та інших гібридів іграшкових і відеоігор, наприклад Disney Infinity, в якому зображені персонажі Disney. Восьме й нинішнє покоління відеоігор почалося з випуску Nintendo Wii U у 2012 році, а потім Playstation 4 та Xbox One у 2013 році. Незважаючи на наявність пульта дистанційного керування з сенсорним екраном, який дозволяв грати в ігри поза телевізором та мати можливість грати в ігри Wii, Wii U був комерційним провалом — протилежністю його конкурентам — і був припинений у 2017 році. У 2016 році Sony випустила більш потужну версію своєї консолі під назвою Playstation 4 Pro, першу консоль, здатну виводити відео 4K. На початку 2017 року Nintendo випустила свого наступника Wii U, Nintendo Switch, єдину систему, яка дозволяє як телевізійні, так і портативні ігри. Microsoft випустить свою консоль Xbox One X з підтримкою 4K наприкінці 2017 року. Завдяки своїм новим оновленим консолям і Sony, і Microsoft наразі націлені на ігри у віртуальній реальності, технологію, яка може змінити те, як гравці відчують відеоігри.

## **1.2 Аналіз розвитку занурення у відеоіграх та роль ігрової моделі**

Занурення є однією з багатьох бажаних якостей у відеоіграх. Створення захоплюючої відеоігри вимагає свідомих зусиль від розробників ігор. Занурення

залучає гравця, залишаючи його присвяченим грі, а пізніше мотивує повернутися. Немає простого трюку для розробки та створення захоплюючої відеоігри. Хоча існує кілька теорій, навіть визначення занурення у відеоігри не є загальноприйнятим. Аналіз занурення у відеоіграх спрямовано на вивчення того, як занурення впливає на ігровий процес і як цього можна досягти. Дослідження які відображені у цьому розділі нададуть корисну інформацію про занурення гравця у відеоігри. [4]

Наведена нижче діаграма (див. Рисунок 1.3) окреслює масштаб дослідження, показуючи зв'язки між темами. Обмеження цих областей дослідження для більш суворого зосередження на їх зв'язку із зануренням гравця призводить до трьох тем; емоції, інтеграція та потік. Психологія, сторітелінг, ігровий дизайн, потік, емоції та інтеграція створюють теоретичну структуру для вивчення занурення гравця у відеоігри. Ця теоретична база використовується завдяки кореляції з сферою досліджень, що реалізуються в цьому розділі.



Рисунок 1.3 – Діаграма взаємодії елементів занурення

Щоб зрозуміти основи занурення у відеоігри, важливо спочатку вивчити значення самого слова. Занурення не є новою сферою досліджень і не є унікальним для відеоігор. Імерсії досліджували в інших областях, таких як література,

кінематограф і навіть журналістика, ще до того, як з'явилися відеоігри. Незважаючи на те, що ці засоби масової інформації мають багато спільних аспектів з відеоіграми, все ще необхідно вивчати занурення щодо ігор. Це не означає, що теорії про занурення щодо інших медіа були б марними в лудології. [5]

Слово занурення походить з латині кінця 15 століття, що походить від слова *immersio* і перед цим *immergere*, що означає занурюватися. В даний час це слово розширилося і має різні значення від опису небесної події до вираження методу вивчення іноземної мови. У контексті цього дослідження мета слова занурення полягає в тому, щоб зобразити залучення. Словник Oxford University Press визначає занурення як «глибоке психічне залучення до чогось», тоді як словник Cambridge University Press визначає занурення як «повне залучення до чогось». Хоча словники здатні так детально визначити слово занурення, у лудології це не так просто.

Людологи мають різні визначення занурення у відеоігри, але жодне з них не є універсальним. Однак значення занурення у відеоігри зазвичай розуміють гравці та розробники ігор, навіть якщо немає академічного визначення. Деякі ігрові форуми містять теми, де гравці обговорюють, що саме вони вважали зануренням у певних відеоіграх. На форумі GameSpot один користувач так описує досвід занурення: «Ігри, які змусили мене забути, що я граю у відеоігри» – «Я стаю частиною іншого світу, який важко було покинути». Цей опис підкреслює глибоке психічне залучення гравця. Подібна участь у грі є лише одним із багатьох прикладів, які використовуються для опису занурення гравця.

Деякі дослідження включають такі інтерпретації: пом'якшення розумового поділу між гравцем і аватаром, відчуття участі, просторової присутності, глибоке залучення та метафора того, що гравець «входить» у дію. Ці кілька прикладів демонструють різноманітні точки зору щодо занурення у відеоігри.

Визначаючи занурення, важливо визнати, що воно побудоване з кількох аспектів. Занурення – це досвід, поєднання різних елементів, а не окрема подія. Визначення не слід залишати занадто вузьким, але важливо надати йому достатньо деталей, щоб уникнути непорозумінь. Визначення, використане в цьому аналізі,

розглядає численні виміри занурення. Занурення визначається як «інтенсифікація інтерналізованої участі, яка поєднує ряд вимірів». [6]

Інтерактивність є ключовим елементом, який відрізняє відеоігри від традиційних медіа. Занурення може відбуватися в книгах або фільмах без будь-якої взаємодії між витвором та споживачем. Інтерактивність дає відеоіграм унікальну можливість залучити гравця та запропонувати їм основу для отримання сильних переживань та емоцій. Занурення — це атрибут, який виникає внаслідок взаємодії між гравцем і грою. Інтерактивність може бути пов'язана з участю гравця багатьма способами. У відеоіграх є два види інтерактивності: взаємодія між гравцем та ігровою системою та взаємодія між гравцями. Обидва є дуже цінними методами для розвитку занурення. Взаємодія між гравцями не є необхідною для створення захоплюючого досвіду, але, безумовно, може сприяти цьому. Майже всі приклади, представлені в цьому розділі, створені на основі взаємодії та вибору, який зробили гравці.

Ігрові персонажі можуть мати значний вплив на сюжет гри. У іграх із персонажем, яким керує гравець, роль персонажа у залученні гравця може бути надзвичайно цінною. Чим більше споріднений герой, тим сильніші емоції, які може викликати історія. Приклад налаштування персонажа можна побачити на рисунку знизу (див. Рисунок 1.4). Деталі, які гравець налаштовує, можуть не бути схожими з рисами його самого, але він все одно може себе відчувати спорідненим з персонажем, так як налаштував його риси власноруч. Для цього в багатьох ігрових застосунках з можливістю створити власного персонажа присутній редактор персонажа – це набір інструментів та шаблонів для тонкого налаштування зовнішнього вигляду персонажу. Зокрема в деяких відеоіграх можливо налаштовувати голос, поведінку та інші знакові риси характеру персонажу. Такий підхід реалізований, як правило, в іграх з уклоном на рольову модель, коли гравцю надається можливість відігравати певну роль у художньому світі відеоігри.



Рисунок 1.4 – Приклад створення аватару персонажа

Налаштування персонажа може запропонувати гравцеві сильні відчуття ідентифікації. У багатьох сучасних рольових відеоіграх гравець може налаштувати персонажа на свій смак. Таких персонажів часто називають аватарами. Налаштування дозволяє гравцеві більше ідентифікувати себе з персонажем, посилюючи його мотивацію до гри. Навпаки, стверджується, що гравець стає більш емоційно залученим у гру, коли грає за попередньо створеного персонажа. Ці відмінності можна пояснити мотивація різних гравців. Гравцю, який віддає перевагу рольовим іграм, може сподобатися грати більше як аватар, ніж попередньо створений персонаж. Тоді як гравець, якому подобаються елементи розповіді в грі, може відчувати глибшу емоційну залученість, коли грає за попередньо створеного персонажа. Персонажі можуть мати дивно потужний вплив на занурення гравця. Протиріччя між мотивацією гравця та мотивацією ігрового персонажа може створити проблеми із залученням гравця. Гравці повинні мати можливість синхронізувати свої здібності вирішувати проблеми з можливостями

ігрового персонажа. Людонаративний дисонанс — це суперечність або розрив між гравцем, дизайном гри та/або елементами розповіді. Цих ситуацій слід уникати, оскільки в гіршому випадку гравець може виявити дисонанс переважаючим і повністю перестати піклуватися про персонажа, розповідь або ігровий процес і припинити грати. Отже, гравець повинен дбати про персонажа, за якого грає, і відчувати, що хоче досягти тієї ж мети за допомогою аватара. Коли ігровий процес і мотивація гравця і персонажа синхронізовані, гравець може відчувати так звану гармонію лудонаративності.

### **1.3 Існуючі аналоги репрезентуючих моделей**

Репрезентуючі моделі у відеоіграх, як правило, створюються за допомогою редактору персонажів, які надають змогу налаштувати більшість зовнішніх рис персонажу на свій погляд. Наприклад, розмір, положення та колір очей, довжину та вид волосся, колір шкіри, тип носу, вух та губ, макіяж, татуїровки та інші. Відносно нещодавно до деяких редакторів персонажів додалися елементи штучного інтелекту, наприклад, побудова лиця персонажу по фотографії користувача.

Деякі ігри дають заздалегідь визначений характер з ретельно розробленим зовнішнім виглядом. Інші ігри використовують інший підхід і дозволяють налаштувати зовнішній вигляд свого персонажа, використовуючи повзунки для регулювання рис обличчя та форми тіла, дозволяючи поєднувати попередні налаштування та додавати аксесуари. Розглянемо приклади редакторів персонажів. [7]

#### **Серія ігор «The Sims»**

Коли ви створюєте Сіма (персонажа гри «The Sims»), ваш контроль над персонажем майже абсолютний. Ви можете зробити обличчя будь-яким способом, ви можете змінити форму тіла та вибрати бажаний одяг для різних випадків. ви також можете визначити особистість і амбіції Сіма, створивши персонажа з

внутрішніми мотивами (див. Рисунок 1.5). Гравцям дуже цікаво створювати Сімів за допомогою великого асортименту інструментів для персоналізації персонажа. [8]



Рисунок 1.5 – Редактор персонажу «The Sims»

### **Black Desert Online**

Редактор персонажів Black Desert Online є одним з найвражаючих на ринку, особливо якщо врахувати тонкий контроль над деталями. Лише кілька ігор дозволяють користувачеві коригувати такі речі, як стрижка. Зазвичай, як тільки ви вибираєте волосся, воно статично, і ви не можете нічого з цим зробити, окрім як вибрати інше або змінити колір. Але Black Desert дозволяє переміщати волосся, змінювати їх довжину та наносити мелірування. Ця свобода поширюється і на інші риси вашого майбутнього характеру: пропорції кінцівок, форму обличчя, незначні зміни очей чи носа. Така увага до деталей створеного персонажу позитивно впливає на відчуття спорідненості з персонажем, так як якщо ви можете налаштувати так точно свого аватару, то він буде краще репрезентувати вас у створеному виміру гри, чим аватар, який створений через менш гнучкий редактор персонажу. (див. Рисунок 1.6)





Рисунок 1.6 – Редактор персонажу «Black Desert»

### **Dragon Age: Inquisition**

Для того, щоб персонаж репрезентував гравця у всесвіті гри, необов'язково, щоб він відносився до людської раси [9]. Наприклад, у Dragon Age: Inquisition на вибір є чотири види створінь (люди, ельфи, гноми, кунарі), кожен із яких має кілька попередніх налаштувань голови, які можна додатково налаштувати від підборіддя до волосся за допомогою повзунків і точного поля налаштування зовнішності. (див. Рисунок 1.7)

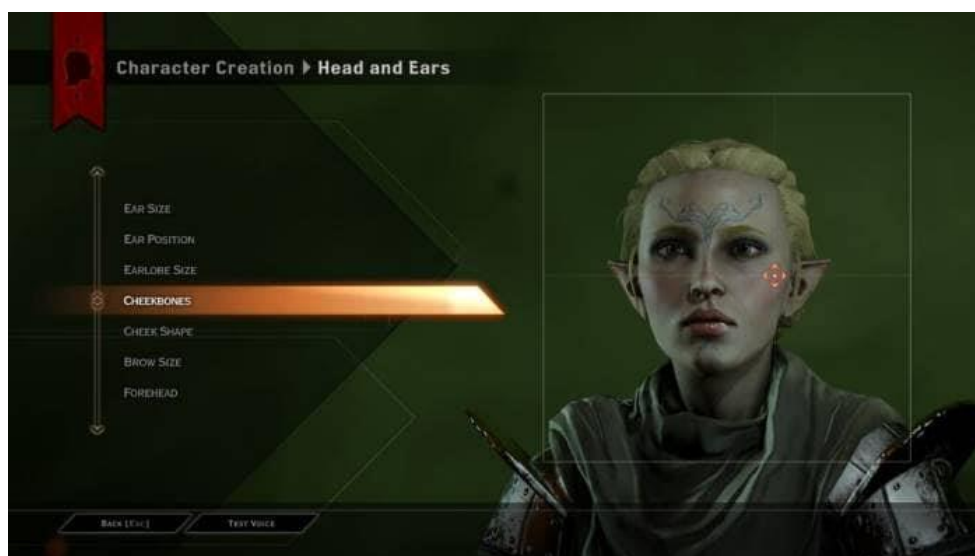


Рисунок 1.7 – Редактор персонажу «Dragon Age: Inquisition»

#### **1.4 Постановка задачі**

Для досягнення поставленої мети, на основі аналізу розвитку відеоігор та занурення у них, потрібно розробити програму, яка на основі зовнішніх даних людини створює 3D модель користувача за допомогою машинного навчання, таким чином репрезентуючи його аватар у створеному всесвіті ігрового застосування. Для цього потребується розробити ігровий застосунок та генератор 3D моделі персонажів.

Практична частина роботи складається у розробці програмного забезпечення, яка на вхід приймає 2D зображення, а на виході результатом становиться 3D-модель та ігрового застосування, який підтримує імпорт 3D моделей через інтерфейс розробника.

Прикладом успішної роботи проекту є схожість гравця та аватара, який його ототожнює у ігровому застосуванні. Схожість не може бути виміряна технічним методом, тому успішність роботи оцінюється кінцевим користувачем. Персонаж повинен репрезентувати гравця, а саме викликати почуття «спорідненості» з репрезентуючим його аватаром та впевненість, що останній представляє його особистість у межах ігрового світу.

#### **Висновки до розділу 1**

У даному розділі було проведено аналіз історії розвитку відеоігор з метою розглянути чому і коли з'явилась потреба у зануренні до створеного всесвіту. Було проаналізовано розвиток занурення у відеоігри з метою визначити як сильно впливає зануреність на ігровий досвід користувача. Поставлена задача роботи, яка становить реалізацію ігрового застосування з редактором персонажу та генератора 3D моделі. Висновком чому і коли з'явилась потреба у зануренні і чому вона важлива вважається те, що у ході розвитку відеоігор, користувачів, окрім інтерактивного ігрового процесу, потрібно було зацікавлювати додатковими елементами, які схожі по структурі з творами мистецтва, наприклад, розглянуте явлення «занурення», що надає змогу гравцю більше повірити у створений вимір

гри через ототожнення себе з персонажем, за якого він грає. Такий рівень інтерактивності не може бути досягнуто іншими творами мистецтв. Це безпосередньо впливає на ігровий досвід, викликаючи в гравця більше емоцій, ніж зі звичайним персонажем, та надає можливість гравцю відчувати персонажа, хвилюватися за нього та його перемоги та поразки переносити на себе. Було поставлено питання оптимальності підходу до реалізації програми для формування 3D моделі та ігрового застосунку, які виконуються одночасно. Підставою для цього питання є допущення, що одночасне виконання ресурсоємних програм може затрачувати велику кількість ресурсів, яких кінцевий користувач може не мати, або велику кількість часу, що не підходить для користувача ігрового застосунку.

## 2 АНАЛІЗ ВПЛИВУ ШТУЧНОГО ІНТЕЛЕКТУ НА ВІДЕОІГРИ

### 2.1 Роль і технології машинного навчання у відеоіграх

Штучний інтелект застосовувався до ігор із самих ранніх днів — від традиційних ігор, таких як шашки, до сучасних стратегічних ігор у реальному часі, як-от StarCraft II. У цій статті ми розглянемо, як штучний інтелект був застосований до покрокових ігор і ігор у режимі реального часу, і що є передовим краєм прикладного машинного навчання для ігор.

Ігри мають довгу історію спільної еволюції AI та машинного навчання. Побудова синтетичних світів в іграх послужила корисним тестовим стендом для алгоритмів AI. Крім того, застосування штучного інтелекту в іграх дозволяє створювати конкурентоспроможних супротивників, щоб кидати виклик гравцям.

Перший відомий приклад AI в іграх був натхненний як демонстрація обчислювальної техніки на Фестивалі Британії.[10] Комп'ютер Nimrod, виготовлений на замовлення від Ферранті, був представлений у 1951 році і використовував гру Nim для демонстрації своїх математичних можливостей. Nim — це гра для двох гравців, де гравці по черзі видаляють один-три об'єкти з колекції об'єктів. В одному варіанті гравець, який видаляє останній предмет, стає переможцем. Nimrod використовував 480 вакуумних ламп, попередників транзисторів, для обробки та відображення.

Хоча Nimrod був спеціально розроблений для гри в Нім, Ферранті вважав, що створення машини для гри у відносно складну гру призведе до вирішення складних проблем. У Nimrod не було програми, а скоріше запрограмований набір логіки для гри. Німроду не вистачало звичайного штучного інтелекту, але його евристичні особливості гри в Нім зробили його конкурентоспроможним у грі, що здивувало та лякало гравців та глядачів.

Одне з найперших застосувань традиційного машинного навчання в іграх було реалізовано Артуром Семюелом з ІВМ (який ввів термін «машинне навчання»). У 1956 році Семюел взяв на себе завдання шашки, яке включає просту

гру, а також складну стратегію. Її розробка на першому комерційному науковому комп'ютері IBM, IBM 701, створила дві ключові концепції машинного навчання для ігор. Першим був винахід альфа-бета обрізки. Патрік Вінстон так описав основну ідею альфа-бета: «Якщо у вас є ідея, яка, безперечно, погана, не витрачайте час, щоб побачити, наскільки вона дійсно жахлива».

Альфа-бета-обрізка — це оптимізація пошуку дерева для алгоритму Minimax, що використовується для мінімізації кількості вузлів, які оцінюються в дереві — у цьому випадку в контексті ігор для двох гравців. [11] У двох словах, алгоритм Minimax намагається максимізувати мінімальний виграш, коли комп'ютер вибирає хід (представлений у вигляді дерева простору стану гри, де кожен рівень дерева є ходом гравця або комп'ютера). Альфа-бета вилучає частини дерева з оцінки, оскільки вони не покращують нагороду гравця. У прикладі (див. Рисунок 2.1) лише середній хід просуває гравця X до перемоги (за умови, що обидва гравці грають оптимально).

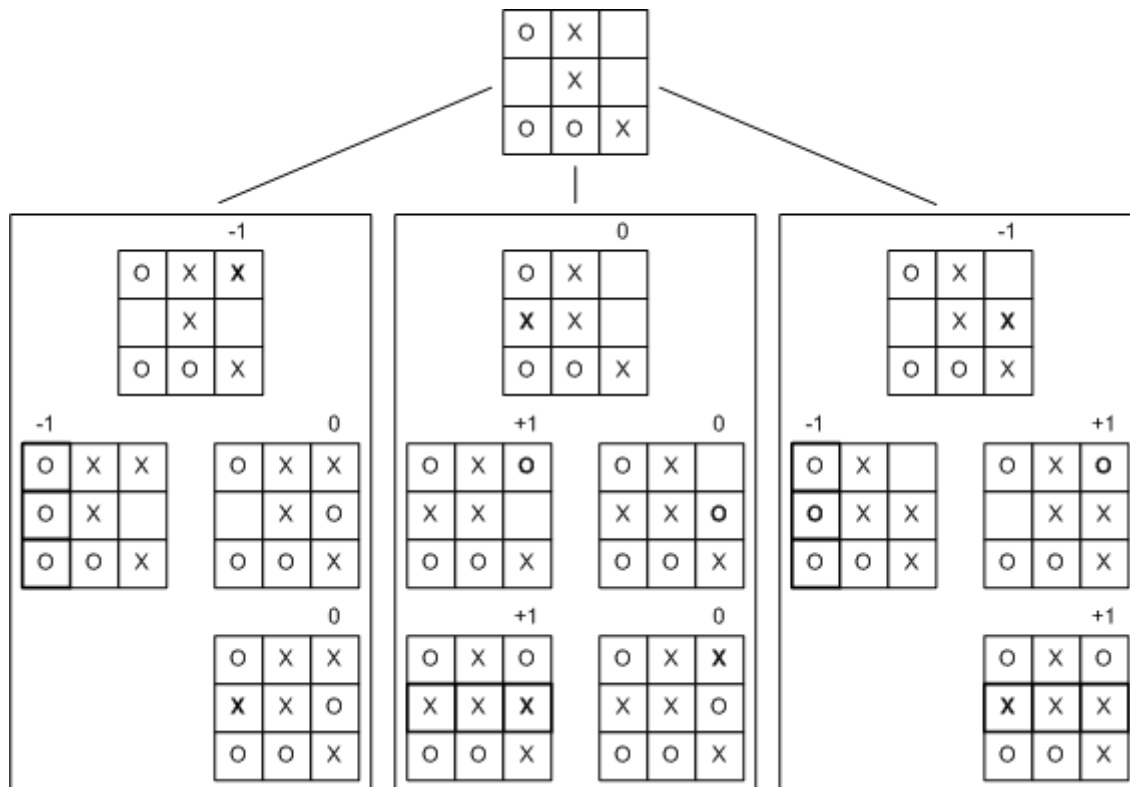


Рисунок 2.1 – Дерево мінімакс для гри у хрестики-нулики

Другою ключовою концепцією, яку винайшов Семюел, була самогра з зачуванням напам'ять. Семюел заставив грати свою програму проти себе, щоб покращити її гру, а також запам'ятати кожен хід, з яким він зустрічався (збережений на магнітній стрічці), разом із кінцевим значенням «винагороди» для оптимізації гри. Його програма грала в шашки на аматорському рівні, але ввела концепції, які використовуються і сьогодні.

Через сорок років альфа-бета-пошук повернувся до гри в шахи. У 1996 році Deep Blue з IBM переміг Гаррі Каспарова, гросмейстера з шахів. Deep Blue виконував пошук дерева стану гри, обрізаного за допомогою альфа-бета-обрізання, але паралельно для збільшення швидкості визначення наступного ходу комп'ютера (пошук 200 мільйонів позицій в секунду до 20 ходів у майбутньому). У 2015 році Google застосував машинне навчання до гри Go, використовуючи глибоке навчання (нейронну мережу) для вибору ходу і пошук Монте-Карло для застосування раніше вивчених ходів. AlphaGo переміг професійного гравця-людину в 2015 році і гравця № 1 у рейтингу в 2017 році. Цікаво, що AlphaGo повторно застосувала самоігру, щоб посилити свою гру (створену Семюелом для шашок майже 60 років тому), а також для збереження раніше відомих ходів.

Ранні відеоігри не мали жодної концепції штучного інтелекту в своїх ігрових супротивниках, оскільки вони поклалися на державні машини для передбачуваного руху (наприклад, у Space Invaders). У 1980 році Pac-Man додав деяку складність ворогам для пошуку шляху до гравця (або подалі від гравця у випадку втечі). Кожен ворог також включав певну мінливість особистості, щоб зробити її менш передбачуваною. [12]

Скінченні автомати (FSM), хоча насправді не є частиною AI, були частиною ігор протягом тривалого часу, забезпечуючи основу для того, що здається розумною поведінкою. Скінченний автомат — це модель для обчислень, яка дозволяє змінювати стани на основі деяких зовнішніх вхідних даних. У контексті ігор FSM забезпечують різну поведінку ігрових сутностей. Приклад (див. Рисунок 2.2) ілюструє негравця (NPC), якому гравець може призначити завдання. Після

призначення NPC виконує завдання на основі різних способів поведінки, необхідних для його виконання.

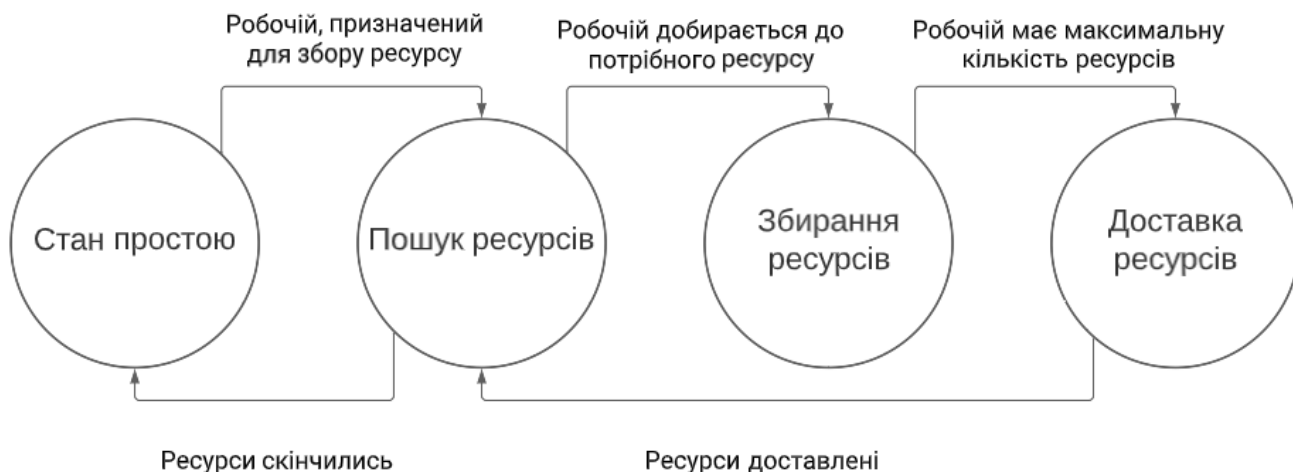


Рисунок 2.2 - Кінцевий автомат для робочого NPC в RTS

FSM, показаний на рисунку 2, представляє робочого NPC, якому гравець призначає завдання збору ресурсів. Після його призначення NPC знаходить потрібний ресурс, збирає його, а потім доставляє ресурс до точки збору. Цей процес триває до тих пір, поки не будуть вичерпані всі ресурси і NPC стане простоювати.

Звичайним елементом ігор є здатність NPC орієнтуватися в навколишньому середовищі. Цю можливість зазвичай називають пошуком шляху і має ряд рішень. Один з найпоширеніших називається А. Алгоритм А є варіантом алгоритму пошуку шляху Дейкстри на основі графів і розглядає місця, які NPC може перейти, як вузли в графі (див. Рисунок 2.3).

A\* працює з двома списками: відкритим списком і закритим списком. Відкритий список містить усі ще не відвідані вузли, а закритий список містить ті вузли, які були відвідані. Починаючи з вузла з відкритого списку, алгоритм знаходить ці вузли, доступні з цього вузла, і додає їх до відкритого списку (якщо вони вже не були додані до закритого списку). Оцінка, обчислена для поточного вузла на основі значення для вузла (проїзд деяких вузлів може бути дорожчим, ніж інших) та евристики поточного вузла (мінімальна відстань від вузла до цілі). Цей

процес триває до тих пір, поки не буде досягнута мета. Опинившись там, алгоритм просто слідує за зменшенням оцінки вузла назад до початкового вузла, щоб визначити шлях.

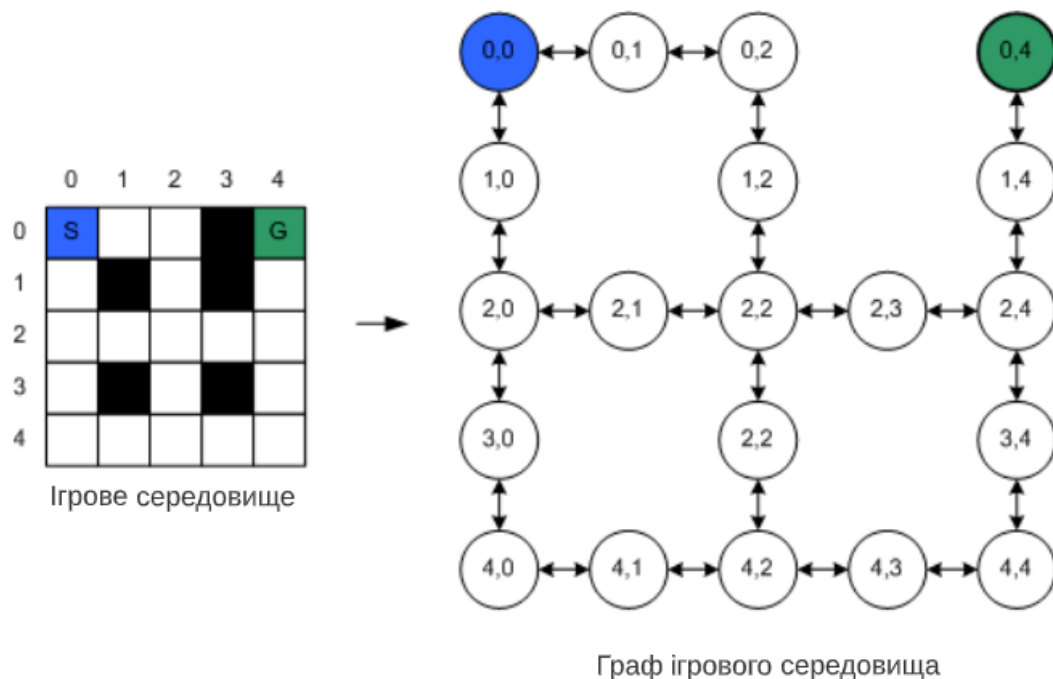


Рисунок 2.3 – A\* реалізує пошук у графі, щоб визначити найкращий шлях

A\* є одним із варіантів пошуку шляху, але існують інші алгоритми. Оригінальний алгоритм пошуку шляху в програмах з великими картами та малими процесорами об'єднав би середовище в більші вузли. Цей підхід, званий ієрархічним пошуком шляху, привів до меншої кількості вузлів, що зводило до мінімуму необхідний пошук.

Age of Empires від Relic Entertainment включала систему, засновану на правилах, яка керувала поведінкою супротивника. Age of Empire — це стратегічна гра в режимі реального часу, де гравець будує цивілізацію, збирає ресурси, будує технології, будує військо, а потім перемагає супротивника. Age of Empires розробила засновану на правилах систему контролю ворожих цивілізацій. Древа



рішень — це споріднений підхід до систем, заснованих на правилах. Дерева рішень кодують рішення (або тести) до наслідків (наприклад, дії, які NPC виконує в грі).

У системі, заснованій на правилах, правила складають базу знань, яка кодує знання експерта. Механізм виведення буде сканувати правила, щоб визначити ті правила, які можуть бути задоволені, і вирішить будь-які конфлікти, що виникають (якщо можна виконати кілька правил, виберіть одне, використовуючи рішення конфлікту). Нарешті, вирішена дія виконується. У наступному прикладі показано правило, яке визначає, що якщо замок можна побудувати (умова), то замок має бути побудований (дія).

Вміст в іграх, як-от їхнє середовище, важливий для ігрового процесу, а певні типи ігор (наприклад, Rogue) покращують можливість відтворення, випадковим чином створюючи середовище для кожного рівня. Такі ігри, як Minecraft, створюють свої надзвичайно детальні світи за допомогою процедурної генерації, а інші ігри, як-от Dwarf Fortress, створюють не лише своє середовище, а й історію та знання для цього ігрового всесвіту.

Підходи до процедурної генерації широкі й різноманітні, але ви можете знайти фрактали, а також використовувати джерела шуму, такі як шум Перліна, для створення більш природних текстур або ландшафтів. Використання процедурної генерації для ландшафтів стало настільки поширеним у відеоіграх, що воно має власну назву: процедурна генерація ландшафту.

Нещодавній цікавий розвиток процедурної генерації ландшафту полягає у використанні передачі стилю. Використовуючи глибокі згорткові нейронні мережі, CNN із глибоким навчанням і певний стиль мистецтва (наприклад, «Зоряна ніч» Ван Гога), місцевість може бути автоматично перенесена на новий стиль. Місцевість використовується як вхідні дані для CNN, який раніше був навчений у стилі Ван Гога, а на виході була нова сцена, відфільтрована з оригіналу. Це трудомісткий процес, тому це не те, що можна зробити в реальному часі, а використовується для візуалізації певного стилю під час розробки гри.

Нагадаємо, що машинне навчання часто використовується як тестовий стенд для алгоритмів. Розглянемо, як алгоритми машинного навчання були застосовані до простих відеоігор.

В якості прикладу згадаємо вже розглянутий у першому розділі Atari's Pong – одну з перших успішних відеоігор, що імітують гру у пінг-понг. У цій простій грі використовувалися весла, щоб «вдарити» м'яч, а весло було розділене на вісім сегментів, щоб змінити кут, під яким м'яч відштовхувався.

Андрій Карпаті застосував глибоке навчання з підкріпленням (RL), щоб створити гравця в понг (див. Рисунок 2.4). У RL алгоритм відчуває оточення і виконує дію у відповідь (припускаючи відкладену винагорода). У цій реалізації середовище RL — це відмінності між двома знімками ігрового екрана в часі (різниці пікселів), які нейронна мережа спостерігає як рух. Після кожного ходу винагорода з оточення становить 1, якщо суперник пропускає м'яч, -1, якщо AI гравець пропускає м'яч, і в іншому випадку винагорода дорівнює 0.

Проста нейронна мережа включає прихований шар із 200 одиниць, який приймає дельту введення зображення (середовища). Вхідний сигнал подається на вихідний шар, який потім визначає ймовірність переміщення лопатки вгору або вниз. У міру отримання винагорода мережа оновлюється методом навчання під наглядом за допомогою градієнтного спуску за стратегіями.

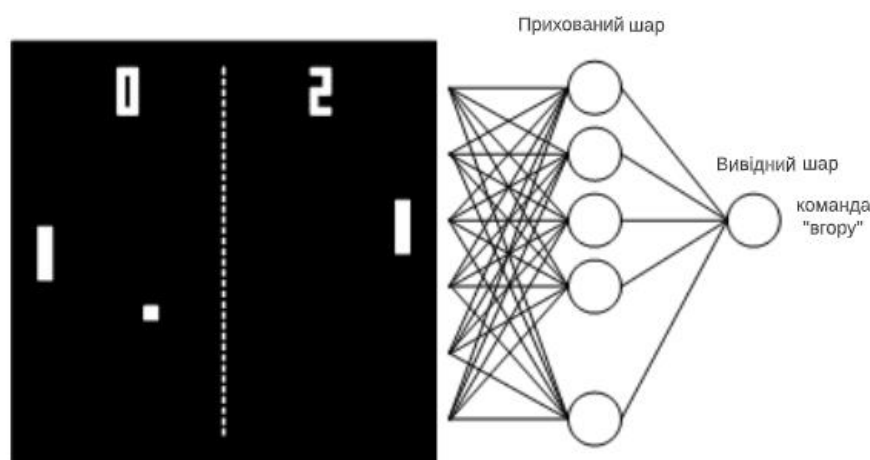


Рисунок 2.4 – Створення гравця в понг за допомогою нейронної мережі

Подібний підхід був застосований до більш складної гри: Super Mario від Nintendo (див. рисунок 2.5). Але замість того, щоб об'єднати пікселі до зменшеної версії та передати дельти зображення в нейронну мережу з прямим зв'язком, Маріо вирішив проблему за допомогою глибокої згорткової нейронної мережі (CNN) і Q-навчання. До поточного зображення екрана відеоігри були введені попередні чотири зображення, на додаток до попередніх введів контролера. Виходи представляли собою входи контролера в гру.

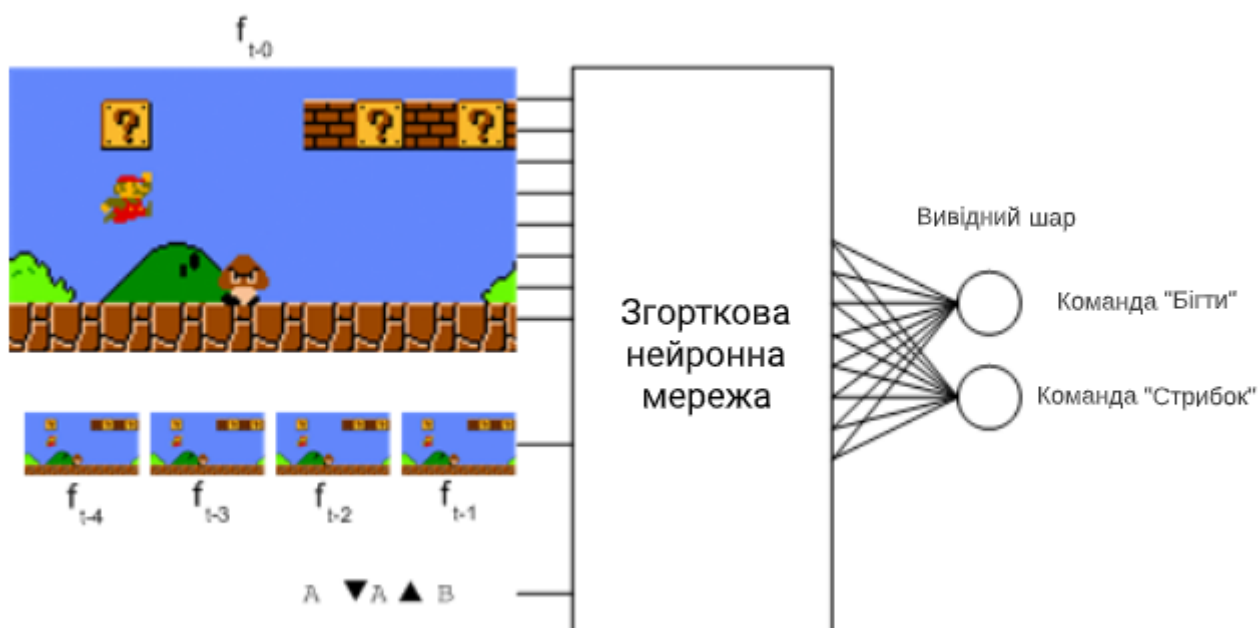


Рисунок 2.5 – Навчання гри в Super Mario за допомогою глибокого Q-learning

AI отримував винагороду не лише за завершення гри, але й за те, що він рухався вправо (на відміну від руху вліво, що було негативно винагороджено). CNN у цьому прикладі був дуже глибоким і становив близько 6,6 мільйонів параметрів.

Останнім прикладом розглянемо розробку гравців із машинним навчанням у грі Dota 2. Dota 2 — це стратегічна гра в реальному часі, де дві команди по п'ять гравців у кожній працюють, щоб знищити структуру опонентів, яка називається «Стародавня» (англ. Ancient). Кожна команда працює, щоб знищити стародавню іншої команди, одночасно захищаючи свою. Гра складається з різних героїв, які мають різні стилі гри (основа (англ. Carry) чи підтримка (англ. Support)). Герої

підвищують рівень, щоб вивчити нові навички та отримати золото, щоб придбати предмети, які мають особливі властивості. Цікавою особливістю Dota 2 є те, що для перемоги потрібна значна командна робота та співпраця.

OpenAI Five розробив штучний інтелект Dota 2, який зміг перемогти чинних чемпіонів світу з кіберспорту Dota 2 у квітні 2019 року. Їхній AI був створений з нуля і не передбачав ніяких притаманних знань про гру чи евристики, щоб керувати її грою. Він навчився грати в гру за допомогою самостійної гри, починаючи з випадкових ваг для своїх нейронних мереж. Навчені AI (яких по п'ять на кожного героя) грали один проти одного в 80% ігор і 20% проти минулих AI. Використовуючи 128 000 ядер ЦП і 256 графічних процесорів, AI щодня разом грають в Dota 2 протягом 180 років. Кожен AI виконує близько 20 000 ходів за гру, що означає довгий горизонт для збору інформації про працездатність екземпляра AI.

OpenAI Five — це фактично п'ять різних нейронних мереж, які координуються за допомогою гіперпараметра, який називається командним духом. Командний дух вказує на те, чи повинні окремі AI зосередитися на своїй індивідуальній нагороді (наприклад, вбивати інших героїв і збирати скарби) чи зосереджуватись на загальній командній нагороді. Принцип роботи AI можна розділити на три частини: сприйняття, обробка та вибір дій (див. Рисунок 2.6). Сприйняття включає різні вхідні дані з середовища, які надходять у вбудовування, яке максимально об'єднується для подачі в центр обробки. Довга короткочасна пам'ять (LSTM) — це рекурентна нейронна мережа, яка є завершеною за Тьюрингом, оскільки може обчислювати щось подібне до обчислень загального призначення. LSTM є потужними в навчанні послідовностей і, отже, ідеально підходять для цієї програми. Потім LSTM виводить на набір мереж дій, які вибирають дію для виконання (разом з іншою відповідною інформацією, наприклад, ціль). При цьому враховуються дії, які можна вибрати на додаток до уваги конкретного героя.



Рисунок 2.6 – AI герою Dota 2 для довготривалого навчання

Вхідні дані в мережу складаються з приблизно 20 000 реальних значень, тому простір сприйняття агента досить великий. Кількість дій, які можна вибрати, також досить велика, близько 170 000 унікальних дій (приблизно 1 000, які зазвичай можливі, зведені до мінімуму через перезарядки, предмети, які не утримуються тощо). OpenAI Five використовував проксимальну оптимізацію політики для навчання, яка, як виявилось, проста у використанні та має хорошу продуктивність у виробничих програмах.

Результатом розвитку AI та ML у ігровій сфері стали технології, які міцно закріпилися в іграх та без яких не обходиться майже ні одна гра, наприклад:

- **розумніші неігрові персонажі.** Традиційно, NPC були запрограмовані на заздалегідь прописані дії за допомогою кінцевого автомата. Це означає, що їхні дії були пов'язані з сюжетною лінією або у відповідь на дії гравця, тому NPC мав обмежені та передбачувані дії. Однак завдяки штучному інтелекту та МО наші NPC тепер можуть вивчати ігровий стиль гравця та мати динамічний набір дій, що робить їх менш передбачуваними та більш складними для гравця. Саме ця стратегія навчання у суперника дозволила нам створити найсучасніші шахові движки, такі як AlphaZero.
- **динамічний рендеринг** Однією з проблем, яку компанії, що займаються відеоіграми, намагаються усунути за допомогою штучного інтелекту та машинного навчання, є спотворення перспективи. Це явище виникає, коли

об'єкт виглядає добре, коли гравець знаходиться далеко, але спотворюється і стає піксельним, коли гравець наближається до цього об'єкта. Ігрові компанії використовують алгоритми машинного навчання для динамічного покращення зображень і візуалізації. Це протидіє ефекту спотворення зображення і дозволить об'єкту виглядати точніше, коли він ближче до програвача.

- **генерація діалогів і реалістичні взаємодії.** Ми вже бачили, як AI та ML можна використовувати для покращення дій NPC. Однак ці технології також можна використовувати для покращення ігрового досвіду шляхом формулювання більш точних і реалістичних відповідей NPC. У ряді рольових ігор використовується діалоговий механізм, який значно покращено за допомогою обробки природної мови та методів аналізу настроїв за допомогою алгоритмів ML. Хороший приклад розширеного діалогового інтерфейсу штучного інтелекту та реалістичних взаємодій можна побачити в таких іграх, як The Elder Scrolls IV: Oblivion.
- **генерація світу.** Іншим потужним застосуванням ML у розробці ігор є використання генерації світу. У ряді популярних ігор, таких як Minecraft і серія Grand Theft Auto, використовується сценарій гри з відкритим світом. Ці ігри було б надзвичайно важко створити без певних функцій генерації світу. Вона дозволяє динамічно картувати місцевість, породжувати NPC та приховувати здобич за допомогою технології машинного навчання.
- **створення захоплюючих ігор.** Одним із найвищих пріоритетів розробників відеоігор є створення гри, яка максимально занурює та наближається до реального світу. Однак моделювання реального світу може бути неймовірно важким процесом. Цей процес можна значно полегшити за допомогою технології машинного навчання. Алгоритм ML можна використовувати, щоб передбачити наслідки дій гравця або навіть моделювати такі речі, як погода в грі.

AI змінює те, як ми граємо в ігри, а також те, як створюються ігри. Застосовувані методи еволюціонували, але деякі підходи залишаються незмінними донині, як-от ідея Артура Семюела про самогру для створення навчальних агентів. Від шашок і IBM 701 до складних ігор у реальному часі, які навчаються за допомогою розподілених мереж ЦП і графічних процесорів, машинне навчання є невід'ємною частиною ігор і тестовим стендом для майбутніх підходів машинного навчання.

## 2.2 Вплив машинного навчання на 3D моделювання

Створення 3D-об'єктів може бути складним і трудомістким завданням для тих, хто не має попереднього досвіду в 3D-моделюванні. Це може бути великою проблемою для ігрових движків, оскільки розробникам доводиться або вивчати 3D-моделювання, або платити за моделі. Якби моделі натомість створювалися автоматично, процес розробки був би легшим і міг би призвести до того, що більше розробників використовують ігровий движок. Щоб зробити автоматичне генерування 3D життєздатним, потрібно дотримуватись трьох вимог, щоб програма була зручною для споживачів:

- Якість, вихідна роздільна здатність.
- Швидкість, час генерації.
- Простота, зручність у використанні.

Наприклад, можна створити тривимірний об'єкт, використовуючи одне вхідне зображення та згорткові нейронні мережі. Це буде вважатися досить простим для третьої вимоги, зручної для споживача. Що потрібно перевірити, так це обмеження вихідної роздільної здатності та часу генерації, які не зазначені в жодній із документів, на які посилаються вище.

Хоча використання нейронних мереж для розпізнавання двовимірних зображень на даний момент є одним із провідних додатків ML, обробка 3D все ще чекає свого великого моменту. Основні проблеми полягають у великій кількості даних, які постачаються з кожною тривимірною формою, і потребі в подальшій

обробці згенерованих фігур, щоб зробити їх застосовними. Тож фактично, незважаючи на існуючу автоматизацію, все ще потрібна велика кількість людської ручної роботи, щоб створити красиву і придатну для використання 3D-форму, будь то літак, крісло чи автомобіль.

Це накладає суворі обмеження на використання нейронних мереж в 3D-аналізі та обробці – в той час як штучний інтелект може виявляти рак на медичних сканах з надлюдською точністю, а досягнення такого ж ефекту за допомогою 3D-моделі залишається справою майбутнього.

Завдяки останнім дослідженням вчені з Вроцлавського університету науки і техніки створили нейронну мережу, яка автоматизує створення 3D-об'єктів (сітки) з вхідних даних. Всупереч поширеним підходам, ця нейронна мережа дає цілком зрозумілий результат.

Попередні підходи до обробки 3D за допомогою AI використовували парадигму навчання з наглядом, яка вимагала від науковців даних попередньо обробити та підготувати набір даних для навчання машині. Враховуючи доступність даних і необхідну величезну кількість роботи, цей підхід, м'яко кажучи, викликає труднощі.

Крім того, виконання всієї цієї роботи вручну є стомлюючим і дорогим заняттям, що вимагає обширних знань в області, багато робочих годин і великої кількості обчислювальної потужності. Тож надання ключа до автоматизації здається чудовою та ще не повністю вивченою сферою досліджень. Передові підходи, які використовувалися раніше, все ще залишалися чорною скринькою – хоча результати були доступні, ні користувач, ні дослідник не мали уявлення про те, як вони були отримані. А незрозумілий процес штучного інтелекту завжди несе певний ризик, коли незвичайна комбінація факторів викликає несподіваний збій.

Дослідження цього розділу складається з двох нейронних мереж. Перший – це кодер, який перетворює тривимірний об'єкт у багатовимірний вектор. Таким чином, форма стає зрозумілою для нейронної мережі, яка в основному є надзвичайно складним інструментом для виконання обчислень.



Мережу кодера можна вважати еквівалентом машинного навчання сітківки, яка перетворює вхід світла в нейронні імпульси так само, як мережа кодера кодує тривимірну форму у вектор. Позитивним побічним ефектом мережі кодерів є надзвичайне зменшення розміру вхідного сигналу з повноцінної тривимірної форми в рядок чисел. Завдяки такому перетворенню будь-яка передача може відбуватися набагато ефективніше.

Другим кроком є обробка входу в мережі декодера. Використовуючи вхідні дані, отримані нейронною мережею кодера, декодер видає дерево рішень, яке показує, як форму була отримана за допомогою набору примітивів.

«Примітив» — це термін, який використовується в тривимірній геометрії для опису найпростішої тривимірної фігури, яку неможливо зменшити далі, як куб, кубоїд, овал або куля. Нейронна мережа, створена дослідницькою групою Тоорлоох, розглядає примітиви як базовий набір будівельних блоків, які використовуються для відновлення форми, декодованої мережею декодера. [13]

Мережа маніпулює розмірами та положеннями примітивів, щоб надати більш витончені та складні форми. З точки зору мережі, найпростіше зображення будинку складається з куба, піраміди та кубоїда, які вставляють у «дах», щоб відобразити димар.

Детальну інформацію про те, як мережа створює певну форму, можна побачити на графі нижче (див. Рисунок 2.7):

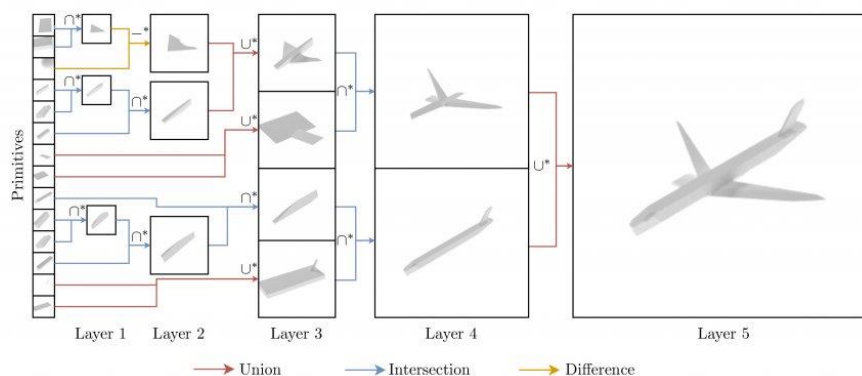


Рисунок 2.7 – Отримане дерево CSG з прогнозу реконструкції літака з набору перевірки.

Під час проекту перед мережею було поставлено завдання зібрати разом фігури літака, дивана з набором подушок, корабля, телевізора, лавки, столу та крісла (див. Рисунок 2.7).

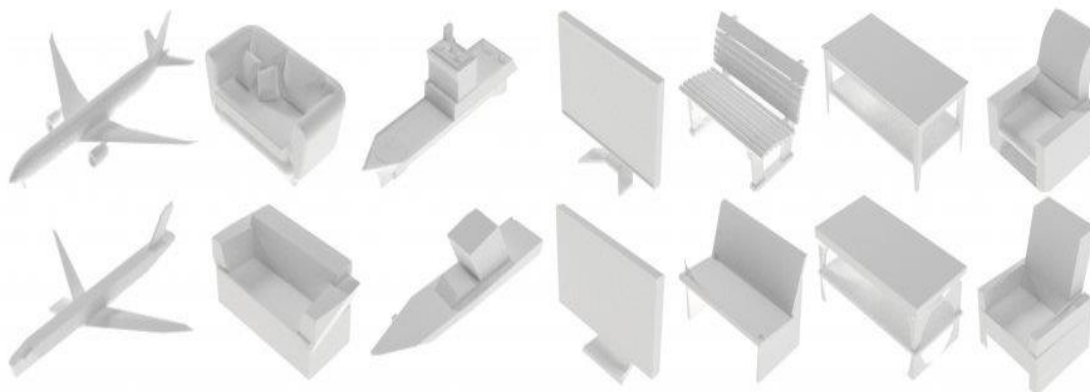


Рисунок 2.8 – Реконструкції поверхні для завдань 3D автокодування.

Використання набору примітивів як будівельних блоків, окрім зменшення обчислювальних зусиль, необхідних для створення тривимірної сітки об'єкта, має цінність, якою не можна нехтувати – пояснюваність.

Поширене припущення щодо нейронних мереж полягає в тому, що творець майже не має уявлення про процес, що стоїть за результатом. Це лише частина правди. Творець завжди має інформацію про вихідний набір даних і парадигми, які він або вона використовував при проектуванні мережі.[14]

Однак обробка, що виконується всередині мережі, зазвичай приховується в тіні. Не завжди потрібно точно знати, що стало причиною того чи іншого рішення мережі, якщо результат задовольняє.

З іншого боку, однак, відсутність пояснення може бути небезпечною, коли потрібна 100% впевненість, наприклад, під час критичних операцій або охорони здоров'я. Нейронна мережа, описана в цьому розділі, надає не тільки тривимірну форму об'єкта, але й інструкції щодо того, як це було зроблено, які примітиви були використані, де і що було змінено, включаючи такі операції, як зміна розміру, згортання або вирізання певних форм.

## Висновок до розділу 2

У другому розділі було проведено аналіз впливу машинного навчання на відеоігри. Було встановлено, що не тільки машинне навчання впливало на ігри, але й відеоігри впливали на машинне навчання. Маючи спільний еволюційний шлях, вони покращували один одного, так як синтетичні світи в іграх послужила корисним середовищем для тестів алгоритмів AI, а останній в свою чергу дозволяє створювати конкурентоспроможних супротивників, щоб кидати виклик гравцям, допомагає з побудовою ігрових елементів, наприклад, згенерований ландшафт та діалоги.

Штучний інтелект у сфері 3D моделювання поки чекає свого часу, так як все ще потрібна велика кількість людської ручної роботи, щоб створити красиву і придатну для використання 3D-форму, тому що, але не зважаючи на це, AI активно застосовується у сфері 3D моделінгу.

## 3 ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ПРОБЛЕМИ РЕПРЕЗЕНТАЦІЇ ГРАВЦЯ

### 3.1 Технології для створення ігрового додатку

Коли справа доходить до розробки ігор, вибір правильного ігрового движка для гри є одним із найважливіших питань, яке вплине як на саму гру, так і на досвід, який розробник отримає. Також слід враховувати багато факторів: Яку гру потрібно створити? Чи ця гра у 2D чи 3D просторі? Наскільки розробник досвідчений? Наскільки активною є спільнота движка або фреймворка? Які ресурси доступні для двигуна? З огляду на всі ці фактори, які потрібно враховувати, вибрати правильний фундамент для роботи може бути справжньою проблемою. Це особливо актуально в сучасну епоху, коли технології постійно розвиваються, а це означає, що нові двигуни та оновлення постійно виходять з новими інноваціями.

У ході дослідження було складено список найкращих ігрових движків на момент розробки бакалаврської кваліфікаційної роботи, а також сильні та слабкі сторони кожного движка на основі деяких із згаданих факторів.

Перед вибором ігрового движка важливо визначити, що таке ігровий движок. Ігровий движок — це просто частина програмного забезпечення, яка постачається з інструментами, спеціально орієнтованими для створення ігор. Хоча це визначення є трохи загальним, для цього є причина: інструменти, надані кожним двигуном, можуть сильно відрізнятися. Однак у більшості випадків інструменти ігрових движків призначені для того, щоб спростити процес розробки гри, реалізувавши найбільш стандартні потреби у сучасній грі.

Наприклад, у минулі часи, як можна було б відобразити на екрані щось настільки складне, як 3D-модель? 3D-моделі — це просто купа крихітних квадратів із призначеними значеннями кольорів (тобто пікселі). Таким чином, по суті, нам потрібно сказати комп'ютеру, як взяти те, що ми бачимо з 3D-моделі, перетворити його в 2D-зображення, а потім призначити всі ці маленькі значення кольору на нашому екрані. У ці давні часи вам довелося б багато програмувати вручну. Можна

уявити, наприклад, що потрібно додати до цього такі речі, як обчислення освітлення, анімацію тощо. У результаті ви отримуєте складний сценарій, написання якого займає години, і, зрештою, це лише невелика фонові частина, яка не має нічого спільного з фактичними механіками відеоігор.

Ігрові движки забезпечують попередньо вбудовані функції, які значною мірою спрощують розробку ігрового застосунку, наприклад:

- відтворення графіки
- розрахунок фізики
- обробка виявлення зіткнень (колізій)
- відтворення анімації та звуків
- компіляція коду

Таким чином, ігрові движки чудово підходять, якщо потрібно зосередитися на логіці та естетиці відеоігор, які, можливо, є найважливішою частиною розробки ігор.

Перш ніж перейти далі, потрібно зупинитися на іншій темі: ігрових бібліотеках та фреймворках. Хоча часто використовуються як синоніми, коли справа доходить до теми ігрових движків, вони технічно відрізняються і варті обговорення. У той час як движок відеоігор є повноцінним програмним забезпеченням, ігрові бібліотеки та фреймворки — це більш-менш просто програмні скрипти. Таким чином, замість створення в рамках окремої програми, ігрові бібліотеки та фреймворки дозволяють створювати ігри за допомогою більшості редакторів коду. У цьому випадку потрібно імпортувати, застосувати або інстальювати бібліотеку/фреймворк відповідно до наданих для неї інструкцій. Так чому це має значення? Що ж, між ними є деякі плюси і мінуси. З позитивного боку, ігрові бібліотеки та фреймворки мають набагато нижчу точку входу. Не потрібно турбуватися, чи комп'ютер може запускати певну частину програмного забезпечення. Їх також дуже легко інтегрувати в проекти. Крім того, незважаючи на те, що це лише набір даних і попередньо запрограмований код, ви все одно отримуєте багато тих же переваг, що й ігровий движок. Наприклад, багато ігрових

бібліотек цілком здатні обробляти такі речі, як виявлення зіткнень і фізика. Таким чином, розробник все одно не застрягне в тій самій втомі розробки, як якщо б він програмував з нуля.

При цьому ігрові бібліотеки та фреймворки мають свої недоліки. Оскільки вони не є повноцінним програмним забезпеченням, вони просто не мають такої кількості зручних інструментів. Натомість розробник покладається виключно на виклики імпортованого коду для кожного елемента функціональності, і це може бути багато викликів залежно від типу гри, яку розробник намагається створити. Аналогічно, більшість ігрових бібліотек і фреймворків просто не настільки потужні, як ігри, створені на движку.

При цьому ні в якому разі не потрібно применшувати ігрові бібліотеки та фреймворки та їх зручність. Вони все ще є фантастичним варіантом для створення ігор. Проте важливо зазначити, що існують ключові відмінності, коли стає вибір найкращого движка відеоігор.

## Unity

Ігровий движок Unity, який розробляється з 2005 року, став основним продуктом індустрії інді-ігор. Оскільки щороку додаються постійні оновлення та нові основні функції, такі як Unity Reflect, підтримка движка неймовірна. Движок не тільки добре підходить як для 2D, так і для 3D ігор будь-якого типу, але він також є популярним вибором для створення ігор віртуальної реальності та розробки AR, а також завдяки багатьом компаніям і розробникам, які створюють зручні SDK для движка.[15]

Крім цього, Unity також має значну спільноту з магазином ігрових ресурсів (ассетів) із безкоштовними та платними варіантами. Оскільки це такий надійний двигун і безкоштовний для розробників, які заробляють менше 100 тис. доларів на рік, це добрий вибір для новачків, незалежно від того, що вони хочуть розробити.

При цьому, якщо ви хочете розробити цілу студію відеоігор навколо Unity, ліцензії можуть бути дорогими, хоча вони мають більше функцій. Крім того, Unity

може бути більш важким для вашої системи, якщо ви використовуєте деякі з високотехнологічних демонстрацій, які показують всі можливості движка. Нарешті, варто зазначити, що оскільки Unity так часто оновлюється, може бути легко пропустити нові функції або важко знайти старі, оскільки інтерфейс користувача та система доступу до них можуть змінитися (див. Рисунок 3.1).

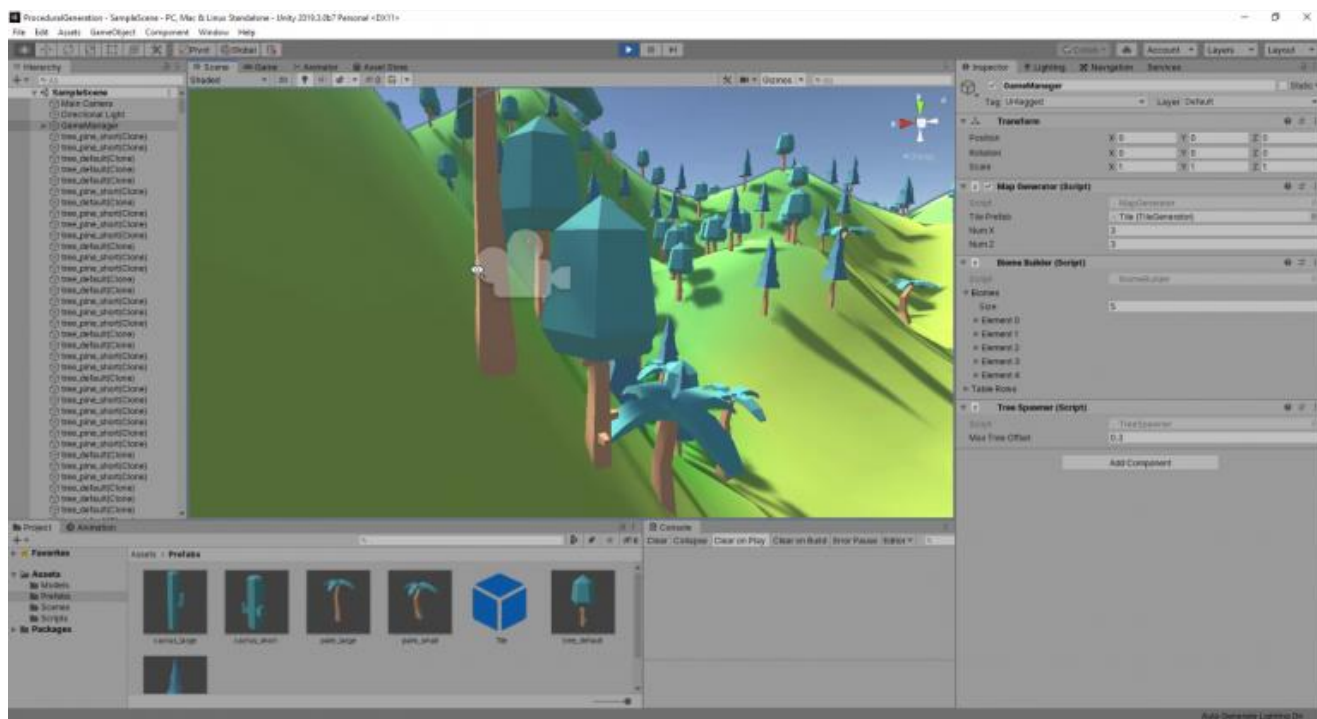


Рисунок 3.1 – Програмний інтерфейс Unity

Сильні сторони ігрового движка Unity:

- безкоштовно для розробників, які заробляють менше 100 тисяч доларів
- чудово підходить для 2d та 3d ігор
- потужна підтримка розробки мобільних ігор
- доступність VR I AR SDK Asset Store з великою кількістю безкоштовних активів

слабкі сторони:

- дорогі ліцензії для професіоналів
- для демонстрацій ресурсоємних технологій потрібні потужні комп'ютери
- багато змін інтерфейсу

Деякі приклади ігор, створених за допомогою Unity:

- Osiris: New Dawn
- Hollow Knight
- Rick and Morty: Virtual Rick-ality
- Genshin Impact

## Unreal Engine

Завдяки своїм потужним графічним можливостям із освітленням, шейдерами та іншим, Unreal Engine є основою багатьох найпопулярніших сьогодні ігор із трьома А (Triple A Game, високобюджетна гра). Враховуючи його широке використання в цьому секторі, двигун був розроблений спеціально для вирішення багатьох складних завдань так, щоб виконувати їх ефективніше, ніж інші двигуни. Як і інші в цьому списку, двигун також є відкритим, що означає, що спільнота також постійно вдосконалює двигун.[16]

Поряд зі своїм візуальним планом (див. Рисунок 3.2), завдяки якому навіть непрограмісти можуть розробляти свої ігри, Unreal дійсно є потужним движком, який здатний практично на все, включаючи VR. Як і в Unity, в Unreal існує Marketplace, де ви можете отримати безкоштовні ассети.

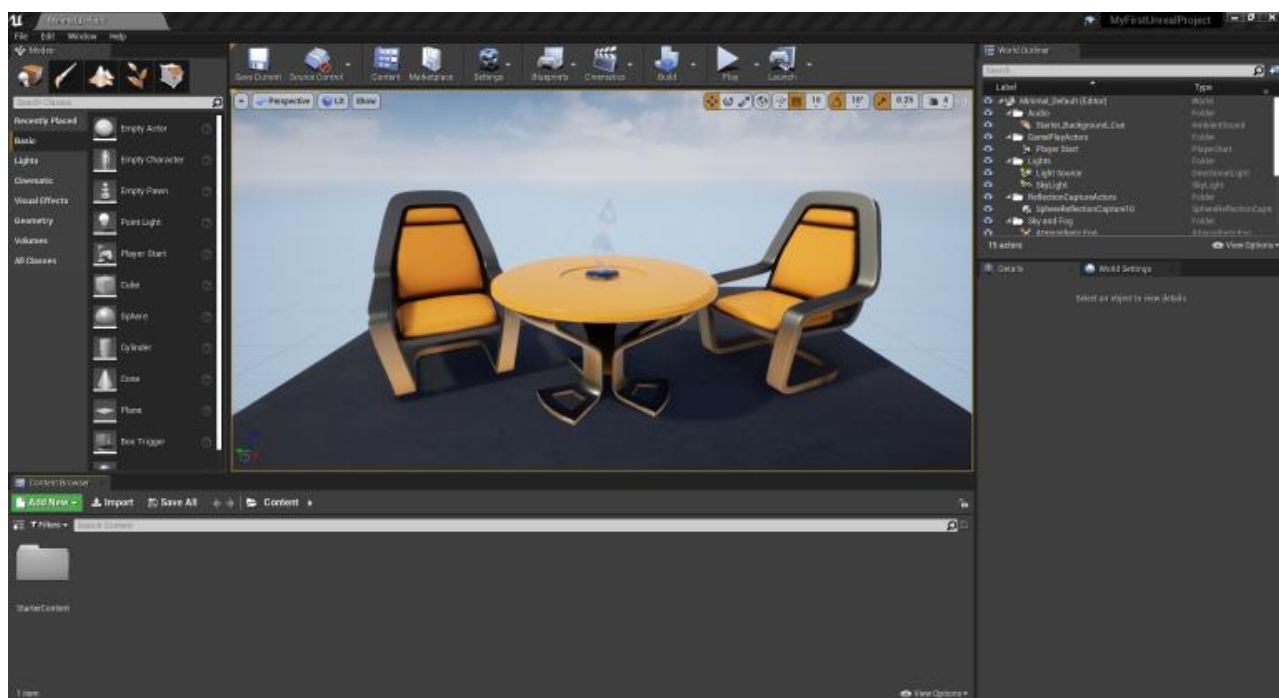


Рисунок 3.2 – Програмний інтерфейс Unreal Engine



Однак багато розробників ігор повідомляють, що Unreal Engine краще підходить для великих проектів, над якими потрібно працювати у команді. Крім того, оскільки програма важка для графіки, багато розробників знайдуть, що для движка потрібен потужний комп'ютер порівняно з іншими двигунами, такими як Unity. Крім цього, варто відзначити, що, хоча Unreal Engine може створювати 2D ігри на додаток до 3D ігор, цей движок не обов'язково найкраще підходить для цієї задачі.

Сильні сторони движка Unreal Engine:

- чудово підходить для високоякісної графіки
- більш продуктивні, ніж інші двигуни
- найкращий вибір для VR
- візуальний план для не програмістів
- великий ринок з безкоштовними ассетами

Слабкі сторони:

- не найкращий для простих або сольних проектів
- високоякісна графіка вимагає більш потужних комп'ютерів
- краще для 3D, ніж 2D ігор

Приклади ігор, створених за допомогою Unreal Engine:

- Soulcalibur VI
- Gears 5
- Dragon Quest XI

## **Godot**

Незважаючи на те, що Godot існує з 2014 року, лише нещодавно двигун дійсно набрав популярності. Ігровий движок Godot є хорошим вибором, якщо розробник шукає безкоштовний двигун з відкритим кодом, що означає, що ви можете змінити движок і продавати свої ігри, як вам заманеться. Движок підтримує

як 2D, так і 3D можливості, тому він добре підходить для будь-яких ігор, які розробник намагається створити.[17]

Що стосується слабких сторін, Godot використовує для написання сценаріїв свою особисту мову програмування, яка називається GDScript (див. Рисунок 3.3). Хоча мова працює ефективно і нагадує Python (улюблена мова для багатьох розробників), оскільки вона була спеціально розроблена для Godot, досвідчені розробники ігор можуть вважати, що адаптація до іншої мови буде складною. Крім того, оскільки Godot не так відомий, як деякі інші движки, доступних ресурсів не так багато в порівнянні з основним двигуном, таким як Unity або Unreal.

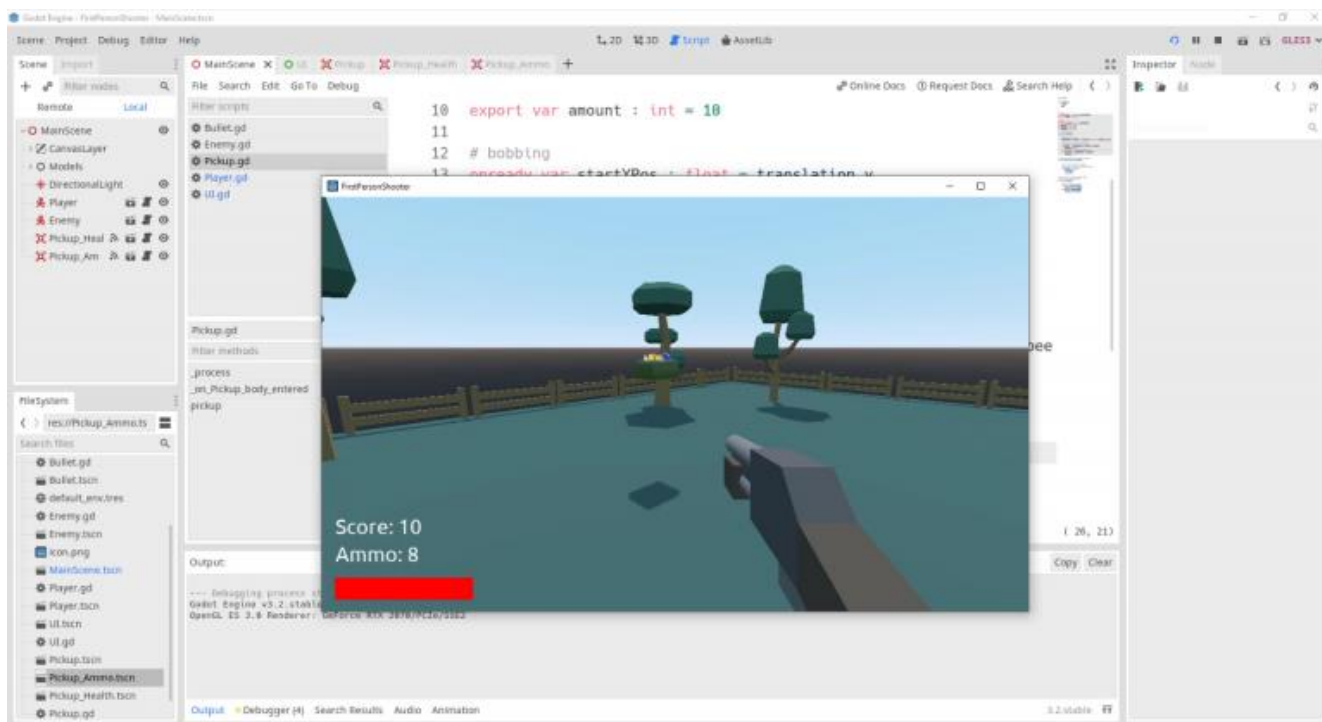


Рисунок 3.3 – Програмний інтерфейс Godot та мова програмування GDScript

Сильні сторони Godot:

- працює з 2D і 3D іграми
- повністю безкоштовний і з відкритим кодом
- комерційно відкритий
- активна спільнота
- унікальна архітектура для розробки ігор

Слабкі сторони:

- GDScript може не сподобатися досвідченим розробникам ігор
- не так багато ресурсів, як в інших двигунах

Приклади ігор, створених за допомогою Godot:

- ProtoCorgi
- Deep Sixed
- Gun-Toting Cats

## GameMaker Studio 2

Цей популярний движок відеоігор, створений у 2017 році, є найновішою версією GameMaker Studio, яка існує з 1999 року і з того часу змінила багато імен. GameMaker широко підтримується на багатьох платформах, включаючи Nintendo Switch. Це також дуже зручний вибір для тих, хто не має досвіду програмування, оскільки в основному використовує візуальну мову сценаріїв із перетягуванням, щоб дозволити користувачам будь-якого рівня навичок створювати ігри своєї мрії.

Однак для тих, хто віддає перевагу кодуванню, він також пропонує свою мову GameMaker для програмування користувацької поведінки (див. Рисунок 3.4), яка виходить за межі того, що може охоплювати візуальне програмування. Загалом, движок дуже зручний для початківців і відкриває розробку ігор практично кожному.[18]

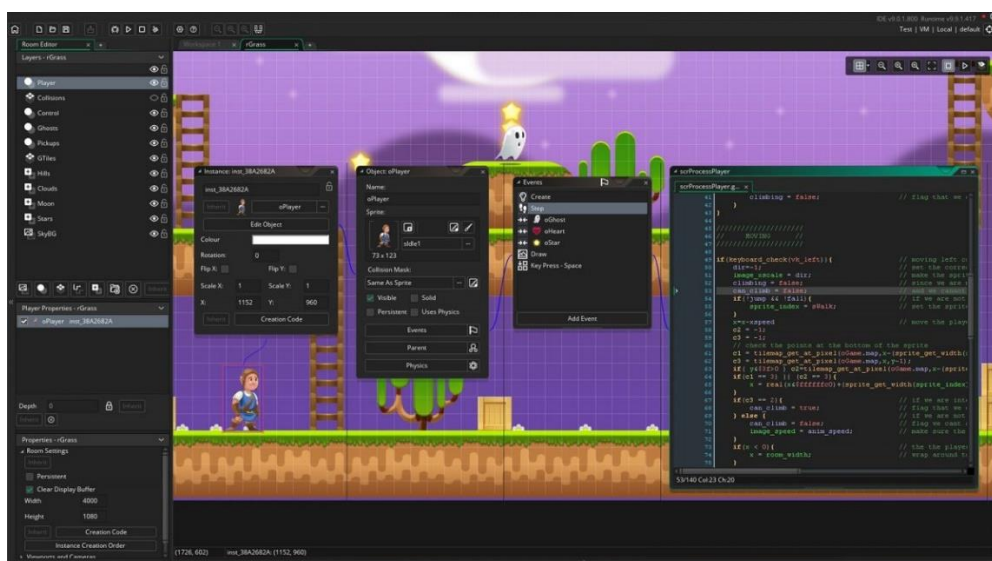


Рисунок 3.4 – Програмний інтерфейс GameMaker Studio 2 та мова GameMaker

На відміну від багатьох інших движків і фреймворків у цьому дослідженні, GameMaker Studio 2 є запатентованою, тому цей движок може бути не найкращим вибором для тих, хто шукає щось з обмеженим бюджетом. Крім того, він призначений спеціально для 2D-ігор. Незважаючи на те, що він має обмежені можливості 3D, він не схожий по взаємодії з 3D на движки Unity, Unreal або Godot.

Сильні сторони движка GameMaker Studio 2:

- підтримується на багатьох платформах
- легке програмування за допомогою перетягування
- надзвичайно зручний для початківців

Слабкі сторони:

- більше орієнтований на 2D ігри
- отримання движка коштує грошей

Приклади ігор, створених за допомогою GameMaker Studio 2:

- Hyper Light Drifter
- UNDERTALE
- Forager

### **Вибір движка для створення ігрового застосунку**

Загалом, ці чотири ігрових движків, що наведені вище, — лише деякі з багатьох движків, доступних для розробки ігор. Однак, оскільки ці движки та фреймворки працюють у багатьох із найпопулярніших інди та високобюджетних ігор, можна бути впевненим в їх якості та здатності працювати майже з будь-якою грою, яку розробник захоче створити. Звісно, у кожного є свої сильні та слабкі сторони, але всі вони витримали випробування часом і залишаються головними претендентами на найкращий движок розробки ігор у 2022 році. Також немаловажним фактором являється вибір найкращої мови програмування для розробки ігор, яка допоможе розробнику зробити цей вирішальний вибір.

Для фінального вибору движка треба розробити порівняння на основі отриманих даних сильних та слабких сторін ігрових движків з метою визначити найкращий движок для виконання цілей даної бакалаврської кваліфікаційної роботи (див. Таблиця 3.1).

Таблиця 3.1 – Вибір ігрового движку

Ігрові движки	Unity	Unreal Engine	Godot	GameMaker Studio 2
Основна мова програмування	C#	C++	GDScript	GameMaker
Робота з 3D	Добра	Добра	Середня	Погана
Підтримка ассетів	Добра	Середня	Середня	Середня
Простота розробки/кількість навчальних матеріалів	Добра	Середня	Нижче середньої	Нижче середньої

На основі критеріїв, зазначених у таблиці 3.1, було прийнято рішення обрати ігровий движок Unity, так як він має добру підтримку роботи з 3D простором, велику кількість ассетів через магазин Asset Store, доволіно простий у розробці та має велику кількість навчальних матеріалів, що в сумі дає змогу швидко і якісно створити ігровий застосунок. Немаловажним фактором являється мова програмування, яку підтримує Unity – мова програмування C#. Так як ця мова є відносно простою для освоєння, універсальною та потужною, ця мова програмування добре підходить для розробки відеоігри.

### 3.2 Технології для побудови 3D-моделі

Програмне забезпечення для 3D-моделювання включає програми, які розробляють базові тривимірні моделі об'єктів або персонажів, і часто включають додаткові функції для конкретизації моделей з реалістичними деталями. Продукти 3D-моделювання можуть створювати моделі з різноманітними підходами та інструментами, а також часто включають функції 3D-малювання для додавання текстур і кольору. Ці програми використовуються в не тільки в індустрії відеоігор, але і в індустрії телебачення та кіно, маркетингу та віртуальної реальності. Після

створення за допомогою програмного забезпечення для 3D-моделювання, моделям можна надати життя за допомогою інструментів 3D-рендерінгу та 3D-анімації. Графічні дизайнери можуть використовувати грубі 3D-моделі без візуалізації для візуального використання на веб-сайтах і медіа-кадрах. Інструменти з можливостями 3D, які спеціально призначені для проектування та архітектури будівель, проектування цивільного будівництва чи інших функцій, пов'язаних із САПР, не входять до категорій дослідження цього розділу, так як вони не підходять для створення 3D-моделі, яку потрібно створити за завданням бакалаврської кваліфікаційної роботи.

У цьому розділі будуть розглянуті продукти, які повинні:

- надати інструменти для побудови та модифікації 3D графічних моделей
- дозволити користувачам зберігати та експортувати готові моделі
- не бути орієнтованими спеціально на проектуванні та розробці архітектури будівель, проектуванні цивільного будівництва чи САПР.

Існує кілька різних факторів, які впливають на індустрію 3D-моделювання, але, зрештою, відмінність між посереднім і орієнтованим на деталі дизайном має значення, і у цьому розділі досліджуються найкращі програмні забезпечення для цих цілей. Що стосується можливостей, то індустрія 3D-анімації та рендеринга розвивалася вражаючими темпами, оскільки популярні потокові сервіси та дизайнерські фірми спостерігали підвищений попит на 3D-графіку VFX та 3D-моделі для дизайну продуктів.

## **Maya**

Maya часто використовується для потужної 3D-графіки і активно використовується в індустрії 3D. Завдяки різноманітним інструментам 3D-моделювання, анімації та візуалізації це програмне забезпечення для 3D-моделювання дозволяє створювати реалістичний і плавний 3D-дизайн.

Поряд із візуальним програмуванням Vifrost, це одне з найкращих програм для 3D-моделювання для попереднього моделювання волосся, частинок, тканини,

що значно полегшує процес анімації. Крім цього, його універсальність полягає в його здатності.

Це програмне забезпечення для моделювання дозволяє вам перемикатися між ЦП і ГП, дозволяючи створювати об'ємні та реалістичні 3D-світи та формувати 3D-об'єкти та сцени за допомогою інструментів моделювання в програмному забезпеченні Maya. Autodesk Maya працює на Windows, Linux і macOS.[19]

І хоча Maya дорога, Maya чудово підходить для стандартних відео та анімаційних фільмів. І з огляду на його побічний набір функцій, це має бути одним із ваших варіантів для 3D-дизайну.

Сильні сторони Maya:

- видатні функції для детального 3D моделювання, візуалізації, анімації та затінення
- простий у використанні для розробки складного 3D моделювання рідини
- використовується для моделювання та візуалізації 3D VFX у великих голлівудських фільмах
- здатний до найкращого такелажу та анімації

Слабкі сторони:

- досить дорого коштує
- іноді може виходити з ладу через велике навантаження
- займає значний обсяг пам'яті

### **Autodesk 3ds Max**

Коли потрібно обрати найкраще програмне забезпечення 3D для операційних систем Windows, Autodesk являється одним із найкращих варіантів. Хоча 3D-моделювання іноді може бути досить нюансованим, Autodesk має функції, які охоплюють основні області кіно, архітектури та навіть дизайну продуктів за допомогою 3D-моделювання.

Що робить Autodesk потужним професійним програмним забезпеченням для 3D-моделювання, так це той факт, що воно поставляється з вражаючими методами безпосереднього маніпулювання та моделювання, які поставляються разом із великою великою бібліотекою ресурсів 3D-дизайну.

Завдяки вражаючому набору функцій, Autodesk 3ds Max — це програмне забезпечення для 3D-моделювання, яке є одним із найкращих у галузі, хоча воно має дорогу ціну. Проте, якщо порівнювати з Maya (іншим програмним забезпеченням від Autodesk), з ним, безперечно, набагато легше працювати як новачкам, так і досвідченим дизайнерам.[20]

Сильні сторони Autodesk 3ds Max:

- видатна продуктивність, функції моделювання та візуалізації
- легкий і швидкий 3D інструмент
- відносно легше навчитися в порівнянні з багатьма програмами для 3D-моделювання на ринку
- постачається з вражаючими ефектами затінення та освітлення

Слабкі сторони:

- не працює на системах з слабкими характеристиками
- інтерфейс користувача можна покращити
- робота над 2D-проектами може бути громіздкою

## **ZBrush**

ZBrush є популярним вибором серед художників, які також хочуть друкувати 3D іграшки та фігурки за допомогою інструментів, спеціально призначених для цього.

ZBrush — найкращий додаток для ліплення та 3D-моделювання для створення органічних форм, а з останніми оновленнями він добре працює і з об'єктами з більш твердою поверхнею.



Функціональність ZBrush не обмежується лише ліпленням та моделюванням, оскільки її також можна використовувати для створення текстур та UV-карт. Тож для 3D-художників це означає, що реалістичні текстури і навіть тканини можуть бути відтворені, щоб додати більше гіперреалізму до звичайних 3D-моделей і скульптур.[21]

Сильні сторони ZBrush:

- одна з найкращих програм для 3D-ліплення
- добре працює, коли справа доходить до друку та моделювання
- чудово підходить для гіперреалістичного моделювання

Слабкі сторони:

- не найкраще програмне забезпечення для ригінгу та анімації
- досить дорого коштує

## **Blender**

Якщо ви шукаєте безкоштовне програмне забезпечення для 3D-моделювання, одним із найкращих варіантів – Blender. Безкоштовний застосунок для моделювання, текстурування, анімації та візуалізації. Ця програма з відкритим кодом існує з 1998 року, і за цей час вона придбала цілу купу художників, викладачів та ентузіастів, які стоять за її подальшим розвитком.

Blender може похвалитися вражаючим набором інструментів для 3D-моделювання та скульптури і вважається цілком життєздатною альтернативою платним програмам моделювання. Останніми роками вона стає все більш поширеною в планах великих студій. Колись Blender був відомий своїм нестандартним способом роботи, але багато з цих проблем було вирішено, тому він буде більш знайомим, якщо ви переходите з іншого додатка.[22]

Сильні сторони Blender:

- найкращий у категорії безкоштовних програмних забезпечень в галузі 3D-моделювання

- має широкий спектр функцій, що робить його придатним для різноманітних VFX та функцій моделювання
- має вражаючу колекцію доступних доповнень
- активна підтримка суспільства та розробників
- має велику кількість матеріалів для вивчення програми
- має відкритий код

Слабкі сторони:

- інтерфейс користувача може зробити процес навчання складним

### **Міхато**

Міхато — це рішення для 3D-анімації, яке може анімувати 3D-персонажів для фільмів та відеоігор. Міхато надає величезну колекцію високоякісних 3D-персонажів, а також сотні анімацій персонажів. Міхато гарантує, що персонажі готові до використання з автоматичною реалізацією людського скелета. Символи та анімації можна експортувати в різних форматах до Unity, Unreal Engine і Blender.

Міхато не являється повноцінним 3D-редактором, але він дуже зручний у плані реалізації скелету у персонажів та надання готових анімацій до них.[23]

### **Вибір програмного забезпечення для створення 3D-моделі**

Розглянуті приклади програмного забезпечення для створення 3D-моделей є одними з найпопулярніших на ринку 3D-моделювання. Вони існують у цій галузі дуже довго, обзавелися великою кількістю користувачів та розробників, які підтримують свої продукти і по сей день. Всі вони реалізують базові потреби для 3D-моделювання в тій чи іншій степені. У кожного з перелічених програмних забезпечень є свої сильні та слабкі сторони, починаючи від внутрішньої реалізації деякого функціоналу та закінчуючи моделлю поширення, або просте кажучи — ціною продукту. Але усі з них являються достойними варіантами, якщо потрібно розробити 3D-модель.

Для остаточного вирішення яке програмне забезпечення найкраще підходить для реалізації або редагування 3D-моделі, потрібно порівняти їх за деякими категоріями (див. Таблиця 3.2).

Таблиця 3.2 – Вибір програмного забезпечення для розробки 3D-моделі

<b>Програмне забезпечення для 3D-моделювання</b>	<b>Maya</b>	<b>Autocad 3ds Max</b>	<b>ZBrush</b>	<b>Blender</b>
<b>Ключові особливості</b>	Обробка захоплення руху, моделювання nurbs, шари	Мова скриптів, модифікатор редагування полі, сплайн-система	Настроюванні пензлики, шари ліплення	Модифікатор шкіни, комбінації клавіш, доповнення
<b>Найкраще підходить для</b>	Анімація та візуальні ефекти	Моделювання та риггінг	High-poly моделювання, ліплення	Анімації, незалежних проєктів
<b>Складність</b>	Середньо	Середньо	Середньо	Важко
<b>Ціна</b>	1470\$ кожен рік	1470\$ кожен рік	795\$ за ліцензію	Безкоштовно

На основі даних Таблиці 3.2, було прийнято рішення обрати основним програмним забезпеченням для розробки 3D-моделі Blender. Він являється потужним редактором, який підтримує усі сучасні потреби 3D-моделюванні, може працювати на відносно слабких за технічними характеристиками комп'ютерах, має велике суспільство користувачів та розробників, які поширюють учбові ресурси для опанування програми та одним з найважливіших критеріїв для вибору є те, що Blender має відкритий код та являється повністю безкоштовною програмою, на відміну від інших редакторів 3D-моделей, але ніяк не поступається їм у якості.

Додатковим програмним забезпеченням являється програма Mixamo для більш зручної та автоматичної роботи з анімаціями та скелетами персонажів. Програмне забезпечення від компанії Adobe також повністю безкоштовне та надає змогу реалізувати анімацію персонажів у дуже короткий час.

### 3.3 Технології машинного навчання

Основною технологією для машинного навчання виступає насамперед мова програмування та бібліотеки, які розраховані на роботу з штучним інтелектом та нейронними мережами.

Безпосереднім лідером у цих галузях по кількості орієнтованих бібліотек є мова програмування Python.

#### Python

Python — це потужна мова програмування, яку легко вивчати. Вона має ефективні високорівневі структури даних і простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованою природою роблять його ідеальною мовою для написання сценаріїв і швидкої розробки додатків у багатьох областях на більшості платформ.

Інтерпретатор Python і велика стандартна бібліотека вільно доступні у вихідній або двійковій формі для всіх основних платформ з веб-сайту Python і можуть вільно поширюватися. Той самий сайт також містить дистрибутиви та вказівники на багато безкоштовних модулів, програм та інструментів сторонніх розробників, а також додаткову документацію.[24]

Сильні сторони Python:

- зручний для початківців
- велика спільнота розробників та користувачів
- гнучкий і розширюваний
- обширні бібліотеки
- можливість вбудовування в інші програми
- висока масштабованість
- підтримка IoT
- портативний

Слабкі сторони:

- проблеми з дизайном
- повільніший, ніж скомпільовані мови
- проблема з безпекою
- робоче середовище
- високе споживання пам'яті
- динамічно-типізована мова
- складна багатопоточність
- збірщик сміття призводить до потенційної втрати пам'яті

Як і будь-яка інша мова програмування, Python також має безліч переваг і недоліків. Неможливо оголосити жодну мову програмування «один розмір для всіх».

Однак, якщо розглядати плюси Python, вони переважають його мінуси. Від розробника залежить, наскільки добре він проаналізує вимоги до проекту, перш ніж вибрати Python у якості основної мови програмування.

### **Бібліотеки Python**

Як було зазначено вище, Python має видатну кількість різноманітніших бібліотек. Деякі з цих бібліотек знадобляться у розробці програми для формування 3D-моделі. Потрібно використовувати не тільки ті бібліотеки, які спрощують роботу з машинним навчанням, але і інші бібліотеки, як, наприклад, стандартна бібліотека Python – OS, яка надає можливості взаємодії з системою.[25]

Бібліотеки, які знадобляться у розробці програми для формування 3D-моделі:

- 1) PyTorch
- 2) json
- 3) skimage
- 4) tqdm
- 5) OpenCV

- 6) trimesh
- 7) PyOpenGL
- 8) Freeglut
- 9) Ffmpeg

Розглянемо детальніше для чого призначені ці бібліотеки, які ми перерахували. 1) PyTorch — це оптимізована бібліотека тензорів глибокого навчання, заснована на Python і Torch, і в основному використовується для додатків, що використовують графічні і центральні процесори. 2) JSON означає JavaScript Object Notation. JSON — це легковаговий формат для зберігання та транспортування даних. 3) Scikit-image, або skimage, — це бібліотека Python з відкритим кодом, яка призначена для попередньої обробки зображень. 4) tqdm — це бібліотека Python, яка використовується для створення вимірювачів прогресу або індикаторів прогресу. 5) OpenCV (Open Source Computer Vision Library) — бібліотека комп'ютерного бачення та машинного навчання з відкритим вихідним кодом. 6) trimesh — це бібліотека та набір утиліт для читання, запису та маніпулювання тривимірними трикутними сітками, що зберігаються як .ply файли. 7) PyOpenGL — це стандартизована бібліотека, яка використовується як міст між Python і API OpenGL. Вона пропонує вбудовані зручні графічні та аудіобібліотеки. 8) Ffmpeg - це бібліотека, яка надає доступ до дуже швидкого конвертору відео та аудіо.

## **Google Colab**

Colaboratory, або скорочено «Colab», — це продукт від Google Research. Colab дозволяє будь-кому писати та виконувати довільний код Python через браузер, і особливо добре підходить для машинного навчання, аналізу даних та освіти. Більш технічно, Colab — це розміщена служба ноутбуків Jupyter, яка не вимагає налаштування для використання, а також надає безкоштовний доступ до обчислювальних ресурсів, включаючи графічні процесори. Google Colab буде використовуватися у розробці програмного рішення, тому що така складна

програма, яка формує 3D-модель за допомогою машинного навчання вимагає занадто багато ресурсів, які у виконавця бакалаврської кваліфікаційної роботи немає.[26]

### **Висновки до розділу 3**

У цьому розділі було розглянуто технології для розробки ігрового застосунку, формування 3D-моделі, машинного навчання та інші супутні рішення для виконання завдання бакалаврської кваліфікаційної роботи. Було проведено дослідження чотирьох найпопулярніших движків для розробки ігор, а саме Unity, Unreal Engine, Godot та GameMaker Studio 2. За допомогою порівняння сильних та слабких сторін кожного движка, було прийнято рішення обрати Unity у якості основного движку, тому що він має велику кількість якісних інструментів для роботи з 3D простором, безкоштовний та простий у використанні. Було порівняно найвідоміші програмні забезпечення для 3D-моделювання, а саме Maya, Autodesk 3ds Max, ZBrush та Blender. На основі порівняння їх сильних та слабких сторін обрано 3D-редактор Blender насамперед тому що це безкоштовний проект з відкритим кодом, коли інші редактори коштують значних грошей. Але не дивлячись на безкоштовну модель розповсюдження, Blender активно використовується у великих компаніях, має багато учбових ресурсів для швидкого опанування програмою та відносно небагато навантажує систему. Розглянуто технології для машинного навчання та додаткові рішення для розробки програмного забезпечення для формування 3D-моделі. Було вирішено обрати мову Python, яка є найкращою мовою програмування для швидкого та якісного написання програми, а бібліотеки, які спеціально розроблені для неї, надають усі потрібні можливості для якісної розробки потрібного застосунку. Також було обрано рішення від компанії Google – Colab, яка дозволяє безкоштовно завантажити в неї програмний код мови Python та використовувати її графічні процесори, оскільки ресурсів виконавця бакалаврської кваліфікаційної роботи не вистачить для точного результату.

## 4 СТРУКТУРА ТА РЕАЛІЗАЦІЯ ПРОГРАМИ ДЛЯ ФОРМУВАННЯ 3D МОДЕЛІ

### 4.1 Структура програми для формування 3D моделі

Метою цієї програми є досягнення високоякісної 3D-реконструкції одягнених людей з одного зображення з роздільною здатністю, достатньою для відновлення детальної інформації, такої як пальці, риси обличчя та складки одягу, що може бути проблематично через індивідуальні обмеження у ресурсах графічного процесору (див. Рисунок 4.1). Через це існуючі підходи не повною мірою використовують зображення людей з високою роздільною здатністю (наприклад, 1024 або більше), які зараз легко отримати за допомогою датчиків камер на мобільних телефонах. Це пов'язано з тим, що попередні підходи спираються на цілісне міркування для відображення між двовимірним виглядом зображуваної людини та її тривимірною формою, де на практиці використовуються зображення зі зменшеною вибіркою через надмірні вимоги до пам'яті. Хоча локальні латки зображень мають важливі ознаки для детальної тривимірної реконструкції, вони рідко використовуються на входах з повною високою роздільною здатністю через обмеження пам'яті поточного графічного обладнання. Підходи, спрямовані на подолання цього обмеження, можна розділити на дві групи. У першому методі проблема розкладається методом від грубого до тонкого, де високочастотні деталі тиснені поверх поверхонь низької точності. У цьому підході використовується низька роздільна здатність зображення для отримання грубої форми. Потім дрібні деталі, представлені як нормалі поверхні або зміщення, додаються за допомогою пост-процесу, такого як Shape From Shading, або композиції в нейронних мережах. У другому таборі використовуються високоточні моделі людей для «галюцинації» правдоподібних деталей. Хоча обидва підходи призводять до реконструкції, яка здається детальною, вони часто не достовірно відтворюють справжні деталі, які присутні на вхідних зображеннях.





Рисунок 4.1 – Приклад 3D-реконструкції з розподільною здатністю 128x128

У цій роботі представлено наскрізну багаторівневу структуру, яка визначає тривимірну геометрію одягнених людей з безпрецедентно високою роздільною здатністю зображення 1k з вирівнюванням пікселів, зберігаючи деталі в оригінальних вводах без будь-якої постобробки. Цей метод відрізняється від грубо-точного підходу тим, що на грубих рівнях не застосовується явне геометричне представлення. Натомість неявно закодований геометричний контекст поширюється на вищі рівні без передчасного визначення геометрії. Цей метод базується на нещодавно представленій технології Pixel-Aligned Implicit Function (PIFu) – Неявна функція, яка вирівнюється з пікселями [27]. Ця функція дозволяє нам плавно об'єднати вивчене цілісне вбудовування з грубих міркувань з характеристиками зображення, отриманими з введення високої роздільної здатності, принципово. Кожен рівень поступово включає додаткову інформацію, яка відсутня на грубих рівнях, при цьому остаточне визначення геометрії здійснюється лише на найвищому рівні.

Метод приймає зображення з роздільною здатністю  $512 \times 512$  як вхідні дані та отримує ембедінги ознак з низькою роздільною здатністю ( $128 \times 128$ ). Щоб отримати результати з більш високою роздільною здатністю, ми розміщуємо додатковий модуль передбачення, вирівняний із пікселями, поверх цього каркаса, де тонкий модуль приймає зображення з більшою роздільною здатністю ( $1024 \times 1024$ ) і кодує в особливості зображення з високою роздільною здатністю ( $512 \times 512$ ). Другий модуль використовує ембедінги ознак з високою роздільною здатністю, а також 3D-ембедінги з першого модуля для прогнозування поля ймовірності зайнятості. Для подальшого покращення якості та точності реконструкції спочатку прогнозується карти нормалей для передньої та зворотної сторін у просторі зображення та передаємо їх у мережу як додатковий вхід (див. Рисунок 4.2).

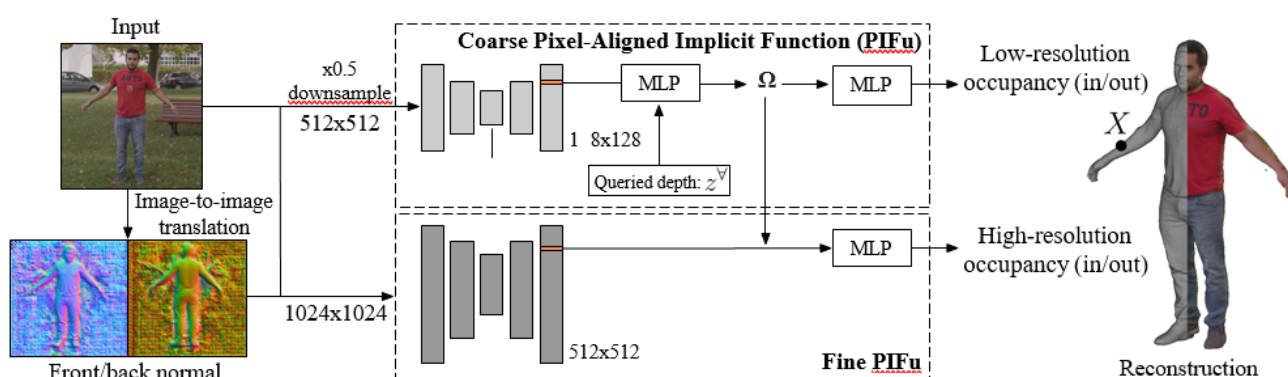


Рисунок 4.2 – Огляд методу побудови 3D-моделі

Коротко опишемо основу PIFu, яка становить грубий рівень нашого методу (верхня половина на Рисунку 4.2). Мета тривимірної оцифровки людини може бути досягнута шляхом оцінки зайнятості щільного тривимірного об'єму, яка визначає, чи знаходиться точка в 3D просторі всередині людського тіла чи ні. На відміну від попередніх підходів, де цільовий тривимірний простір дискретизовано, а алгоритми зосереджені на чіткій оцінці зайнятості кожного вокселя метою PIFu є моделювання функції  $f(X)$ , яка передбачає двійкове значення зайнятості для будь-якого даного 3D положення в безперервному просторі камери.

$$X = (X_x, X_y, X_z) \in R^3, \quad (4.1)$$

де  $X$  – двійкове значення зайнятості;

$X_x, X_y, X_z$  – координати точки в просторі;

$R^3$  – параметр залежності;

$$f(X, I) = \begin{cases} 1 & \text{якщо } X \text{ знаходиться у меші} \\ 0 & \text{в іншому випадку} \end{cases}, \quad (4.2)$$

де  $I$  – RGB зображення;

Оскільки під час навчання в пам'яті не зберігається явний 3D-об'єм, цей підхід являється ефективним, і, що важливіше, не потрібна дискретизація для цільового 3D-об'єму, що важливо для отримання високоякісної 3D-геометрії людського тіла. RIFu [27] моделює функцію  $f$  через архітектуру нейронної мережі, яка навчається наскрізним способом.

Зокрема, функція  $f$  спочатку витягує ембедінг ознак зображення із спроектованого двовимірного розташування в  $\pi(X) = x \in R^2$ , яке ми позначимо через  $\Phi(x, I)$ . Для  $\pi$  використовується ортогональна проекція, отже,  $x = \pi(X) = (X_x, X_y)$ . Потім він оцінює зайнятість запиту 3D точки  $X$ , і таким чином:

$$f(X, I) = g(\Phi(x, I), Z), \quad (4.3)$$

де  $Z = X_z$  — глибина уздовж променя, визначена двовимірною проекцією  $x$ ;

Зауважте, що всі тривимірні точки вздовж одного променя мають точно однакові ознаки зображення  $\Phi(x, I)$  з одного і того ж прогнозованого місця  $x$ , і, таким чином, функція  $g$  повинна зосередитися на змінній вхідній глибині  $Z$ , щоб усунути неоднозначність зайнятості тривимірних точок уздовж проміння. Архітектура згорткової нейронної мережі (CNN) використовується для функції вбудовування 2D ознак  $\Phi$  і багатошаровий перцептрон (MLP) для функції  $g$ .

Великомасштабний набір даних, синтетично згенерований шляхом відтворення сотень високоякісних сканованих 3D-моделей людського мешу, використовується для наскрізного навчання функції  $f$ . На відміну від методів на основі вокселів, RIFu не створює дискретизований об'єм як вихід, тому навчання

можна виконувати шляхом вибірки 3D точок і обчислення втрати заповнюваності в вибіркових місцях, без створення тривимірних сіток. Під час висновку 3D простір рівномірно відбирається, щоб зробити висновок про зайнятість, і остаточно ізоповерхня витягується з порогом 0,5 за допомогою маршових кубів. Оскільки під час навчання в пам'яті не зберігається явний 3D-об'єм, цей підхід ефективний у пам'яті, і, що ще важливіше, не потрібна дискретизація для цільового 3D-об'єму, що важливо для отримання високоякісної 3D-геометрії для цільових людей. PIFu моделює функцію  $f$  через архітектуру нейронної мережі, яка навчається наскрізним способом.

Як було зазначено, метод складається з двох рівнів модулів PIFu: грубий рівень, подібний до PIFu, зосереджений на інтеграції глобальної геометричної інформації шляхом входу зменшеного зображення розміром  $512 \times 512$  і створення основних функцій зображення  $128 \times 128$ , і тонкий рівень, який зосереджується на додаванні більш тонких деталей, беручи вихідне зображення з роздільною здатністю  $1024 \times 1024$  як вхід, і створюючи основні функції зображення з роздільною здатністю  $512 \times 512$ . Модуль тонкого рівня використовує ембедінг ознак 3D, витягнуті з грубого рівня замість значення абсолютної глибини. Модуль грубого рівня також приймає передбачені карти нормалей фронтального та зворотного боку:

$$f^L(X) = g^L(\Phi^L(x_L, I_L, F_L, B_L), Z), \quad (4.3)$$

де  $I_L$  – це вхідне значення з нижчою роздільною здатністю;

$F_L, B_L$  – прогнозовані карти нормалей з однаковою роздільною здатністю;

$x_L \in R^2$  – проекція 2D розташування  $X$  у просторі зображень  $I_L$ ;

Тонкий рівень позначається як:

$$f^H(X) = g^H(\Phi^H(x_H, I_H, F_H, B_H), \Omega(X)), \quad (4.4)$$

де  $I_H, F_H, B_H$  – це вхідне зображення, фронтальна карта нормалей та карта нормалей заднього боку відповідно з роздільною здатністю  $1024 \times 1024$ .

$x_H \in R^2$  є місцем 2D проекції з високою роздільною здатністю, отже, у нашому випадку  $x_H = 2x_L$ . Функція  $\Phi^H$  кодує ознаки зображення з входу з високою

роздільною здатністю і має структуру, подібну до витягувача функцій з низькою роздільною здатністю  $\Phi^L$ . Ключова відмінність полягає в тому, що сприйнятливим полем  $\Phi^H$  не охоплює все зображення, але завдяки його повністю згортковій архітектурі мережу можна навчати за допомогою випадкового розсувного вікна і робити висновки з вихідною роздільною здатністю зображення (тобто 1024x1024). Нарешті,  $\Omega(X)$  — це тривимірний ембедінг, витягнутий з мережі грубого рівня, де ми беремо вихідні характеристики з проміжного шару  $g^L$ . Оскільки тонкий рівень бере ці функції з першого вирівняного за пікселем MLP як 3D-ембедінг, глобальна якість реконструкції не повинна погіршуватися, а має покращуватися. Якщо проект мережі може належним чином використовувати підвищену роздільну здатність зображення та пропускну здатність мережі. Крім того, тонкій мережі не потрібно обробляти нормалізацію (тобто створювати глобально узгоджену тривимірну глибину) і, отже, не потрібно бачити все зображення, що дозволяє нам тренувати його за допомогою кадрування зображень. Це важливо, щоб дозволити вводити зображення з високою роздільною здатністю, не обмежуючись пам'яттю.

## 4.2 Реалізація програми для формування 3D моделі

Для реалізації програми для формування 3D моделі застосуємо технологію PIFU. У файлі `simple_test.py` ми передаємо потрібні аргументи до парсеру, а саме шлях до папки, звідки програма буде брати зображення з розширенням `.jpg` або `.png`, шлях до папки, де згенерується результат роботи програми у вигляді файлу для 3D моделей з розширенням `.obj` та зображення мещу з передньої та задньої сторони, аргумент роздільної здатності зображення для того щоб повідомити програму якої якості потрібно сформувати 3D модель, шлях до папки з чекпоїнтами та аргумент чи потрібно нам обрізати наше зображення та викликаємо файл функцію файлу `reson – resonWrapper` (див. Рисунок 4.3).

```

apps > simple_test.py > ...
1  from .recon import reconWrapper
2  import argparse
3
4  #Налаштування
5  parser = argparse.ArgumentParser()
6  parser.add_argument('-i', '--input_path', type=str, default='./sample_images')
7  parser.add_argument('-o', '--out_path', type=str, default='./results')
8  parser.add_argument('-c', '--ckpt_path', type=str, default='./checkpoints/checkpoint.pt')
9  parser.add_argument('-r', '--resolution', type=int, default=512)
10 parser.add_argument('--use_rect', action='store_true', help='use rectangle for cropping')
11 args = parser.parse_args()
12
13 resolution = str(args.resolution)
14
15 start_id = -1
16 end_id = -1
17 cmd = ['--dataroot', args.input_path, '--results_path', args.out_path, \
18       '--loadSize', '1024', '--resolution', resolution, '--load_netMR_checkpoint_path', \
19       args.ckpt_path, \
20       '--start_id', '%d' % start_id, '--end_id', '%d' % end_id]
21 reconWrapper(cmd, args.use_rect)

```

Рисунок 4.3 – Лістинг коду з файлу simple\_test.py

У файлі recon.py знаходиться функція reconWrapper, яка парсить аргументи та передає їх функції recon – основна функція для реконструкції 3D моделі. У цієї функції ми використовуємо програмно-апаратну архітектуру паралельних обчислень Cuda, що дозволяє суттєво збільшити обчислювальну продуктивність завдяки використанню графічних процесорів фірми Nvidia та бібліотеку PyTorch — фреймворк машинного навчання для мови Python з відкритим вихідним кодом, створений на базі Torch. За допомогою них ми реалізуємо дві нейронні мережі, одну для формування загальної геометрії 3D моделі, а другу для покращення вихідного результату. Змінні називаються netMR та netG відповідно. Також ми викликаємо функцію gen\_mesh яка за допомогою нейронної мережі netMR, яку ми визначили до цього, генерує Полігональну сітку або меш – це сукупність вершин, ребер та граней, які визначають форму багатогранного об'єкта в тривимірній комп'ютерній графіці та об'ємному моделюванні (див. Рисунок 4.4).

```

def recon(opt, use_rect=False):
    state_dict_path = None
    if opt.load_netMR_checkpoint_path is not None:
        state_dict_path = opt.load_netMR_checkpoint_path
    elif opt.resume_epoch < 0:
        state_dict_path = '%s/%s_train_latest' % (opt.checkpoints_path, opt.name)
        opt.resume_epoch = 0
    else:
        state_dict_path = '%s/%s_train_epoch_%d' % (opt.checkpoints_path, opt.name, opt.resume_epoch)

    start_id = opt.start_id
    end_id = opt.end_id

    cuda = torch.device('cuda:%d' % opt.gpu_id if torch.cuda.is_available() else 'cpu')

    state_dict = None
    if state_dict_path is not None and os.path.exists(state_dict_path):
        print('Resuming from ', state_dict_path)
        state_dict = torch.load(state_dict_path, map_location=cuda)
        print('Warning: opt is overwritten.')
        dataroot = opt.dataroot
        resolution = opt.resolution
        results_path = opt.results_path
        loadSize = opt.loadSize

        opt = state_dict['opt']
        opt.dataroot = dataroot
        opt.resolution = resolution
        opt.results_path = results_path
        opt.loadSize = loadSize
    else:
        raise Exception('failed loading state dict!', state_dict_path)

    if use_rect:
        test_dataset = EvalDataset(opt)
    else:
        test_dataset = EvalWPoseDataset(opt)

    print('test data size: ', len(test_dataset))
    projection_mode = test_dataset.projection_mode

    opt_netG = state_dict['opt_netG']
    netG = HGPIFuNetwMML(opt_netG, projection_mode).to(device=cuda)
    netMR = HGPIFuMRNet(opt, netG, projection_mode).to(device=cuda)

```

Рисунок 4.4 – Функція recon для реконструкції моделі

Розглянемо функцію `gen_mesh`. Ця функція формує полігональну сітку на основі 2D зображення, яке ми передали у файлі «`simple_test.py`» у вигляді аргументу. Вона за допомогою нейронної мережі передбачає чи окрема 3D позиція відноситься до полігональної сітки або ні. Це допомагає побудувати цілісність полігональної сітки та уникнути артефактів, щоб окремі вершини та ребра «не висіли у повітрі» та були не скріплені до основного мешу (див. Рисунок 4.4).

```

def gen_mesh(res, net, cuda, data, save_path, thresh=0.5, use_octree=True, components=False):
    image_tensor_global = data['img_512'].to(device=cuda)
    image_tensor = data['img'].to(device=cuda)
    calib_tensor = data['calib'].to(device=cuda)

    net.filter_global(image_tensor_global)
    net.filter_local(image_tensor[:,None])

    try:
        if net.netG.netF is not None:
            image_tensor_global = torch.cat([image_tensor_global, net.netG.nm1F], 0)
        if net.netG.netB is not None:
            image_tensor_global = torch.cat([image_tensor_global, net.netG.nm1B], 0)
    except:
        pass

    b_min = data['b_min']
    b_max = data['b_max']
    try:
        save_img_path = save_path[:-4] + '.png'
        save_img_list = []
        for v in range(image_tensor_global.shape[0]):
            save_img = (np.transpose(image_tensor_global[v].detach().cpu().numpy(), (1, 2, 0)) * 0.5 + 0.5)[:, :, ::-1] * 255.0
            save_img_list.append(save_img)
        save_img = np.concatenate(save_img_list, axis=1)
        cv2.imwrite(save_img_path, save_img)

        verts, faces, _, _ = reconstruction(
            net, cuda, calib_tensor, res, b_min, b_max, thresh, use_octree=use_octree, num_samples=50000)
        verts_tensor = torch.from_numpy(verts.T).unsqueeze(0).to(device=cuda).float()

        color = np.zeros(verts.shape)
        interval = 50000
        for i in range(len(color) // interval + 1):
            left = i * interval
            if i == len(color) // interval:
                right = -1
            else:
                right = (i + 1) * interval
            net.calc_normal(verts_tensor[:, None, :, left:right], calib_tensor[:,None], calib_tensor)
            nml = net.nmls.detach().cpu().numpy()[0] * 0.5 + 0.5
            color[left:right] = nml.T

        save_obj_mesh_with_color(save_path, verts, faces, color)
    except Exception as e:
        print(e)
    torch.cuda.empty_cache()

```

Рисунок 4.5 – Функція gen\_mesh для генерування мешу

Для візуалізації результату у файлі «render\_turntable.py» ми використовуємо функцію make\_rotate, яка на основі 180 скріншотів з різного ракурсу формує «стенд», на якому створена 3D модель обертається навколо своєї осі. Це зручно, тому що дає можливість подивитись як програма відпрацювала не тільки на передній частині, але й на інших та які артефакти виникли під час формування 3D моделі.



```

import numpy as np
import sys
import os

sys.path.insert(0, os.path.abspath(os.path.join(os.path.dirname(__file__), '..')))
ROOT_PATH = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

from lib.render.mesh import load_obj_mesh, compute_normal
from lib.render.camera import Camera
from lib.render.gl.geo_render import GeoRender
from lib.render.gl.color_render import ColorRender
import trimesh

import cv2
import os
import argparse

width = 512
height = 512

def make_rotate(rx, ry, rz):

    sinX = np.sin(rx)
    sinY = np.sin(ry)
    sinZ = np.sin(rz)

    cosX = np.cos(rx)
    cosY = np.cos(ry)
    cosZ = np.cos(rz)

    Rx = np.zeros((3,3))
    Rx[0, 0] = 1.0
    Rx[1, 1] = cosX
    Rx[1, 2] = -sinX
    Rx[2, 1] = sinX
    Rx[2, 2] = cosX

    Ry = np.zeros((3,3))
    Ry[0, 0] = cosY
    Ry[0, 2] = sinY
    Ry[1, 1] = 1.0
    Ry[2, 0] = -sinY
    Ry[2, 2] = cosY

    Rz = np.zeros((3,3))
    Rz[0, 0] = cosZ
    Rz[0, 1] = -sinZ
    Rz[1, 0] = sinZ
    Rz[1, 1] = cosZ
    Rz[2, 2] = 1.0

    R = np.matmul(np.matmul(Rz,Ry),Rx)
    return R

```

Рисунок 4.6 – Функція make\_rotate для візуалізації результату

Також для формування більш точної 3D моделі, програму було завантажено у сервіс Google Colab, щоб застосувати їх графічні можливості.

Графічних можливостей безкоштовної версії Google Colab вистачило на генерацію моделі роздільною здатністю 256x256. Результат знаходиться в папці «results». В моделі присутні деякі помилки генерації, але їх можливо виправити за допомогою 3D редактору Blender. Результат візуалізовано за допомогою функції файлу «render\_turntable.py» make\_rotate.



Рисунок 4.7 – Візуалізований результат роботи програми

#### 4.3 Підготовка моделі та її імпорт в ігровий застосунок

Для підготовки моделі до імпорту в ігровий застосунок, потрібно видалити артефакти, які з'явилися у процесі генерації моделі та додати їй текстуру. Як можна побачити на рисунку 4.8, модель має артефакти в області шиї та під лівою пахвою. За допомогою інструментів маніпулювання з вершинами, ребрами та гранями, та пензликами для моделей можливо швидко прибрати ці помилки реконструкції.

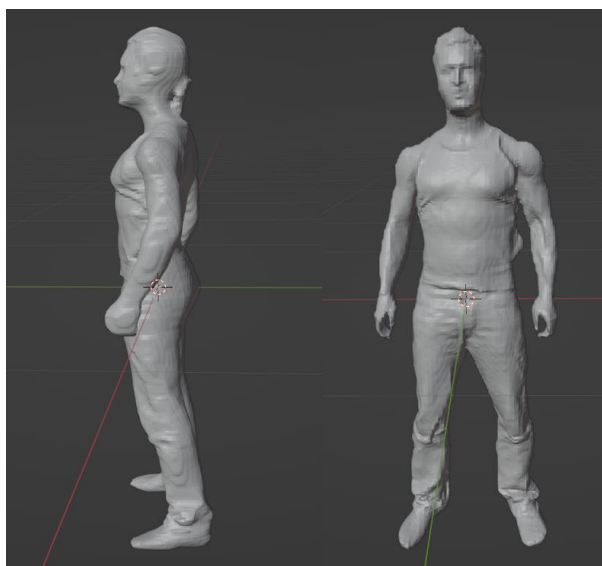


Рисунок 4.8 – Модель до прибрання артефактів

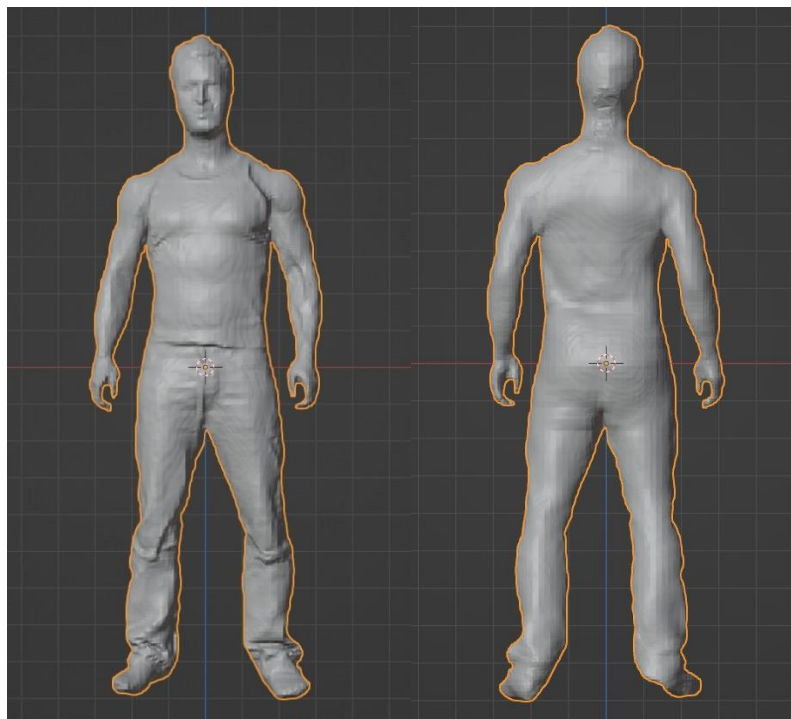


Рисунок 4.9 – Модель після прибрання артефактів

Тепер потрібно додати до моделі текстуру. За допомогою вікна UV Editing ми можемо додати 2D зображення, яке використовувалось в якості основи для реконструкції моделі, у вигляді текстури. Як ми бачимо на рисунку 4.10, текстура натягнулась з обох боків. Це можна виправити за допомогою вибору зони для імпорту на текстурі.

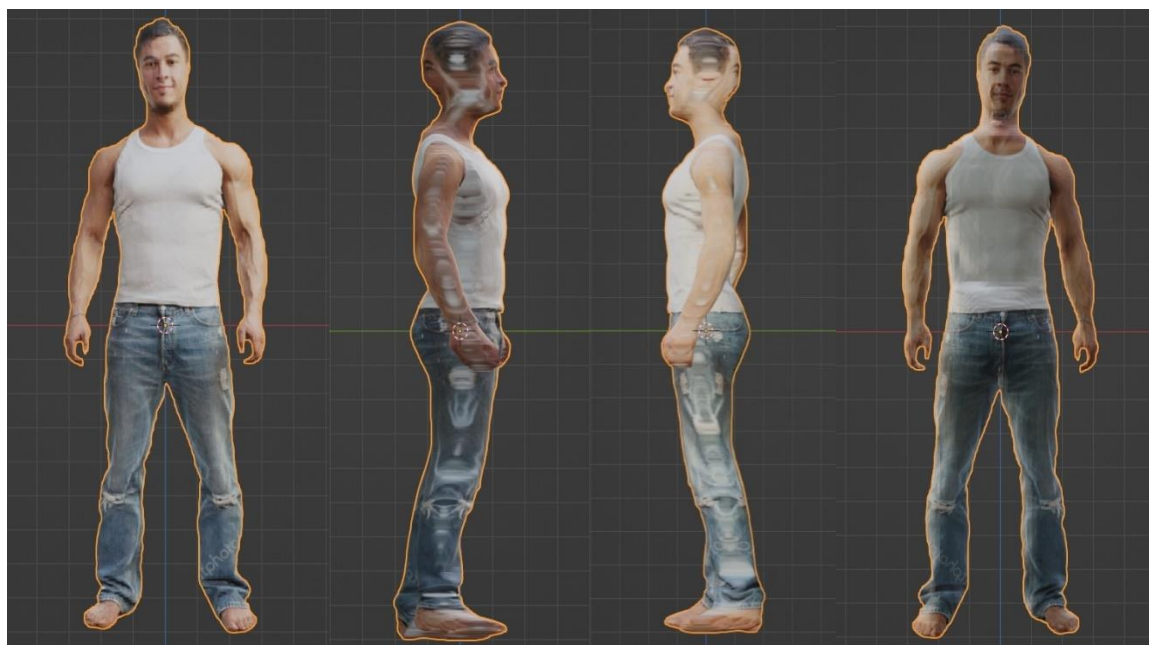


Рисунок 4.10 – Модель з текстурою до виправлення



Рисунок 4.11 – Модель з текстурою після виправлення

Тепер за допомогою Adobe Mixamo надамо моделі скелет для анімації. Це можливо створити і в редакторі Blender, але Mixamo автоматично та швидко налаштовує скелет для персонажів. Від користувача потрібно тільки розтавити відповідні маркери на зап'ясті, лікті, коліна, підборіддя та пах.

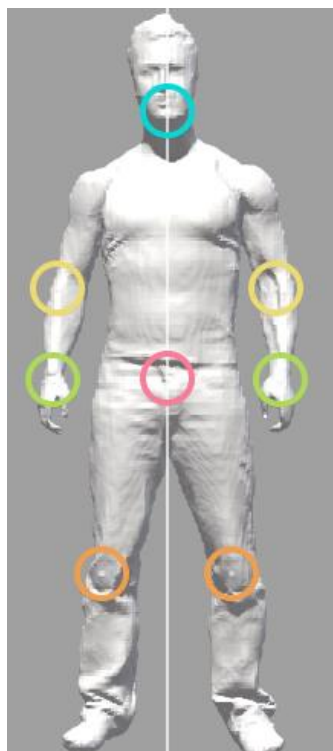


Рисунок 4.12 – Модель з розставленими маркерами

Після того, як спрацювала функція Auto-Rigger для автоматичного додання скелету до персонажа, можна імпортувати його у ігровий додаток.



Рисунок 4.13 – Результат додання скелету та анімацій.

Так як з створенної моделі за допомогою редактору було прибрано всі артефакти, натягнуто текстуру, розроблено скелет та анімації, то це означає, що вона повністю готова до імпорту в ігровий застосунок.

За допомогою ігрового движка Unity було розроблено примітивну 3D гру для демонстрації застосування створенної моделі. Було реалізовано ходьбу, біг, стрибок та інші супутні анімації персонажу. Ігровий застосунок складається з вікон меню, менеджера аватара та тестового рівня. Реалізовано примітивний геймплей у вигляді збору монеток та відображені їх кількості на екрані.

При запуску ігрового застосунку, перше, що бачить користувач – це головне меню з кнопками «Start Game», «Settings», «Exit» (див. Рисунок 4.14). Перша кнопка переводить користувача в вікно менеджера аватару. Друга кнопка активує налаштування ігрового застосунку. У цьому вікні він може обрати налаштування графіки. (див. Рисунок 4.15). Третя кнопка активує вихід з гри.



Рисунок 4.14 – Головне меню ігрового застосунку

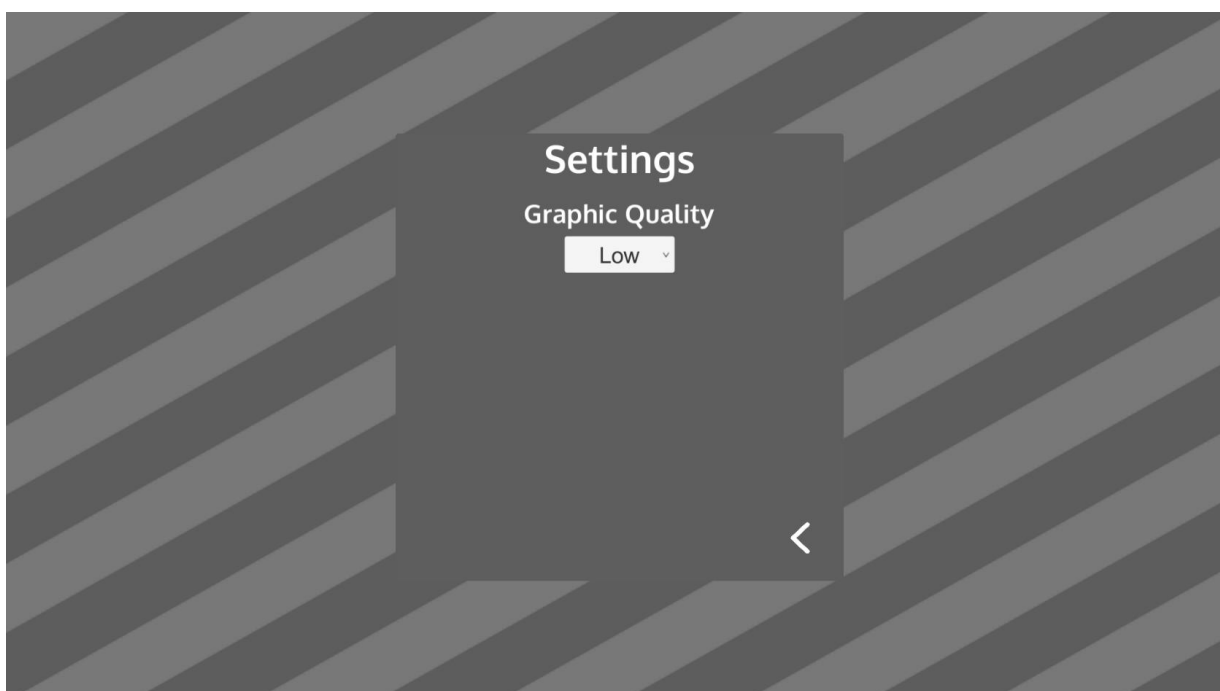


Рисунок 4.15 – Налаштування ігрового додатку

Відкривши вікно менеджера аватару (див. Рисунок 4.16), користувач бачить кнопку «Import» з можливістю вибору файлу зображення для відображення в області нижче кнопки. Це концепція як би міг виглядати ігровий додаток, якщо користувач міг би завантажувати свою модель та грати за неї, але, нажаль, наразі ігрові движки такий функціонал не підтримують, тому завантаження зображення нінащо не впливає (див. Рисунок 4.17). Після імпорту зображення з'являється кнопка «Play» для запуску тестового рівня гри (див. Рисунок 4.18).

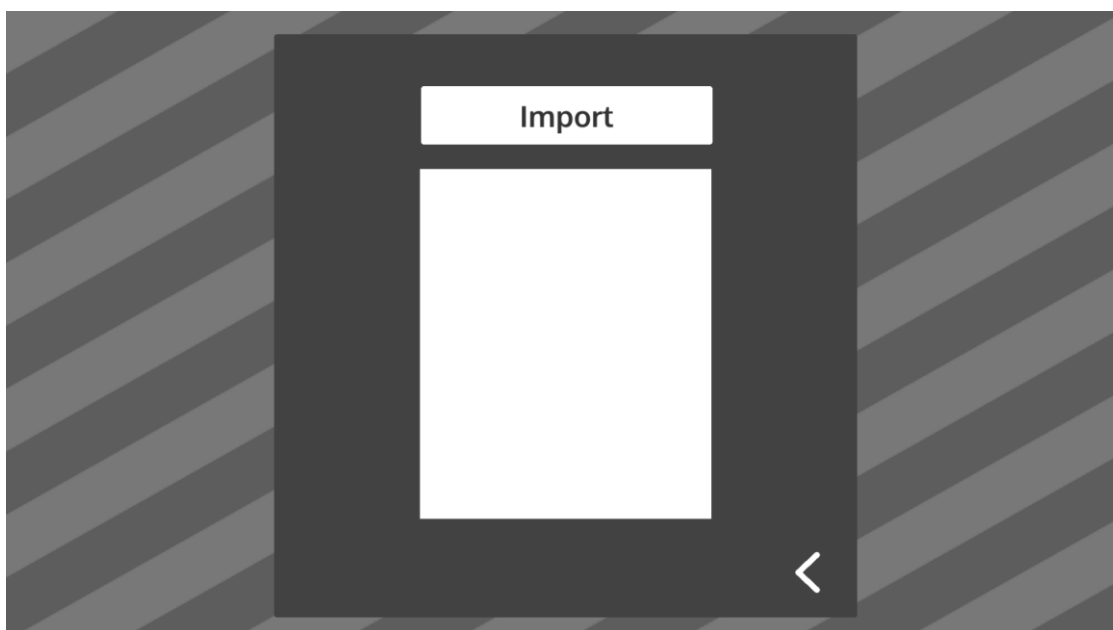


Рисунок 4.16 – Вікно менеджера аватару

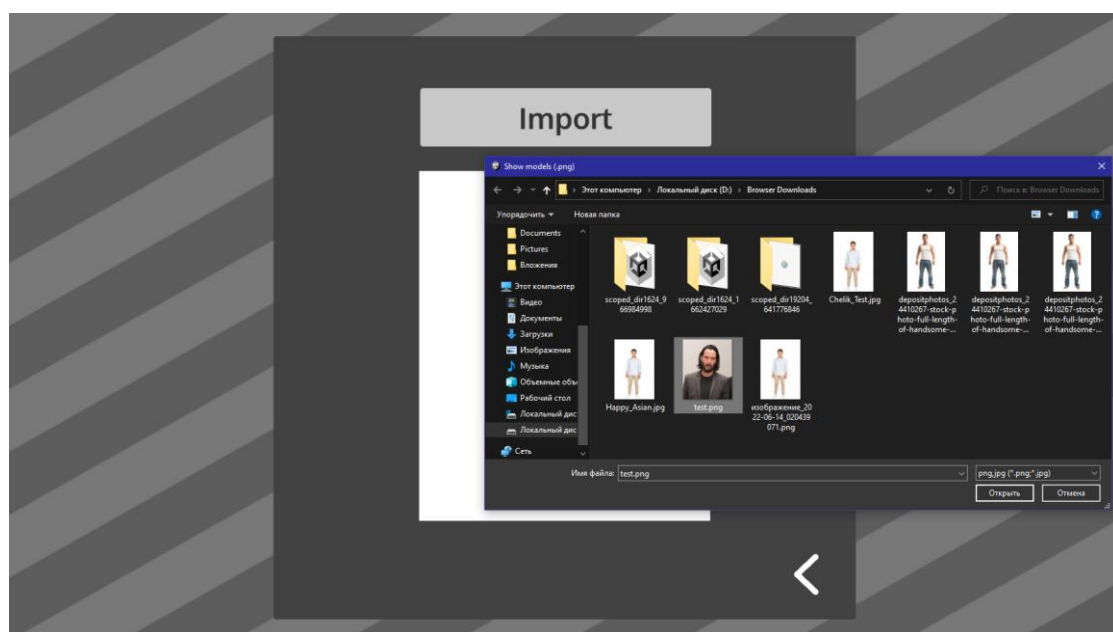


Рисунок 4.17 – Завантаження користувацького зображення

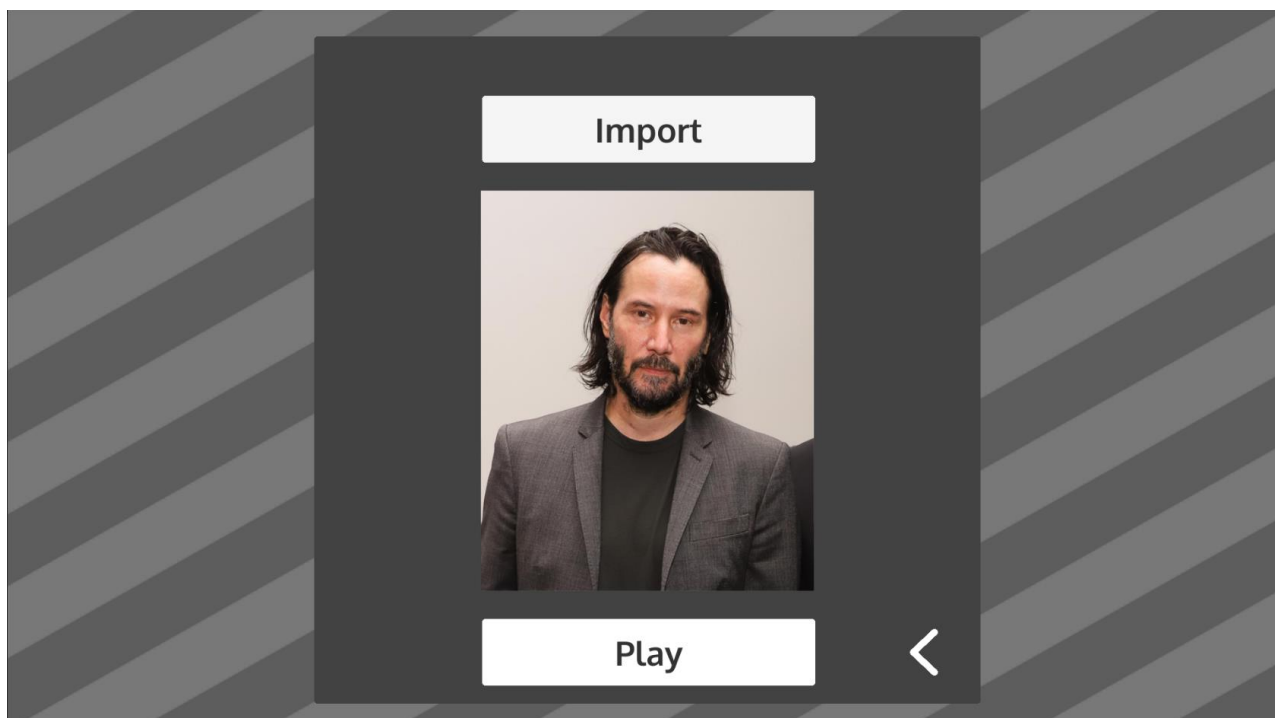


Рисунок 4.18 – Імпорт зображення виконано успішно

На тестовому рівні існує персонаж гравця та кімната з платформами і монетами (див. Рисунок 4.19). Персонаж здатний ходити, бігати, пригати, має анімацію простою та падіння та може почати танцювати, що активується окремою клавішею «G». Геймплей складається в збиранні монеток. (див. Рисунок 4.20).

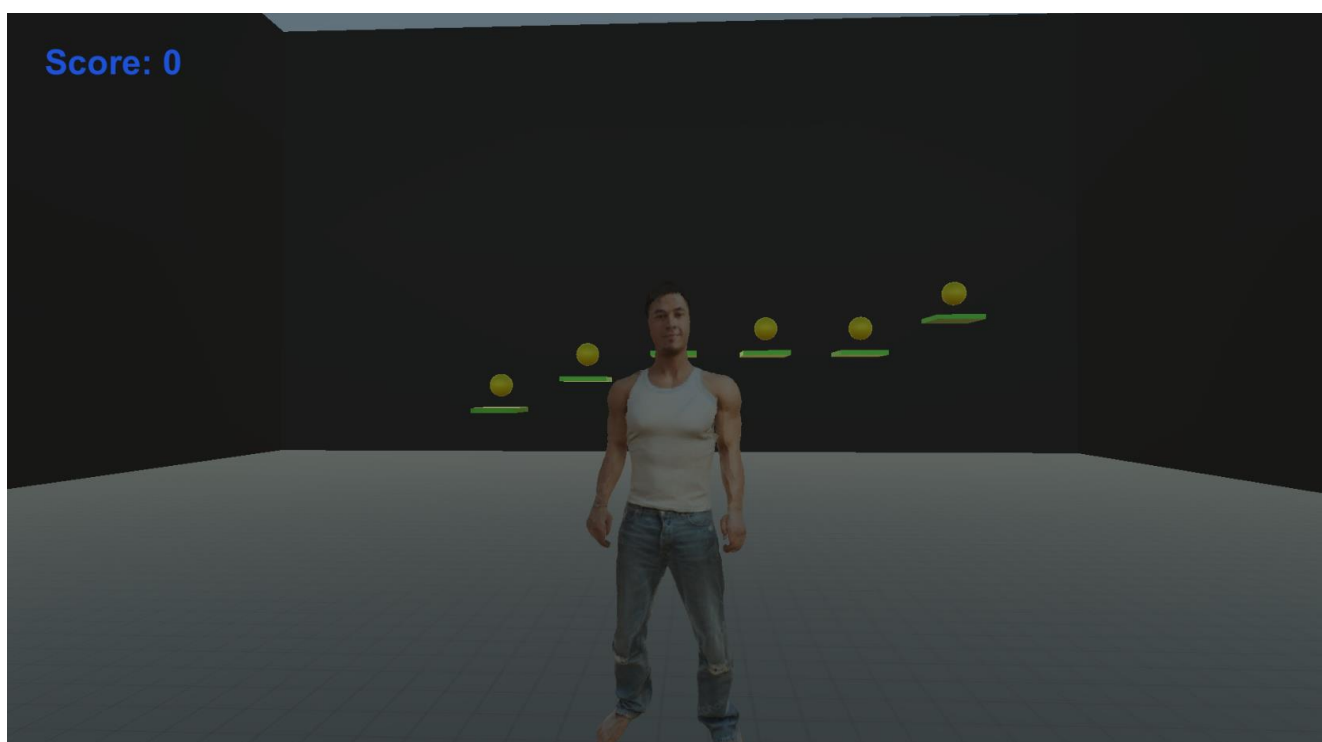


Рисунок 4.19 – Зовнішній вигляд тестового рівня та персонажу у грі



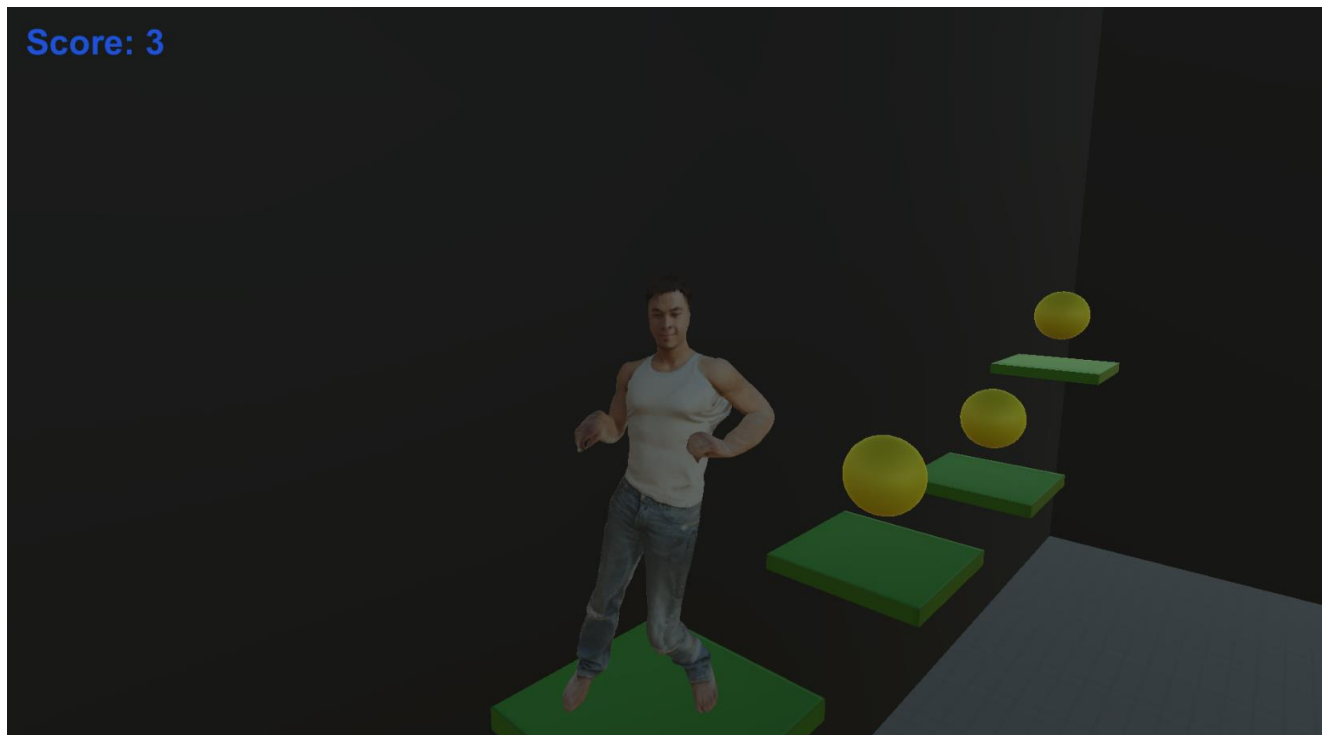


Рисунок 4.20 – Збирання монеток та танець персонажу.

Хоча ігровий додаток має основні риси кожної відеогри, а саме ігрове меню, налаштування, рівень та геймплей, це лише демонстрація підтримки ігровим застосунком імпорту згенерованого персонажу.

**Спеціальний розділ**  
**ОХОРОНА ПРАЦІ**  
**до кваліфікаційної роботи**

на тему:

**«Формування 3D-моделі для ігрового застосунку  
засобами штучного інтелекту»**

Спеціальність 122 «Комп'ютерні науки»

**122 – БКР – 401.21810125**

*Виконав студент 4-го курсу, групи 401*

\_\_\_\_\_ *Чепара І.В.* \_\_\_\_\_

*(підпис, ініціали та прізвище)*

«\_\_\_» \_\_\_\_\_ 2022 р.

*Консультант \_\_\_\_\_ ст. викладач \_\_\_\_\_*

*(наук. ступінь, вчене звання)*

\_\_\_\_\_ *Макарова О.В.* \_\_\_\_\_

*(підпис, ініціали та прізвище)*

«\_\_\_» \_\_\_\_\_ 2022р.

**Миколаїв – 2022**

## 5 ОХОРОНА ПРАЦІ

Метою даного розділу є дослідження створення безпечних і здорових умов праці на робочому місці або у виробничому приміщенні працівникам пов'язаних з інформаційними технологіями.

Перед суспільством на усіх етапах його розвитку стояло завдання підвищення ефективності трудового процесу, удосконалення методів виробництва продуктів та засобів, які необхідні для його існування. У своїй практичній діяльності фахівець будь-якої галузі має вміти аналізувати потенційні небезпеки на різних етапах проектування, експлуатації устаткування, визначати заходи усунення небезпек упроваджувати вимоги нормативних документів щодо нових розробок.

Забезпечення безпеки праці – необхідна умова будь-якого технологічного процесу, зокрема з застосування комп'ютерної техніки. Із зростанням кількості автоматизованих виробництв і комп'ютерних систем зазнають зміни чинники, які впливають на формування умов праці, які у свою чергу призводять до негативної дії на фізіологічну систему людини, знижуючи її працездатність і ініціюючи неправильні дії. Відомо, що кількість нещасних випадків, викликаних небезпечними діями, значно більше, ніж кількість нещасних випадків, викликаних небезпечними умовами праці.

На сьогодні, враховуючи інтенсивний розвиток комп'ютерної техніки, різко зросло і далі зростатиме кількість галузей і сфер діяльності людини, у яких використовуються інформаційні технології. Сучасний світ комп'ютерної техніки настільки широкий і різноманітний і так швидко розвивається, що кожна молода людина, яка сьогодні вчиться в середньому або вищому навчальному закладі, поза усяких сумнівів знайде в ньому своє місце. Тому надзвичайно важливо, щоб на початку своєї діяльності студент набув відповідних навичок зі збереження власного і суспільного здоров'я. [28]

«Комп'ютерні захворювання», як і галузь, яка породила їх появу, ще достатньо молода, але на відміну від самої техніки ще мало вивчені. Значною мірою

поява «комп'ютерних захворювань» зумовлена тим, що шкідлива дія такого популярного і, на перший погляд, безпечного сучасного способу праці, яким є комп'ютер, ще далеко не всім відома. Крім того, на користувачів комп'ютерами впливає цілий комплекс чинників малої інтенсивності, негативна дія яких розвивається поступово і приховано. Тому захворювання виявляються тільки після багатьох місяців або навіть років праці, коли боротися з ними вже у край важко. У багатьох випадках важко навіть установити головну причину захворювань.

У таких умовах все більше значущим і необхідним стає потреба формування у майбутніх фахівців відповідних знань щодо особливостей впливу несприятливих виробничих чинників на комп'ютеризованих робочих місцях, способів і засобів, направлених на мінімізацію такого впливу, збереження здоров'я і професійного довголіття користувачів.

Отже, охорона праці сприяє систематизації знань з безпечних методів праці щодо напряму інформаційних технологій та створенню комплексного підходу до організації робочих місць користувачів інформаційних технологій, що відповідають сучасним вимогам безпеки.

### **5.1 Міжнародні та державні норми з питань безпеки праці користувачів інформаційних технологій**

Вивчаючи цю тему, необхідно пам'ятати, що державна політика України в галузі безпеки праці спрямована на створення безпечних і здорових умов праці, запобігання нещасним випадкам та професійним захворюванням. Вона базується на ряді принципів, основними з яких є пріоритет життя і здоров'я працівників, повна відповідальність роботодавця за створення безпечних та належних умов праці, підвищення рівня промислової безпеки, комплексне розв'язання завдань з охорони праці, соціальний захист працівників, повне відшкодування шкоди особам, які потерпіли від нещасних випадків на виробництві та професійних захворювань.

Україна є членом Міжнародної організації праці (МОП). Вона ратифікувала 63 конвенції МОП, із них 14 – за роки незалежності. Положення цих конвенцій лягли в основу чинного в Україні законодавства, що регулює соціально-трудові відносини.

Законодавство Євросоюзу у сфері охорони праці можна умовно розділити на дві групи:

- директиви ЄС щодо захисту працівників;
- директиви ЄС щодо випуску товарів на ринок (включаючи обладнання, устаткування, машини, засоби колективного та індивідуального захисту, які використовують працівники на робочому місці). [29]

Необхідно вивчити міжнародну систему охорони безпеки та здоров'я, основою якої є міжнародний стандарт OHSAS 18001 «Система менеджменту охорони здоров'я та безпеки персоналу» будується на принципі добровільного дотримання вимог охорони праці, виходячи із встановленого законодавством допустимого рівня ризику для життя і здоров'я працівників. Міжнародний стандарт OHSAS так само, як і система управління якістю ISO 9000 і система управління охороною навколишнього середовища ISO 14000, побудований на основі циклу Демінга. Вивчаючи цю тему, треба знати вимоги, що встановлює підхід OHSAS.

Вимоги про те, якою має бути ергономіка в країнах ЄС, закріплено в законодавстві Союзу й, насамперед, у директиві № 89/391 «Про заходи щодо поліпшення безпеки й здоров'я трудящих». Відповідно до неї роботодавці зобов'язані оцінювати виробничі ризики й забезпечувати адекватні захисні і профілактичні заходи, гарантувати відповідне навчання й інструктаж працівників з дотримання заходів безпеки, а також надавати працівникам інформацію та консультації і дозволяти їм брати участь в обговоренні всіх питань із забезпечення безпеки й гігієни праці. [30]

Розробкою загальних єдиних нормативних документів для користувачів інформаційних технологій займаються декілька міжнародних організацій:

- International Organization for Standardization (ISO) –

міжнародна організація зі стандартизації;

- Ergonomics committee ISO (TC 159) – комітет з ергономіки міжнародної організації зі стандартизації;
- European Standardization Organization (CEN) – європейська організація зі стандартизації.

Серед низки розроблених нормативних документів з використання ВДТ найбільш часто застосовуються такі стандарти:

- ISO 9241-3, який регламентує ергономічні вимоги за умовами праці і охорони здоров'я користувачів;
- ISO 9001, який визначає якість і рівень виробництва апаратури;
- ISO DIN 9995, який встановлює принципи розміщення елементів клавіатури для роботи з текстом в офісних системах;
- IEC 950, який визначає норми безпеки електротехнічного устаткування.

Треба мати на увазі, що основним нормативним документом України із забезпечення охорони праці користувачів комп'ютерно-інформаційних технологій є НПАОП 0.00-1.28-10 «Правила охорони праці при експлуатації електронно-обчислювальних машин».

## **5.2 Виробничі фактори, які можуть спричинити професійні захворювання в ІТ-індустрії, та методи й засоби їх усунення**

У процесі трудової діяльності на людину впливає комплекс несприятливих чинників виробничого середовища і трудового процесу, у результаті чого можливі виробничі травми і професійні захворювання. Комплексний характер впливу цих чинників визначає необхідність системного підходу під час вирішення питань безпеки праці. [31] Усі етапи технологічного процесу характеризуються інформаційним навантаженням, їх виконання вимагає уваги, зосередженості, що необхідно враховувати під час створення безпечних і нешкідливих умов праці користувачів ІТ.

Необхідно розуміти, що комп'ютер як технічний засіб може бути джерелом небезпечних і шкідливих виробничих чинників. З принципу роботи відеодисплейного терміналу виходить, що він є джерелом:

- для моніторів з електронно-променевою трубкою (ЕПТ) – іонізуючих випромінювань, а саме рентгенівського випромінювання, яке виникає усередині колби під час різкої зупинки електронів, що швидко рухаються;
- оптичного випромінювання у видимому діапазоні, а також в УФ діапазоні, яке виникає завдяки взаємодії електронів з шаром люмінофора;
- електромагнітного випромінювання в радіочастотному діапазоні. При цьому високочастотні поля виникають під впливом електронного променя і пов'язані з частотою формування елемента зображення й інтенсивністю променя, низькочастотні поля виникають у системі горизонтальної розгортки, а дуже низькочастотні поля пов'язані з генерацією (вертикальною розгорткою);
- магнітного поля, що виникає через наявність відхиляючого пристрою;
- електростатичного поля, що виникає у зв'язку з високим потенціалом анодів ЕПТ.

Опрацьовуючи тему, слід згадати ГОСТ 12.0.003-74 «Опасные и вредные производственные факторы» [32], згідно з яким під час роботи на ПК мають місце фізичні та психофізіологічні небезпечні та шкідливі виробничі чинники. Хімічні та біологічні чинники – відсутні. Необхідно визначити та згадати засоби захисту від відповідних фізичних та психофізіологічних виробничих чинників.

Одним із найголовніших виробничих факторів не тільки галузі ІТ-індустрії – це електробезпека. Коли працівник весь час знаходиться у роботі з електронними приладами, важливо створити безпечні умови з врахування загрози електричного типу.

Вплив електричного струму на організм людини залежить від сили струму, напруги, опору тіла людини, виду та частоти струму (чинники електричного характеру). До факторів неелектричного характеру належать: тривалість дії струму,

шлях проходження струму через тіло людини, індивідуальні особливості ураженого, умови довкілля тощо.

Сила електричного струму, що протікає через людину, є одним із основних факторів. Для змінного струму 50 Гц значення граничного струму фібриляції лежить в межах від 100 мА до 5А, а для постійного струму - від 300 мА до 5А. Постійний та змінний струми більше 5А викликають миттєву зупинку серця.

Таблиця 5.1 – Вплив сили струму на організм людини.

Сила струму, мА	Вплив на організм людини	
	Змінний струм частотою 50 Гц	Постійний струм
0.6 ... 1.5	Легке тремтіння рук	Без відчуттів
2 ... 3	Сильне тремтіння рук	Без відчуттів
5 ... 7	Судороги рук	Свербіж, відчуття тепла
8 ... 10	Руки можна «відірвати» від електродів, але важко, сильна біль у кистях рук	Посилене відчуття тепла
20 ... 25	Миттєвий параліч рук, «відірвати» їх від електродів неможливо, потужна біль у тілі, тяжке дихання	Посилене відчуття тепла, незначні скорочення м'язів рук
50 ... 80	Параліч дихання, фібриляція серця	Сильне відчуття тепла, скорочення у м'язах рук, тяжке дихання
90 ... 100	Параліч дихання, упродовж 3 секунд та більше – параліч серця	Параліч дихання

Опір різних тканин тіла має різне значення і складається з трьох послідовно включених опорів, два з яких - опір зовнішнього шару шкіри (епідермісу) та внутрішнього опору тканин тіла. Якщо шкіра зволожена, вона має менший опір, ніж суха. Тобто, при потовиділенні та забрудненні шкіри її захисна функція як діелектрика втрачається. Опір тіла людини є змінною величиною і залежить від місця дотику електричного дроту до тіла, величини струму, що протікає по тілу,



напруги, виду та частоти струму, площі дотику струмопровідної частини, тривалості протікання електричного струму та інших факторів.

Для об'єктивної оцінки категорії тяжкості праці на робочому місці на основі вивчення сукупного впливу всіх матеріально-виробничих елементів умов праці на організм людини (хімічних, фізичних і психологічних) розроблені спеціальні критерії. Кожний виробничий елемент умов праці  $X_i$  на робочому місці одержує бальну оцінку від 1 до 6, якщо він впливає на працівника протягом усього робочого часу. [33] В тих випадках, якщо він впливає на працівника не повний робочий день, а лише частково, елемент оцінюється його тривалістю і визначається за спеціальними діаграмами з урахуванням часу їх впливу.

Інтегральну бальну оцінку тяжкості роботи  $I$  на конкретному робочому місці можна визначити за формулою

$$I = 19,7X_{\text{ср}} - 1,6 X^2,$$

де  $X_{\text{ср}}$  – середній бал усіх біологічно значущих елементів умов праці;

Ця величина дорівнює

$$X_{\text{ср}} = \frac{\sum_{i=1}^n X_i}{n},$$

де  $n$  – кількість врахованих елементів умов праці;

За отриманою бальною оцінкою тяжкості праці можна визначити категорію тяжкості на робочому місці, за якою можна обирати захисні засоби та заходи.

### **5.3 Травми та професійні захворювання на підприємствах та робочих місцях галузі**

Аналіз причин виробничого травматизму показує, що від 30 до 47 % нещасних випадків трапляється внаслідок наявності небезпечних та шкідливих факторів. [34] У результаті впливу небезпечних і шкідливих виробничих факторів і шкідливих речовин мають місце нещасні випадки і професійні захворювання.

**Професійними захворюваннями** вважаються ті, які виникли в результаті професійної діяльності застрахованої особи і викликані впливом шкідливих виробничих чинників. **Нещасний випадок на виробництві** – раптове погіршення стану здоров'я чи настання смерті працівника під час виконання ним трудових обов'язків внаслідок короткочасного (тривалістю не довше однієї робочої зміни) впливу небезпечного або шкідливого чинника. [35]

Праця користувача комп'ютера відбувається за умови надмірного нервово-емоційного й зорового напруження, довготривалих статичних навантажень, обмеженої рухової активності, електростатичних полів, радіочастотного та рентгенівського випромінювання, які призводять до неврозів, відхилень у психіці, погіршення пам'яті, захворювань опорно-рухового апарату, серцево-судинної, імунної та статевої систем, органів зору, шкіри та ін. [36]

Вивчаючи цю тему, необхідно знати усі потенційні травми та професійні захворювання, такі, як: електротравми, професійна офтальмопатія, карпальний тунельний синдром, фізіологічні та гінекологічні порушення, інформаційні неврози і нервові перенапруження, синдром комп'ютерного стресу, синдром тривалого статичного навантаження.

Розслідування та облік нещасних випадків, професійних захворювань і аварій на виробництві проводяться згідно з НПАОП 0.00-6.02-04. [37] Вивчаючи цю тему, згадати методи аналізу травматизму. Методи технічного аналізу дозволяють виявити технічні й організаційні причини травматизму. Існує кілька методів: монографічний метод, груповий метод та економічний метод.

Як організаційно-інформативні заходи щодо попередження травматизму застосовують знаки безпеки. Існують чотири групи знаків безпеки праці: вказівні, заборонні, попереджувальні й приписувальні.

Вивчаючи цю тему, студент має знати сутність та особливості кожного методу виявлення причин травматизму, а також відрізнити знаки безпеки за групою.

Відомо, що **праця** – це сукупність фізіологічних та психічних процесів, які спонукають, програмують і регулюють діяльність людини. Важлива роль у

забезпеченні безпеки праці користувачів комп'ютеризованих систем відводиться їх психологічним властивостям і можливостям. Психологічні явища, що відбуваються в організмі оператора, є регуляторами його діяльності. Вони можуть посилювати або послаблювати дію того чи іншого сигналу на людський організм.

Вивчаючи тему, треба розуміти, що психологічні можливості користувача комп'ютера не є постійними. Вони залежать від інформаційного перевантаження, високого темпу роботи, перенапруження зорового та слухового аналізаторів, емоційного стану людини. Так, після конфліктних ситуацій, виробничих невдач, незаслужених образ з боку керівництва чи колег обсяг уваги різко знижується, порушується пам'ять. Оператор забуває послідовність дій, неправильно оцінює ситуацію, припускається грубих помилок. Тому люди, у яких психологічні можливості обмежені, а емоційний стан нестійкий, не можуть бути призначені на відповідальні роботи, до яких належить й робота за комп'ютером. [38]

Важливою умовою, що забезпечує безпеку праці, є збереження працівниками високого рівня працездатності. Під **працездатністю** розуміють потенційні можливості людини для виконання трудової діяльності протягом заданого часу з певною ефективністю. Великою, зворотною працездатності, є стомлення. **Стомлення** – це фізіологічні зміни в організмі працюючого, викликані витратою енергії в процесі трудової діяльності. Об'єктивні процеси, що виникають під час розвитку стомлення, переломлюються у свідомості працівника у вигляді відчуття втоми. **Втома** – це сукупність тимчасових змін у фізіологічному та психологічному стані людини, які з'являються внаслідок напруженої чи тривалої праці і призводять до погіршення її кількісних і якісних показників, нещасних випадків.

Вивчаючи тему, необхідно знати чинники, що збільшують або зменшують ризик виникнення втоми та перевтоми операторів.

Однією з найважливіших задач ефективної організації трудового процесу є запобігання професійному стресу. **Професійний стрес** – це багатовимірний феномен, що виражається у фізіологічних і психологічних реакціях на складну робочу ситуацію. Серед виробничих стрес-факторів можна виділити: фізичні

(вібрація, шум, забруднена атмосфера); фізіологічні (змінний графік, відсутність режиму харчування); соціально-психологічні (конфлікт ролей, перевантаження або недовантаження працівників, міжособистісні конфлікти, висока відповідальність, дефіцит часу); структурно-організаційні («організаційний стрес»).

**Стрес** – це неспецифічна реакція організму у відповідь на дуже сильну дію (подразник) зовні, яка перевищує норму, а також відповідна реакція нервової системи, стан напруження, який виникає внаслідок дії сильного подразника, незвичайної ситуації. Залежно від подразника, що діє на людину, розрізняють два види стресу: дистрес – за дії негативних емоцій та евстрес – за дії позитивних емоцій. Студенту необхідно знати стадії стресу за класифікацією Г. Сельє.

Слід зауважити, що найгіршим проявом професійного стресу для працівника є професійне вигорання. За визначенням Всесвітньої організації охорони здоров'я (2001), **синдром професійного вигорання** (burnout syndrome) – це стан фізичного, емоційного або мотиваційного виснаження, що характеризується порушенням продуктивності роботи та втому, підвищенням схильності до соматичних захворювань, а також вживанням алкоголю чи інших психоактивних речовин з метою отримання тимчасового полегшення, що має тенденцію до розвитку фізіологічної залежності та (у багатьох випадках) суїцидальної поведінки.

Основними механізмами, що протидіють виникненню синдрому професійного вигорання є саморегуляція і само менеджмент. **Саморегуляція** – це здатність людини підтримувати свій психічний тонус та працездатність за несприятливих умов. Вивчаючи тему, необхідно приділити увагу психічній саморегуляції та знати її методи. **Самоменеджмент** розуміють як послідовне і цілеспрямоване використання ефективних методів, прийомів, технік і технологій, спрямованих на саморозвиток свого творчого потенціалу з метою свідомого управління своїм життям і максимальної самореалізації у всіх його сферах.

Однією з важливих задач з управління діяльністю підприємства, пов'язаних з охороною праці, є забезпечення таких умов на підприємстві, що роблять можливим виконання роботи в безпечній і комфортній обстановці.

**Трудова мотивація** – це процес стимулювання окремого виконавця або групи людей до діяльності, яка спрямована на досягнення мети підприємства, до продуктивного виконання ухвалених рішень або запланованих робіт. Мотивація включає внутрішній стан людини, званий потребою, і зовнішні чинники, що впливають на мотивацію, так звані стимули. Поведінка людини визначається потребами і стимулами, які домінують у даний момент часу.

Неефективна система мотивації може викликати у працівників незадоволеність, що завжди викликає зниження продуктивності праці. З іншого боку, ефективна система стимулює продуктивність персоналу, підвищує ефективність людських ресурсів.

Необхідно розуміти, що стимулювання до належного рівня мотивації може існувати у двох взаємозв'язаних формах:

**Матеріальне стимулювання** виявляється в тому, що працівник при обумовленому обсязі і якості виконуваної роботи може розраховувати на отримання тих або інших матеріальних благ у натуральній або грошовій формі.

**Нематеріальне стимулювання** виявляється в тому, що працівник при обумовленіх вигляді, обсязі і якості виконуваної роботи може розраховувати на отримання тих або інших благ у нематеріальній формі, яка не має прямої грошової оцінки.

Студент має знати види матеріального та нематеріального стимулювання до мотивації безпеки праці.

Вивчаючи тему, необхідно розглянути **методи управління мотивацією праці**:

- адміністративні методи засновані на ухваленні рішень керівниками у сфері нематеріальних заохочень і покарань, розподілу повноважень і відповідальності, визначенням графіка роботи тощо;
- економічні методи засновані на застосуванні матеріальної винагороди як компенсації за трудові зусилля працівника і досягнуті їм результати.

## Висновок до розділу 5

Під час дослідження умов праці людини за комп'ютером було встановлено забезпечення безпеки праці – необхідної умови будь-якого технологічного процесу, зокрема з застосування комп'ютерної техніки. Із зростанням кількості автоматизованих виробництв і комп'ютерних систем зазнають зміни чинники, які впливають на формування умов праці, які у свою чергу призводять до негативної дії на фізіологічну систему людини, знижуючи її працездатність і ініціюючи неправильні дії.

Проаналізовано міжнародні та державні норми з питань безпеки праці користувачів. Вони базуються на ряді принципів, основними з яких є пріоритет життя і здоров'я працівників, повна відповідальність роботодавця за створення безпечних та належних умов праці. Було встановлено, що роботодавці зобов'язані оцінювати виробничі ризики й забезпечувати адекватні захисні і профілактичні заходи, гарантувати відповідне навчання й інструктаж працівників з дотримання заходів безпеки, а також надавати працівникам інформацію та консультації і дозволяти їм брати участь в обговоренні всіх питань із забезпечення безпеки й гігієни праці

Досліджено професійні захворювання, які виникли в результаті професійної діяльності застрахованої особи і викликані впливом шкідливих виробничих чинників. Була опрацьована інтегральна бальна оцінка тяжкості роботи на конкретному робочому місці. Розраховано вплив сили струму на організм людини.

Було розглянуто одну з найважливіших задач ефективної організації трудового процесу – запобігання професійному стресу. Це багатовимірний феномен, що виражається у фізіологічних і психологічних реакціях на складну робочу ситуацію. У результаті дослідження встановлено, що психологічні можливості користувача комп'ютера не є постійними. Вони залежать від інформаційного перевантаження, високого темпу роботи, перенапруження зорового та слухового аналізаторів, емоційного стану людини.

## ВИСНОВКИ

У ході дослідження теоретичної часті бакалаврської кваліфікаційної роботи було з'ясовано як розвивались відеоігри, чому прийшли до концепції «занурення» і як з цим пов'язана 3D-модель персонажу. Було розглянуто аналоги репрезентуючих моделей у відеоіграх і чому гравець хоче себе з ними ототожнювати. Вивчено вплив штучного інтелекту на відеоігри та 3D-моделювання, як вони взаємодіяли одне з одним та що з цієї взаємодії придбали. Були проаналізовані та протипоставлені найпопулярніші технології для розробки відеоігор та для роботи з 3D. У ході практичної часті було обрано найкращі, судячи з порівняних сильних та слабких сторін, програмні забезпечення для виконання поставленої задачі. Було розроблено програму на основі функції для роботи з 3D моделями засобами штучного інтелекту Pixel-aligned Implicit Function (PIFu). Готовий результат було відредаговано за допомогою 3D-редактору Blender, розроблено скелет та анімацію у застосунку Adobe Mixamo та імпортовано у ігровий додаток, розроблений за допомогою ігрового движка Unity. Таким чином, основна ціль дипломної роботи було вирішено.

Було досліджено, що технології для створення достовірного відображення людини у вигляді 3D-персонажу наразі тільки починають набувати популярності. В відкритому доступі таких технологій дуже мало, але одні з найкрупніших компаній досліджують цю потенційну сферу як і для відеоігор, так і для нового напрямлення – метавсесвітів. Враховуючи темпи росту цих технологій, можна припустити, що через короткий час вони придбають світову популярність і кількість і якість технологій у цьому напрямленні сильно зростуть. Але зараз це тільки починає активно розвиватись, ігрові движки наразі не підтримують такого функціоналу, але незабаром, враховуючи інформацію, що було досліджено у цій роботі, це зміниться.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) ІСТОРІЯ РОЗВИТКУ КОМП'ЮТЕРНИХ ТА ВІДЕО ІГОР. ІСТОРІЯ ВИНИКНЕННЯ КОМП'ЮТЕРНИХ ІГОР. Фондеко: веб-сайт. URL:<https://fondeco.ru/uk/istoriya-razvitiya-kompyuternyh-i-video-igr-istoriya-vozniknoveniya/> (дата звернення: 05.05.2022).
- 2) Історія створення комп'ютерних ігор, та їх розвиток на сучасному етапі. Студфайл: веб-сайт. URL: <https://studfile.net/preview/4364269/page:3/> (дата звернення: 07.05.2022).
- 3) The evolution of 3D games. TechRadar: веб-сайт. URL: <https://www.techradar.com/uk/news/gaming/the-evolution-of-3d-games-700995> (дата звернення: 07.05.2022).
- 4) The Power of Experience: The Wonders of Video Game Immersion. GameDesigning: веб-сайт. URL: <https://www.gamedesigning.org/learn/game-immersion/> (дата звернення: 07.05.2022).
- 5) Flow and Immersion in Video Games: The Aftermath of a Conceptual Challenge. FrontierSin: веб-сайт. URL: <https://www.frontiersin.org/articles/10.3389/fpsyg.2018.01682/full> (дата звернення: 10.05.2022).
- 6) HOW TO MAKE IMMERSIVE GAME DESIGN. USV: веб-сайт. URL: <https://usv.edu/blog/how-to-make-immersive-game-design/> (дата звернення: 11.05.2022).
- 7) The Best Video Game Character Creators Of All Time, Ranked. TheGamer: веб-сайт. URL: <https://www.thegamer.com/video-game-best-character-creators-ranked/> (дата звернення: 12.05.2022).
- 8) Use Character Creator Games To Improve Your Character Design Skills. GameDesigning: веб-сайт. URL: <https://www.gamedesigning.org/animation/character-creator/> (дата звернення: 12.05.2022).



- 9) How to Design a Video Game Character. MasterClass: веб-сайт. URL: <https://www.masterclass.com/articles/how-to-design-a-video-game-character> (дата звернення: 14.05.2022).
- 10) The Future Of AI In Gaming. GameDesigning: веб-сайт. URL: <https://www.gamedesigning.org/gaming/ai-in-gaming/>
- 11) Artificial Intelligence in Video Games. TowardsDataScience: веб-сайт. URL: <https://towardsdatascience.com/artificial-intelligence-in-video-games-3e2566d59c22> (дата звернення: 14.05.2022).
- 12) AI in Game Development. ItChronicles: веб-сайт. URL: <https://itchronicles.com/artificial-intelligence/ai-in-game-development/>(дата звернення: 14.05.2022).
- 13) The Impact of Machine Learning on 2D/3D Registration for Image-Guided Interventions: A Systematic Review and Perspective. FrontierSin: веб-сайт. URL: <https://www.frontiersin.org/articles/10.3389/frobt.2021.716007/full>(дата звернення: 16.05.2022).
- 14) Pipes of AI – Machine Learning Assisted 3D Modeling Design. Springer: веб-сайт. URL: [https://link.springer.com/chapter/10.1007/978-981-33-4400-6\\_2](https://link.springer.com/chapter/10.1007/978-981-33-4400-6_2)(дата звернення: 16.05.2022).
- 15) Unity Overview. Unity3D: веб-сайт. URL: <https://docs.unity3d.com/520/Documentation/Manual/UnityOverview.html>(дата звернення: 16.05.2022).
- 16) An overview of Unreal Engine. Packt: веб-сайт. URL: <https://www.packt.com/overview-unreal-engine/#:~:text=Unreal%20Engine%20is%20a%20game,the%20gameplay%20of%20your%20game.> (дата звернення: 19.05.2022).
- 17) Overview of Godot's key concepts. GodotEngine: веб-сайт. URL: [https://docs.godotengine.org/en/stable/getting\\_started/introduction/key\\_concepts\\_overview.html](https://docs.godotengine.org/en/stable/getting_started/introduction/key_concepts_overview.html)(дата звернення: 19.05.2022).

- 18) GameMaker Manual. yoyogames.com: веб-сайт. URL: [https://manual.yoyogames.com/#\\_ga=2.175485410.1112391503.1655801309-425588934.1655801309&t=Content.htm](https://manual.yoyogames.com/#_ga=2.175485410.1112391503.1655801309-425588934.1655801309&t=Content.htm)(дата звернення: 19.05.2022).
- 19) Autodesk Maya Overview. TrustRadius: веб-сайт. URL: <https://www.trustradius.com/products/autodesk-maya/reviews#reviews>(дата звернення: 19.05.2022).
- 20) What Is 3ds Max? – Simply Explained. ALL3DP: веб-сайт. URL: <https://all3dp.com/2/what-is-3ds-max-simply-explained/>(дата звернення: 19.05.2022).
- 21) Zbrush Reviews. Capterra: веб-сайт. URL: <https://www.capterra.com/p/210456/Zbrush/reviews/>(дата звернення: 19.05.2022).
- 22) Blender 2.9 Review. Develop3D: веб-сайт. URL: <https://develop3d.com/visualisation/blender-2-9-review/>(дата звернення: 19.05.2022).
- 23) How to Use Mixamo to Animate Custom 3D Models. MakeUseOf: веб-сайт. URL: <https://www.makeuseof.com/use-mixamo-animate-custom-3d-models/>(дата звернення: 19.05.2022).
- 24) Everything About Python — Beginner To Advanced. Medium: веб-сайт. URL: <https://medium.com/fintechexplained/everything-about-python-from-beginner-to-advance-level-227d52ef32d2>(дата звернення: 22.05.2022).
- 25) Libraries in Python. GeeksForGeeks: веб-сайт. URL: <https://www.geeksforgeeks.org/libraries-in-python/>(дата звернення: 22.05.2022).
- 26) Google Colab for Machine Learning Projects. MachineLearningMastery: веб-сайт. URL: <https://machinelearningmastery.com/google-colab-for-machine-learning-projects/>(дата звернення: 22.05.2022).

- 27) PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. Shunsukesaito: веб-сайт. URL: <https://shunsukesaito.github.io/PIFu/>(дата звернення: 22.05.2022).
- 28) Гогіташвілі Г.Г., Управління охороною праці та ризиком за міжнародними стандартами: навч. посібник / Г.Г. Гогіташвілі, Є.Т. Карчевські, В.М. Лапін. – К.: Знання, 2007. – 367 с.
- 29) Жидецький В.Ц., Охорона праці користувачів комп'ютерів / В.Ц. Жидецький. – 2-е вид., доп. – Львів: Афіша, 2000. – 176 с.
- 30) Охрана труда. Сборник задач : учеб. пособие / Б.В. Дзюндзюк, В.Г. Иванов [и др.]. – Харьков: ХНУРЭ, 2006. – 244 с.
- 31) Крушельницька Я.В., Фізіологія і психологія праці : навч. посібник / Я.В. Крушельницька. – К.: КНЕУ, 2000. – 232 с.
- 32) Охорона праці в галузі комп'ютерингу : підручник / За науковою редакцією В.В. Пасічника. – Львів: «Магнолія 2006», 2012. – 544 с.
- 33) Основы риск-менеджмента / М. Круи, Р.М. Марк, Д. Галій. – М.: Изд-во «Юрайт», 2011. – 400 с.
- 34) Демирчоглян Г.Г., Компьютер и здоровье / Г.Г. Демирчоглян. – М.: Изд-во «Лукоморье», Новый Центр, 1997. – 256 с.
- 35) Основы инженерной психологии / Под ред. Б.Ф. Ломова. – М.: Высш. шк., 1977. – 335 с.
- 36) Моніторинг умов праці : підручник/ В.І. Голінько, С.І. Чеберячко, М.В. Шибка, О.О. Яворська. – 2-ге вид. – Д.: М-во освіти і науки України; Нац. гірн. ун-т, 2014. – 230 с.
- 37) Безпека життєдіяльності (забезпечення соціальної, техногенної та природної безпеки): : навч. посібник/ В.В. Бегун, І.М. Науменко. – К., 2004. – 328 с.
- 38) Методичні рекомендації для проведення атестації робочих місць за умовами праці. Затверджено міністром праці України 1.09.1992 р.