

Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки

(повна назва факультету)

Кафедра комп'ютерних та інформаційних технологій і систем

(повна назва кафедри)

**Пояснювальна записка
до дипломного проекту (роботи)**

магістра

(освітньо-кваліфікаційний рівень)

на тему

«Великі мовні моделі у вирішенні задач маркетингу

(На прикладі маркетингових ігор)»

Виконав: студент б курсу, групи 601-ТН
спеціальності

122 Комп'ютерні науки

(шифр і назва спеціальності)

Коломієць В.М.

(прізвище та ініціали)

Керівник Ляхов О.Л.

(прізвище та ініціали)

Рецензент Олександр Лактіонов

(прізвище та ініціали)

Полтава – 2025 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ І СИСТЕМ**

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

спеціальність 122 «Комп'ютерні науки»

на тему

**«Великі мовні моделі у вирішенні задач маркетингу
(На прикладі маркетингових ігор)»**

Студента групи 601-ТН Коломійця Володимира Миколайовича

Керівник роботи
доктор технічних наук,
професор Ляхов О.Л.

Завідувач кафедри
кандидат фізико-математичних наук,
Двірна О.А

РЕФЕРАТ

Кваліфікаційна робота магістра: 65 сторінок, 17 малюнків, 17 джерел.
Об'єкт дослідження: Великі мовні моделі у вирішенні задач маркетингу (На прикладі маркетингових ігор)

Мета роботи: розробити гру з використанням великих мовних моделей (ВММ або LLM) яка буда виступати як інтерфейс для маркетингових досліджень

Методи: проектування та розроблення розроблення гри за допомогою Godot Engine та GDScript, з використанням Llama-3-instruct як базової моделі

Ключові слова: Godot Engine, GDScript, ігровий рушій, LLM, маркетинг, чатбот, симуляція, торгівля.

ABSTRACT

Master's qualification work: 65 pages, 17 figures, 17 sources. Object of research: Large language models in solving marketing problems (on the example of marketing games)

Goal of work: to develop a game using large language models (LLM or LLM) that will act as an interface for marketing research

Methods: design and development of game development using Godot Engine and GDScript, using Llama-3-instruct as a base model

Keywords: Godot Engine, GDScript, game engine, LLM, marketing, chatbot, simulation, trade.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	5
ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	9
1.1 Об’єкт дослідження.....	9
1.2 Аналіз предметної області	10
1.3 Актуальність дослідження	11
1.4 Аналіз існуючих програмних рішень	12
1.4.1 Аналіз схожих досліджень	14
1.5 Особливості LLM для маркетингових задач.....	19
1.6 Постановка задачі.....	21
РОЗДІЛ 2. МЕТОДИ РОЗРОБКИ КОМП’ЮТЕРНОЇ ГРИ ЯК ІНТЕРФЕЙСУ ВЗАЄМОДІЇ З LLM	24
2.1. Методи розробки інформаційних технологій	24
2.1.1 Інтеграція з LLM	25
2.1.2 Розробка контенту	27
2.1.3 Розробка ігрової механіки.....	28
2.1.4 Тестування	30
2.2 Вибір ігрового рушія	32
РОЗДІЛ 3. РОЗРОБКА КОМП’ЮТЕРНОЇ ГРИ ЯК ЧАТ-БОТ ІНТЕРФЕЙСУ ..	35
3.1 Проектування модулів гри	35
3.1.1 Базові модулі гри	37
3.1.2 Вікно чату	42
3.2 Взаємодія з LLM.....	46
3.2.1 Промпт-інжиніринг	49
3.2.2 Файн-тюнінг моделі	52
3.2.3 Обробка даних LLM	55
ВИСНОВОК	57
ДЖЕРЕЛА	59
ДОДАТОК А СКРИНШОТИ ПРОГРАМИ	61
ДОДАТОК Б КОД ОСНОВНИХ СКРИПТІВ ПРОГРАМИ.....	64
ДОДАТОК В ПРИКЛАДИ ДІАЛОГУ З ЧАТ-БОТОМ.....	67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Комп'ютерна (відео) гра — програма створена з ціллю розваги, в якій гравець повинен виконувати певні цілі

Ігровий рушій — середовище розробки створене спеціально для розробки ігор

LLM — Великі Мовні Моделі (Large Language Models) тип процесійних моделей для обробки натуральної мови

Чат-бот — в контексті роботи, мається на увазі взаємодія з ВММ у вигляді чату\розмови

ВСТУП

Великі мовні моделі (LLM) — це проривна технологія, яка відкриває нові горизонти для автоматизації роботи з текстом і розуміння природної мови. Їх унікальна здатність до генерації текстів, що за якістю майже не відрізняються від людських, і вміння аналізувати гігантські обсяги даних стали основою багатьох сучасних рішень у різних галузях. Від освіти до програмування, від розваг до бізнесу — LLM вражають своєю універсальністю. Зокрема, маркетинг став одним із тих напрямів, де ця технологія не просто знайшла застосування, а й заклала основу для революційних змін у комунікації з клієнтами, створенні контенту та прогнозуванні ринкових трендів.

Сьогодні LLM — це не просто інструмент для роботи з текстом. Це інтелектуальні системи, здатні вчитися на великому масиві даних, розпізнавати складні патерни та генерувати контент, який є як зрозумілим, так і релевантним для конкретної аудиторії. Їх використання варіюється від створення текстів для статей, оглядів або технічної документації до глибокого аналізу поведінки користувачів у соціальних мережах. У маркетингу ці моделі стали невід’ємною частиною таких процесів, як персоналізація рекламних кампаній, створення автоматизованих чат-ботів, які здатні вести діалоги з клієнтами на рівні живого спілкування, та навіть прогнозування продажів.

Уявіть, що маркетинг — це величезний механізм, який обертається завдяки взаємодії між компанією та її клієнтами. Великі мовні моделі стали своєрідним мастилом для цього механізму, дозволяючи робити цю взаємодію швидшою, точнішою та, головне, ефективнішою. Наприклад, LLM використовують для аналізу вподобань і потреб клієнтів на основі даних із соціальних мереж, анкет або історій покупок. Замість того, щоб витратити час і ресурси на класичне дослідження ринку, компанії можуть отримати інсайти за допомогою моделі, яка обробляє тисячі точок даних і дає рекомендації щодо того, який продукт або послугу краще запропонувати конкретній людині.

Автоматизація спілкування також стала важливою складовою маркетингових процесів. Чат-боти, побудовані на основі LLM, здатні відповідати на складні запити клієнтів, допомагати їм обрати товар чи послугу і навіть вирішувати конфліктні ситуації. Завдяки цьому компанії знижують навантаження на служби підтримки й можуть приділити більше уваги стратегічним завданням.

Окремо варто згадати роль LLM у створенні текстового контенту. Написання описів товарів, генерація унікальних рекламних слоганів або навіть створення цікавих історій для маркетингових кампаній — усе це тепер можна автоматизувати. Але справа не лише в швидкості; важливо, що LLM можуть адаптувати стиль написання відповідно до конкретної аудиторії, роблячи текст більш релевантним і емоційно привабливим.

Інтеграція великих мовних моделей у маркетинг відкриває безліч можливостей. Однією з найцікавіших є гейміфікація — створення маркетингових ігор, які залучають клієнтів через інтерактивний контент. Завдяки LLM ігри стають динамічнішими та інтелектуальнішими: персонажі можуть взаємодіяти з гравцем, а сценарії змінюватися залежно від дій користувача. Це не лише розважає, але й дозволяє зібрати цінну інформацію про вподобання клієнтів.

Ще одна важлива можливість — прогнозування. Наприклад, LLM аналізують історичні дані про продажі й можуть вказати, який продукт варто просувати більше, або підказати, коли краще запускати ту чи іншу рекламну кампанію. Усе це дозволяє приймати більш обґрунтовані рішення та зменшувати ризики.

Втім, із використанням LLM пов'язані й певні труднощі. По-перше, це етичні питання: моделі працюють на основі величезних обсягів даних, що часто включають персональну інформацію користувачів. Це створює ризик порушення конфіденційності або маніпуляцій аудиторією. По-друге, ефективність роботи

LLM на пряму залежить від якості даних, на яких вони навчаються. Якщо дані містять помилки або упередження, це впливає на результати роботи моделі.

Попри ці виклики, потенціал LLM для трансформації маркетингу залишається величезним. Вони не лише роблять процеси швидшими та економічнішими, але й дозволяють створювати інноваційні підходи до взаємодії з клієнтами, які раніше були неможливими.

РОЗДІЛ 1.

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Об'єкт дослідження

Об'єктом даного дослідження є інтеграція великих мовних моделей (LLM) у маркетингові процеси як засобу для моделювання та оптимізації промоційної діяльності. У центрі уваги знаходиться моделювання процесу взаємодії між компанією та споживачем через інструменти штучного інтелекту, що базуються на сучасних методах машинного навчання. Це дослідження зосереджене на використанні LLM для побудови ефективного діалогу між брендом і потенційним клієнтом, що є ключовим елементом просування товарів та послуг.

На відміну від традиційних підходів, які спираються на математичне моделювання, базоване на теорії ймовірностей та ігор, дане дослідження орієнтоване на адаптивний підхід. Великі мовні моделі здатні навчатися на основі великих наборів текстових даних, враховуючи контекст і тональність діалогу. Це дозволяє створювати більш гнучкі й персоналізовані маркетингові стратегії, які адаптуються до потреб конкретних споживачів у режимі реального часу.

Особливий акцент зроблено на вивченні можливостей LLM для моделювання діалогу, в якому роль споживача або покупця виконує штучний інтелект. Такий підхід дає змогу аналізувати різні стратегії взаємодії та виявляти оптимальні способи комунікації, які сприяють досягненню маркетингових цілей. Наприклад, мовні моделі можуть імітувати поведінку реальних користувачів, реагуючи на запити та пропозиції компанії, що дозволяє створити реалістичну симуляцію діалогу для тестування маркетингових ідей.

У рамках дослідження пропонується використовувати LLM не лише для аналізу статистичних даних, але й для активної взаємодії з аудиторією через текстові та голосові інтерфейси. Це відкриває можливість для створення

інтерактивних сценаріїв, де моделі адаптуються до поведінки користувача і формують індивідуальні стратегії взаємодії. Такий підхід дозволяє зменшити невизначеність у прогнозуванні результатів маркетингових дій і підвищити ефективність промоції товарів та послуг.

1.2 Аналіз предметної області

Сучасний маркетинг переживає трансформацію завдяки стрімкому впровадженню технологій машинного навчання, серед яких особливе місце займають великі мовні моделі (LLM). Ці моделі дозволяють бізнесу автоматизувати багато процесів, персоналізувати взаємодію з клієнтами та отримувати глибші інсайти на основі аналізу великих обсягів даних.

Одна з ключових проблем традиційного маркетингу полягає в узагальненому підході до аналізу споживачів і створення кампаній. Використання LLM змінює це, забезпечуючи можливість адаптивного моделювання, що враховує індивідуальні характеристики споживачів і контекст. Наприклад, інтеграція таких моделей у процеси аналізу відгуків клієнтів дозволяє не лише визначати загальну тональність повідомлень, а й ідентифікувати конкретні потреби, запити та емоції клієнтів.

У промоційній діяльності LLM також мають значний потенціал. Вони можуть автоматично генерувати персоналізовані рекламні тексти, створювати сценарії для чат-ботів або навіть моделювати цілі маркетингові стратегії. Наприклад, LLM можна використовувати для аналізу поведінки користувачів на веб-сайтах або в соціальних мережах, що дозволяє прогнозувати їхні дії та пропонувати релевантний контент.

Ще одним важливим аспектом є оптимізація витрат. Традиційні маркетингові дослідження, такі як опитування або фокус-групи, потребують значних ресурсів. Натомість LLM здатні виконувати схожі завдання швидше і дешевше. Наприклад, вони можуть створювати симуляції взаємодії між брендом

і клієнтом, що дозволяє тестувати нові ідеї або продукти ще до їх виходу на ринок.

У сучасному маркетингу особливу увагу приділяють багатоканальним комунікаціям, і LLM тут відіграють важливу роль. Вони можуть об'єднувати дані з різних джерел — електронної пошти, соціальних мереж, сайтів, мобільних додатків — і створювати цілісний профіль клієнта. Це дозволяє підвищити точність прогнозів і забезпечити інтегровану взаємодію з аудиторією.

Підсумовуючи, великі мовні моделі змінюють підходи до маркетингових досліджень і стратегій. Вони не лише покращують аналіз даних, але й дозволяють компаніям створювати більш ефективні й адаптивні рішення, що відповідають сучасним викликам ринку. Інтеграція LLM у маркетинг стає одним із ключових інструментів, який визначатиме успішність компаній у найближчому майбутньому.

1.3 Актуальність дослідження

Актуальність даного дослідження обумовлена стрімким розвитком технологій штучного інтелекту та їхнього впливу на сучасні бізнес-процеси, зокрема у сфері маркетингу. У світі, де персоналізація є ключовим фактором успіху у взаємодії з клієнтами, великі мовні моделі (LLM) відкривають нові можливості для розуміння та задоволення потреб споживачів. Їх здатність аналізувати великі обсяги даних, генерувати адаптивний контент і автоматизувати взаємодії створює значну конкурентну перевагу для бізнесу. Використання LLM у маркетингових іграх, які поєднують елементи гейміфікації та інтерактивного досвіду, дозволяє підвищити залученість клієнтів і забезпечити більш глибоке розуміння їхніх вподобань.

Сьогодні маркетингові ігри є одним із найбільш перспективних інструментів для взаємодії з цільовою аудиторією. Вони дозволяють брендам не лише рекламувати свої продукти, але й створювати позитивний досвід для

споживачів, що сприяє формуванню лояльності до бренду. У цьому контексті інтеграція великих мовних моделей, таких як LLaMA 3, та технологій розпізнавання голосу, таких як Vosk, надає нові можливості для розширення функціоналу ігор. Це особливо актуально в умовах зростаючої конкуренції, коли компанії змушені шукати інноваційні підходи для привернення уваги та утримання клієнтів.

Актуальність дослідження також визначається тим, що традиційні методи маркетингових досліджень, такі як опитування чи фокус-групи, часто є дорогими, тривалими та не завжди відображають реальні потреби споживачів. Як зазначається у дослідженні Гарвардської бізнес-школи, великі мовні моделі можуть значно зменшити витрати на дослідження та скоротити час на отримання даних, що робить їх особливо привабливими для сучасного бізнесу. Використання LLM дозволяє моделювати відповіді споживачів на маркетингові запитання, створюючи можливість для швидкого аналізу та коригування стратегій у реальному часі.

Додатково, інтерес до теми посилюється завдяки інноваційному підходу до інтерактивності у маркетингових іграх. Інтеграція голосових технологій робить взаємодію з гравцями більш природною, що сприяє зростанню залученості аудиторії. Наприклад, система Vosk дозволяє реалізовувати голосове управління в ігрових сценаріях, роблячи процес взаємодії більш інтуїтивним та динамічним. Це відповідає трендам сучасного ринку, де споживачі віддають перевагу зручним і персоналізованим рішенням.

1.4 Аналіз існуючих програмних рішень

Сучасні програмні рішення, що інтегрують великі мовні моделі (LLM) у різні проекти, відкривають широкі можливості для індустрії ігор та маркетингу. Це особливо помітно у випадках, коли LLM використовуються для створення інтерактивного та адаптивного ігрового досвіду. Завдяки здатності цих моделей

аналізувати й генерувати текст із високим рівнем природності, вони стають важливим інструментом у створенні діалогових систем, персоналізованих сценаріїв та адаптивного контенту, який реагує на дії гравця чи клієнта.

Однією з найбільш яскравих реалізацій великих мовних моделей є їх інтеграція у чатботи. Такі платформи, як ChatGPT від OpenAI чи Bard від Google, уже широко використовуються у вебсервісах, програмах підтримки користувачів і навіть у творчих процесах. Чатботи на основі LLM здатні підтримувати складні розмови, розпізнавати контекст і забезпечувати індивідуальний підхід до кожного користувача. У сфері відеоігор подібні технології дозволяють створювати NPC, які не тільки реагують на задані фрази, але й пропонують унікальні варіанти діалогу, що залежить від дій гравця.

Рішення у галузі маркетингу також демонструють значний прогрес. Системи на базі LLM використовуються для автоматизації відповіді на запити клієнтів, створення персоналізованих рекомендацій і навіть проведення маркетингових досліджень у режимі реального часу. Одним із найпопулярніших застосувань є генерація рекламного контенту, який враховує специфіку аудиторії, стиль бренду і навіть актуальні тренди. Такі платформи, як Jasper чи Writesonic, дозволяють компаніям швидко створювати рекламні кампанії з високою якістю тексту, мінімізуючи людське втручання.

У сфері розробки ігор існує значний інтерес до використання LLM для автоматизації сценарного планування, генерації унікальних сюжетів чи описів, а також для створення інструментів, що спрощують розробку. Наприклад, платформи на кшталт Inworld AI зосереджуються на створенні інтелектуальних NPC для відеоігор, де мовна модель слугує основою для природної поведінки персонажа. Інші рішення спрямовані на генерацію процедурних текстів, які можуть включати описи локацій, внутрішній монолог персонажів чи навіть випадкові квести. Це значно полегшує роботу розробників, особливо в малих студіях із обмеженим бюджетом.

Існують також спеціалізовані програмні бібліотеки та платформи для інтеграції великих мовних моделей у проекти. OpenAI API, наприклад, надає широкий спектр інструментів для використання мовних моделей у різних сценаріях — від діалогових систем до аналізу даних. Інші фреймворки, такі як LangChain, розроблені спеціально для того, щоб спрощувати впровадження складних моделей у більш широкий контекст, дозволяючи взаємодіяти з кількома базами даних чи зовнішніми API. У контексті відеоігор ці платформи допомагають розробникам швидко інтегрувати мовну модель у структуру ігрового проекту, враховуючи обмеження апаратного забезпечення та оптимізацію продуктивності.

Ще одним важливим напрямом є поєднання мовних моделей із технологіями розпізнавання голосу. Інструменти, такі як Vosk чи Whisper, дозволяють перетворювати голосовий ввід на текст, який потім аналізується LLM. Це відкриває можливості для голосового управління, інтерактивних діалогів у реальному часі та створення нового рівня занурення у гру. Завдяки такій інтеграції, гравець може спілкуватися з NPC голосом, отримуючи природні відповіді, що додає новий рівень реалізму до ігрового досвіду.

Хоча більшість програмних рішень спрямовані на автоматизацію та персоналізацію, слід зазначити, що розробка таких систем досі є викликом через обмеження технологій. Використання LLM у реальному часі, особливо у відеоіграх, часто вимагає значних обчислювальних ресурсів. Тому існує тенденція до оптимізації моделей, як, наприклад, проекти на основі LLaMA, які забезпечують високу продуктивність за мінімальних ресурсів. У результаті розробники отримують можливість реалізовувати інтерактивні системи навіть у середовищах із обмеженими технічними можливостями.

1.4.1 Аналіз схожих досліджень. У дослідженні "Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus?" автор Джон Дж. Гортон аналізує потенціал великих мовних моделей (LLMs), зокрема GPT-3, як інструментів симуляції економічної поведінки. Автор пропонує розглядати LLMs як альтернативу класичній економічній моделі *homo economicus*, замінивши її на *homo silicus* – комп'ютерну модель, яка здатна відтворювати людську поведінку на основі текстових підказок. Ця концепція ґрунтується на здатності LLMs, завдяки масштабному тренуванню на текстових даних, демонструвати поведінку, яка часто є наближеною до реальних людських рішень у різноманітних економічних сценаріях.

Автор розпочинає роботу з теоретичного обґрунтування придатності LLMs для моделювання людської поведінки. Він підкреслює, що LLMs, як і будь-які інші моделі, є недосконалими, але можуть бути корисними, якщо їхні результати допомагають отримати практичні інсайти. Моделі GPT-3, на думку автора, мають два ключових аспекти, що роблять їх унікальними: по-перше, вони є обчислювальними моделями, створеними для генерації реалістичних відповідей, а по-друге, вони акумулюють велику кількість латентної соціальної інформації завдяки навчальному корпусу, який включає мільйони текстів про економічні, соціальні та поведінкові аспекти. Автор визнає, що LLMs, як і будь-які інші інструменти, мають обмеження, зокрема схильність до помилок через недосконалий навчальний корпус або некоректні запити. Проте він наголошує, що їхня здатність адаптуватися до різноманітних умов робить їх корисними для моделювання складних економічних сценаріїв.

Дослідження включає чотири експерименти, кожен із яких побудований на основі відомих експериментів у галузі поведінкової економіки. У першому експерименті використовуються диктаторські ігри, які дозволяють вивчити вплив заданих уподобань на поведінку агентів. GPT-3 продемонструвала здатність змінювати свої рішення залежно від інструкцій, наприклад,

максимізувати рівність, загальну вигоду чи власний інтерес. Менш потужні моделі показали обмежену гнучкість у відповідях, що підкреслює важливість обраного рівня технології.

Другий експеримент досліджував питання справедливості в умовах отримання прибутку, використовуючи сценарій підвищення ціни на снігові лопати після снігопаду. GPT-3 оцінила ситуації за різних умов, включаючи варіації ціни, політичних "поглядів" агентів і формулювання питань. Наприклад, агенти із соціалістичними поглядами розглядали значне підвищення ціни як дуже несправедливе, тоді як лібертаріанці демонстрували більшу толерантність до таких змін. Цей експеримент показав, як політичні уподобання або формулювання запиту можуть суттєво вплинути на сприйняття справедливості.

У третьому експерименті автор досліджував упередження статус-кво на прикладі розподілу бюджету між покращенням автомобільної та дорожньої безпеки. Результати підтвердили, що GPT-3 демонструє схильність до збереження статус-кво, коли один із варіантів явно подається як базовий. Це узгоджується з висновками реальних досліджень і демонструє, як моделі можуть імітувати людські когнітивні упередження.

Четвертий експеримент був присвячений моделюванню ефектів впровадження мінімальної заробітної плати на вибір наймача. Автор створив сценарій, у якому роботодавець мав обрати між двома кандидатами з різним досвідом і зарплатними запитами. Результати показали, що встановлення мінімальної зарплати змушувало наймачів надавати перевагу більш досвідченим працівникам, що відповідає реальним економічним спостереженням. Цей експеримент також підкреслює гнучкість моделей у тестуванні різноманітних економічних політик.

Автор наголошує, що результати таких симуляцій слід перевіряти емпірично, але їхня перевага полягає у швидкості, дешевизні та відсутності етичних проблем, пов'язаних із залученням реальних учасників. Експерименти з використанням GPT-3 можна швидко адаптувати до різних умов, змінювати

параметри та збільшувати вибірки, що робить їх корисним інструментом для попереднього аналізу гіпотез. У висновку автор зазначає, що хоча LLMs не можуть замінити реальні експерименти, вони здатні суттєво спростити їх підготовку, дозволяючи дослідникам краще зрозуміти, як формулювати запитання та які фактори враховувати під час аналізу поведінки.

Дослідження "Using LLMs for Market Research" Джеймса Бранда, Айєлет Ізраелі та Дональда Нгве зосереджується на вивченні можливостей великих мовних моделей (LLMs), таких як GPT-3.5 Turbo, у контексті проведення маркетингових досліджень. Автори досліджують, як ці моделі можуть імітувати відповіді споживачів на опитування, використовуючи знання, отримані під час тренування на величезних обсягах текстових даних. Основною метою роботи є визначення того, наскільки реалістичними є результати, отримані за допомогою LLMs, і чи можуть вони бути практично корисними для маркетологів та дослідників, які прагнуть зрозуміти споживацькі уподобання.

Дослідження починається з огляду поточного стану маркетингових методів. Автори наголошують, що традиційні методи, такі як конджойнт-аналіз, фокус-групи та інші опитування, хоч і є ефективними, вимагають значних ресурсів, як фінансових, так і часових. Вони підкреслюють, що великі мовні моделі, зокрема GPT, можуть стати альтернативою, забезпечуючи значну економію коштів і часу завдяки автоматизації частини дослідницького процесу. GPT, будучи навченою на величезних масивах текстових даних, які включають відгуки споживачів, обговорення на форумах і подібний контент, має потенціал імітувати людську поведінку в умовах маркетингових досліджень. Однак автори зазначають, що залишаються відкриті питання про те, наскільки узгодженою і корисною може бути така імітація в різних контекстах.

Емпірична частина дослідження складається з серії експериментів, які включають порівняння відповідей GPT із результатами опитувань, проведених серед реальних споживачів. Для цього автори організували п'ять окремих опитувань, у яких брали участь люди, котрі оцінювали різні товари, порівнюючи

їх за характеристиками, такими як бренд, ціна, наявність нових функцій тощо. Ці ж самі опитування було адаптовано для GPT через спеціально створені текстові запити, які дозволяли моделі генерувати відповіді, схожі на людські. Дослідники ставили перед собою завдання визначити, чи можуть відповіді GPT узгоджуватися з людськими відповідями як за змістом, так і за структурою.

Результати дослідження показали, що GPT здатен генерувати оцінки готовності платити (WTP), які є порівнянними з тими, що отримані у людських опитуваннях. Наприклад, у випадку дослідження уподобань зубної пасти GPT показав високу узгодженість із результатами, отриманими від реальних респондентів. Модель змогла точно визначити значення фтору як важливого атрибута продукту, і це узгоджувалося з оцінками людських респондентів. Водночас у деяких випадках GPT демонстрував відмінності, наприклад, у ставленні до брендів або до нових, експериментальних характеристик продуктів.

Важливим аспектом дослідження стало використання *fine-tuning* — додаткового налаштування GPT на основі даних із попередніх людських опитувань. Цей підхід суттєво покращив результати моделі, дозволивши їй краще імітувати людські відповіді навіть у випадках, коли йшлося про абсолютно нові характеристики продуктів. Наприклад, під час оцінки нових і незвичних смаків зубної пасти (огірок і млинець) *fine-tuning* дозволив GPT скоригувати свої оцінки, зробивши їх ближчими до реальних уподобань споживачів, які здебільшого негативно ставилися до таких нововведень. Аналогічно, для ноутбуків із вбудованим проектором *fine-tuning* дозволив моделі точніше оцінити попит на цю функцію, узгоджуючи результати з оцінками, отриманими в людських опитуваннях.

Попри позитивні результати, автори виявили кілька обмежень GPT. Однією з ключових проблем стало відображення різноманітності споживачів. Хоча модель успішно відтворювала середні значення WTP, їй важко було точно моделювати відмінності між різними демографічними групами, такими як дохід, стать або політичні погляди. Наприклад, модель демонструвала значні

відхилення в оцінках для окремих груп, і ці відхилення не завжди відповідали реальним даним. Крім того, GPT показав слабкі результати у випадках, коли йшлося про екстраполяцію на нові категорії продуктів, які не були частиною навчального набору моделі.

Дослідники також зосередили увагу на потенціалі GPT як інструмента для швидкого тестування ідей. У ситуаціях, коли час або ресурси обмежені, GPT може стати корисним доповненням до традиційних методів, дозволяючи маркетологам попередньо оцінювати попит на нові функції продуктів перед проведенням масштабних досліджень із реальними учасниками. Автори наголошують, що GPT не слід сприймати як повну заміну людських опитувань, але він може суттєво доповнити їх, особливо на ранніх етапах розробки маркетингових стратегій.

У висновках роботи зазначається, що подальше вдосконалення моделей, зокрема GPT-4 та інших сучасних LLMs, може суттєво розширити їхній потенціал у маркетингових дослідженнях. З огляду на швидкий розвиток цих технологій, автори прогнозують, що їхня здатність до аналізу поведінки споживачів буде лише зростати. Хоча наразі моделі демонструють змішані результати, їхня здатність швидко й економічно аналізувати великі обсяги даних робить їх перспективним інструментом для майбутніх маркетингових досліджень.

1.5 Особливості LLM для маркетингових задач

Великі мовні моделі, такі як GPT-4, LLaMA чи Claude, відкривають нові горизонти для вирішення завдань у сфері маркетингу. Їхня здатність аналізувати текстові дані, генерувати змістовні відповіді та адаптуватися до запитів користувачів дає можливість виводити маркетингові процеси на якісно новий рівень. Однією з ключових переваг LLM є їхня гнучкість у роботі з контентом: вони можуть писати рекламні тексти, аналізувати ринкові тренди, створювати

персоналізовані пропозиції та навіть автоматизувати спілкування з клієнтами через чатботи.

Унікальність цих моделей полягає у здатності до контекстного розуміння. Звичайні алгоритми для створення тексту часто покладаються на шаблони або заздалегідь визначені правила, що обмежує їхню адаптивність. LLM, навпаки, здатні враховувати різноманітні аспекти, такі як стиль, тон, настрій чи навіть культурні відмінності, створюючи контент, який максимально відповідає потребам цільової аудиторії. У маркетингових задачах це стає особливо важливим, адже індивідуальний підхід до кожного клієнта — одна з найцінніших складових успішної стратегії.

Інша значуща особливість LLM — їх здатність до швидкої адаптації й навчання на основі нових даних. У динамічному середовищі, де маркетингові тренди можуть змінюватися буквально щотижня, ця властивість дозволяє брендам залишатися актуальними. Модель можна налаштувати на аналіз соціальних мереж чи інших джерел даних, що дає змогу виявляти найсвіжіші інтереси аудиторії, теми для обговорення та можливості для взаємодії з клієнтами.

Одним із найяскравіших напрямів застосування LLM є автоматизація спілкування з клієнтами. Чатботи, створені на основі таких моделей, здатні підтримувати природний і персоналізований діалог. Наприклад, вони можуть оперативно відповідати на запити, рекомендувати продукти, вирішувати проблеми клієнтів або навіть брати участь у крос-продажах, пропонуючи супутні товари. Більше того, такі системи можуть працювати безперервно, обробляючи тисячі запитів одночасно, що значно підвищує ефективність обслуговування клієнтів.

Також важливою особливістю є здатність до аналізу великих обсягів текстових даних. LLM можна використовувати для автоматизації обробки відгуків клієнтів, визначення їхнього настрою або аналізу ключових запитів. Це дозволяє отримувати цінну інформацію для вдосконалення продуктів чи сервісів

і точніше орієнтувати маркетингові кампанії. Наприклад, аналіз коментарів у соціальних мережах або відгуків на торгових платформах може дати уявлення про поточні проблеми та переваги, що допомагає швидше реагувати на потреби клієнтів.

Крім того, LLM пропонують інтеграцію з іншими технологіями, такими як машинне навчання або аналітичні інструменти. Наприклад, модель може бути частиною комплексної системи, яка одночасно аналізує поведінку користувачів, прогнозує їхні дії та пропонує персоналізовані маркетингові стратегії. Це значно розширює можливості бізнесу, дозволяючи створювати складніші та ефективніші системи управління клієнтським досвідом.

LLM також можуть допомогти у творчих аспектах маркетингу. Генерація текстів для реклами, сценаріїв для відеороликів, електронних листів або навіть унікальних повідомлень для соціальних мереж — усе це знаходиться у межах можливостей моделей. Завдяки цьому можна не лише оптимізувати час, але й забезпечити більшу різноманітність ідей.

1.6 Постановка задачі

Сучасний розвиток технологій відкриває нові можливості для інтеграції штучного інтелекту в ігрову індустрію, що сприяє створенню унікального ігрового досвіду. У рамках цього проєкту ставиться завдання розробити 2D-відеогру, яка поєднає елементи інтерактивного геймплею з можливостями великих мовних моделей (LLM). Гра має виступати не лише розважальним продуктом, а й демонстраційним середовищем для застосування сучасних технологій у вирішенні маркетингових задач, зокрема симуляції процесу продажів.

Основна ідея полягає у створенні гри, де гравець виконує роль комівояжера, завданням якого є продаж різноманітних товарів віртуальним персонажам. Унікальність гри забезпечується інтеграцією Llama 3, яка виступає

основою для поведінки NPC, та використанням голосового взаємодії через Vosk. Це дозволяє гравцю не лише керувати діалогами через текстові команди, але й розмовляти із персонажами, що додає реалістичності ігровому процесу.

Серед технічних вимог до гри слід виділити кілька ключових аспектів. По-перше, ігрове середовище має бути створене за допомогою Godot Engine, що забезпечить високу продуктивність та гнучкість під час розробки. По-друге, інтеграція Llama 3 повинна дозволити кожному NPC мати власний набір характеристик, таких як особистість, настрої, уподобання та реакції на різні пропозиції. Це має створити багатогранний ігровий досвід, адже гравець буде змушений адаптувати свої стратегії продажів залежно від поведінки клієнтів.

Особливе місце займає система голосової взаємодії, що базується на технології Vosk. Вона повинна забезпечити перетворення голосу гравця на текст для подальшої обробки мовною моделлю. Це дозволить спілкуватися з NPC у природній манері, що важливо для занурення в ігровий процес. Також необхідно передбачити можливість коректного розпізнавання голосу навіть у складних умовах, таких як швидка мова або неідеальна вимова.

Задача також включає створення збалансованого ігрового процесу. Продаж товарів має бути не просто діалоговим симулятором, а включати елементи стратегії: гравець повинен враховувати обмеженість ресурсів, вибірковість NPC та необхідність досягнення певних цілей для просування у грі. Наприклад, кожен NPC може мати різні переваги щодо товарів, що стимулює гравця до експериментів із комунікацією та продажами.

Ключовим завданням є забезпечення плавної інтеграції всіх компонентів гри: графічного інтерфейсу, логіки ігрового процесу, мовної моделі та голосового управління. Важливою частиною реалізації є тестування системи, щоб перевірити, чи NPC дійсно взаємодіють із гравцем у реалістичній манері, а голосове розпізнавання відповідає очікуваним стандартам.

У результаті проєкт має представити комплексне рішення, яке покаже, як великі мовні моделі можуть бути використані в інтерактивних медіа для

розв'язання задач маркетингової взаємодії. Гра стане не лише розважальним продуктом, а й демонстрацією можливостей новітніх технологій у створенні унікальних користувацьких досвідів.

РОЗДІЛ 2.

МЕТОДИ РОЗРОБКИ КОМП'ЮТЕРНОЇ ГРИ ЯК ІНТЕРФЕЙСУ ВЗАЄМОДІЇ З LLM

2.1. Методи розробки інформаційних технологій

Розробка комп'ютерних ігор є складним і багатогранним процесом, який поєднує технологічні, творчі та управлінські аспекти. Успішна реалізація ігрового проєкту вимагає не лише технічної експертизи, але й ефективної організації роботи команди, правильного вибору інструментів та ретельного планування. Залежно від масштабу і специфіки гри, використовуються різні методи, які варіюються від класичних підходів до сучасних інноваційних практик.

Одним із найважливіших етапів розробки є визначення ігрової концепції. Це процес формування ключової ідеї, навколо якої будується весь проєкт. Концепція визначає жанр, сюжет, механіку гри та її аудиторію. На цьому етапі розробники часто використовують методи брейнштормінгу для генерації ідей, створюють короткі документальні описи (game concept document) і готують прототипи для оцінки життєздатності концепції. Прототипи можуть бути як паперовими, так і цифровими, залежно від складності ідеї.

Ще одним важливим аспектом є вибір моделі розробки. Найпоширенішими є каскадна модель ігрової розробки (Waterfall) та гнучка модель (Agile). Каскадна модель передбачає покрокове виконання роботи: від проєктування до тестування та випуску гри. Це підхід, що краще підходить для проєктів зі статичними вимогами. Гнучка розробка, натомість, дозволяє проводити роботу ітеративно, в коротких циклах (спринтах), і адаптувати проєкт до змінних умов чи нових ідей. Гнучкі методології, такі як Scrum або Kanban, особливо популярні серед невеликих команд або проєктів з високим рівнем інновацій.

Важливим етапом у розробці є побудова архітектури гри. Це включає визначення основних компонентів, таких як ігровий двигун, графічний та фізичний модулі, система управління даними, а також створення механізмів взаємодії з користувачем. Вибір ігрового двигуна (наприклад, Godot, Unity або Unreal Engine) залежить від вимог проєкту, таких як тип графіки, необхідний рівень деталізації, платформи, на які розробляється гра, та бюджет команди. Godot, наприклад, відомий своєю простотою у використанні, відкритим кодом і гнучкістю для 2D-ігор, що робить його ідеальним вибором для цього проєкту.

Одним із методів забезпечення якісного ігрового досвіду є часте тестування. Ітеративна розробка передбачає створення ранніх версій гри для перевірки її функціоналу, виявлення проблем у механіках чи дизайні. Тестування може проводитися на рівні внутрішньої команди або залученням зовнішніх тестувальників. Крім того, автоматизовані інструменти дозволяють перевіряти технічну частину гри, включаючи продуктивність, стабільність і відсутність помилок у коді.

Крім технічної розробки, суттєвим методом є створення дизайну гри. Геймдизайн формує те, як гравець взаємодіє із середовищем, і охоплює такі аспекти, як створення рівнів, баланс ігрової механіки, розробка персонажів і системи прогресу. Творчі підходи до дизайну можуть включати методи створення нарративів, розробку нелінійних сюжетів або навіть впровадження процедурної генерації контенту, що збільшує варіативність гри.

Значну роль у розробці сучасних ігор відіграє інтеграція новітніх технологій. Наприклад, використання великих мовних моделей, як у цьому проєкті, дозволяє додати унікальну складову до гри — інтелектуальну взаємодію з персонажами. Це створює нові вимоги до архітектури гри, адже необхідно забезпечити безперебійний зв'язок між ігровим середовищем і мовною моделлю, враховуючи затримки обчислень, адаптацію відповідей та інтеграцію голосових технологій.

2.1.1 Інтеграція з LLM. У нашому проєкті великі мовні моделі використовуються для створення інтерактивного ігрового досвіду, що адаптується до поведінки та вибору гравців у реальному часі. Основна ідея полягає у застосуванні можливостей LLM для генерації текстів, діалогів і сценаріїв, які формуються залежно від дій користувачів. Це дозволяє забезпечити динамічність і варіативність ігрового процесу, що відповідає сучасним вимогам до інтерактивних маркетингових ігор.

LLM, такі як LLaMA 3, відіграють ключову роль у моделюванні поведінки персонажів та створенні сценаріїв. Наприклад, кожна взаємодія гравця з ігровим світом, включно з діалогами з персонажами, будується на основі аналізу текстового введення користувача. Модель генерує відповіді, які враховують попередній контекст, загальний стиль гри та навіть особливості конкретного сценарію. Це робить взаємодію більш персоналізованою, дозволяючи кожному користувачеві відчувати себе частиною унікальної історії.

Окрему увагу в проєкті приділено використанню LLM для динамічної генерації контенту. Наприклад, гра може автоматично створювати нові завдання чи варіативні сюжетні розв'язки залежно від прогресу гравця. Це реалізується шляхом інтеграції моделі, яка аналізує поточну ситуацію в грі, враховує параметри гравця (наприклад, його рішення чи успіхи) і пропонує нові ігрові елементи, що відповідають обраному стилю гри.

Іншим важливим напрямком є використання LLM для створення багаторівневих систем взаємодії. Наприклад, у нашому проєкті гравець може задавати текстові запитання або формулювати команди, які мовна модель інтерпретує та використовує для зміни стану ігрового світу. Це створює ефект прямого впливу користувача на ігровий процес, роблячи гру більш інтерактивною та цікавою.

Особливістю нашого підходу є також налаштування LLM для роботи з унікальним контекстом гри. Для цього використовуються спеціалізовані датасети, які навчають модель розуміти стиль мовлення персонажів, логіку

ігрового світу та цільову аудиторію гри. Це дозволяє адаптувати модель під конкретний проєкт, забезпечуючи точність і якість взаємодії.

2.1.2 Розробка контенту. Розробка контенту є одним із найбільш важливих етапів створення відеоігри, оскільки від неї залежить не лише візуальна складова, а й загальна атмосфера, а також досвід, який гравець отримує під час гри. Контент включає в себе всі елементи гри, що не стосуються чисто технічної частини: персонажів, діалоги, місця, об'єкти, анімацію, музику та багато іншого. В контексті ігор з інтерактивними сюжетами та великим елементом діалогів контент займає центральне місце, оскільки він безпосередньо впливає на те, як гравець взаємодіє з ігровим світом.

У сучасних іграх контент створюється з урахуванням кількох аспектів. Це можуть бути як статичні елементи, такі як графіка та текстури, так і динамічні елементи, включаючи алгоритми штучного інтелекту, що визначають поведінку персонажів, або навіть можливість інтерактивного створення контенту гравцем. Відеоігри можуть містити як заздалегідь створені сценарії, так і генеровані в реальному часі, що забезпечує більшу гнучкість у створенні досвіду для кожного користувача.

У випадку гри з LLM чатботом-покупцем контент зводиться здебільшого до двох аспектів. Перший — це програмний контент, що включає код для роботи з великими мовними моделями, визначення їхньої поведінки і інтеграцію у геймплей. Це може включати налаштування сценаріїв, які визначають, як і в яких ситуаціях чатбот буде взаємодіяти з гравцем, а також розробку системи відповіді чатбота на запити гравця. Другий аспект — це контент, що стосується створення самих NPC, їхніх особливостей і поведінки під час взаємодії з покупцями.

Цей контент передбачає не лише програмування поведінки чатбота, але й наповнення його "персонажної" частини: створення історій, які цей чатбот може розповідати, або визначення його унікальних рис, таких як специфічний стиль спілкування або певні торгові переваги. Оскільки майже весь контент гри

побудований навколо цієї взаємодії, важливо, щоб він був максимально детальним та адаптивним.

Враховуючи важливість діалогів, контент також включає в себе текстові матеріали, які генеруються LLM або надаються в рамках заздалегідь заданих параметрів. Це може бути також механіка формування запитів, обробка яких дозволяє чатботу допомогти гравцеві в продажу товару, а також забезпечення плавності і реалістичності цих взаємодій.

В іграх, що включають таку інтерактивну складову, розробка контенту може бути частково автоматизована. Наприклад, використання великих мовних моделей дозволяє генерувати величезну кількість контенту без потреби в ручному написанні кожного діалогу. Це не лише економить час, але й дає можливість створювати більш різноманітні і багаті на контекст сценарії взаємодії, що важливо для збереження цікавості гравця.

Водночас, навіть у разі застосування автоматизованих технологій, розробка контенту потребує значних зусиль для налаштування і тестування поведінки моделей, розробки систем для моніторингу якості контенту та забезпечення того, щоб його частини логічно і органічно вписувались у загальний сюжет ігрового процесу.

2.1.3 Розробка ігрової механіки. Ігрова механіка є основою будь-якої відеоігри, оскільки вона визначає, як гравець взаємодіє з ігровим світом та які елементи цього світу можна контролювати. Вона включає в себе всі основні системи, правила і процеси, які впливають на геймплей, і є тим, що забезпечує унікальність ігрового досвіду. Розробка механіки завжди має починатися з розуміння того, яку ціль має гра, який тип досвіду хоче отримати гравець і як механіка буде підтримувати цей досвід.

Ключовим аспектом геймдизайну є баланс між складністю та доступністю гри. Задача розробника — створити систему, яка буде достатньо складною, щоб підтримувати інтерес гравця, але водночас досить зрозумілою, щоб не відлякати

новачків. Це означає, що ігрові механіки повинні бути адаптивними, дозволяючи гравцеві легко опанувати основи, але при цьому пропонувати достатньо глибини для тих, хто хоче зануритися у гру більше.

Механіки можуть бути різноманітними і охоплювати багато аспектів, від бою до управління ресурсами, від навігації до соціальних взаємодій. У контексті гри, яка інтегрує чатбота на основі великої мовної моделі (LLM), важливо звернути увагу на механіки, пов'язані з діалогами та взаємодією гравця з NPC. Це може бути не тільки звичне введення тексту, але й використання голосових команд, що дозволяє гравцю ще більш інтуїтивно взаємодіяти з ігровим світом. Важливо, щоб система діалогів була побудована таким чином, щоб вона доповнювала основні механіки гри, а не ставала окремою, ізольованою частиною.

Важливим елементом геймдизайну є концепція зворотного зв'язку. Гравець повинен отримувати негайний відгук на свої дії, щоб розуміти, чи правильно він діє. Цей зворотний зв'язок може бути візуальним, звуковим чи навіть текстовим, якщо мова йде про взаємодію з чатботом. Так, наприклад, якщо гравець продає товар NPC, то він має отримати чіткий сигнал про успіх чи невдачу цієї операції: чи був товар прийнятий, чи його не прийняли через певні умови, наприклад, через ціну або недостатній попит. Це створює відчуття реалістичності й збагачує геймплей, що в свою чергу дозволяє гравцеві більше занурюватися у світ гри.

Крім того, важливо враховувати концепцію прогресії в ігровій механіці. Прогресія визначає, як змінюється стан гри, чи то через покращення персонажа, відкриття нових рівнів чи розширення можливостей гравця. У випадку з чатботами важливо продумати систему, яка дозволяє гравцеві покращувати свої взаємодії з NPC або збільшувати ефективність своїх торгових операцій. Це може бути механіка розвитку навичок або досягнень, яка підвищує здатність гравця впливати на ціни, запаси товарів чи інші важливі фактори в грі.

Ігрова механіка також може включати складніші аспекти, такі як стратегічне планування чи взаємодія з іншими гравцями в багатокористувацькому режимі, проте в контексті гри, орієнтованої на однокористувацький досвід, більше уваги варто приділити створенню цілісної й логічно взаємопов'язаної системи, де кожна дія гравця має зрозумілий і передбачуваний результат. У випадку гри з чатботом ця система повинна бути гнучкою, даючи можливість варіювати типи взаємодій залежно від обставин, вибору гравця або розвитку його персонажа.

Не можна забувати й про механіку побудови світу, яка забезпечує унікальність кожного ігрового середовища. Оскільки у вашій грі важливим є фактор взаємодії з NPC та покупцями, це означає, що навколишнє середовище, в якому відбуваються ці операції, повинно бути детально продумане та відповідати контексту гри. Навіть якщо основна механіка зосереджена на діалогах і торгівлі, вона повинна гармонійно вписуватися в загальний ландшафт і світ гри.

Теорії геймдизайну постійно еволюціонують, і що сьогодні є ефективною механікою, завтра може виявитися менш актуальним. Важливо стежити за новими тенденціями та інноваціями у галузі, зокрема, за використанням новітніх технологій для інтеграції великих мовних моделей у геймплей, і намагатися впроваджувати їх так, щоб механіка гри залишалася зрозумілою, привабливою і насиченою.

2.1.4 Тестування. Тестування є невід'ємною частиною процесу розробки будь-якої комп'ютерної гри, оскільки воно забезпечує перевірку якості гри, її стабільності та відповідності вимогам. Процес тестування охоплює різноманітні етапи, від виявлення помилок у коді до оцінки ігрового процесу з точки зору користувача. Тестування не лише допомагає виявити помилки, але й дає змогу підтвердити, що всі ігрові механіки працюють так, як задумано, і що гравець отримує належний досвід.

Одним із найбільш поширених методів тестування є плейтестинг, або тестування ігри кінцевими користувачами. Плейтест дозволяє розробникам оцінити, як гравці взаємодіють з грою, чи знаходять вони її зрозумілою, чи виникають у них труднощі при виконанні різних завдань чи взаємодії з ігровими системами. В рамках цього процесу гравці можуть тестувати різні аспекти гри, включаючи бойову механіку, систему діалогів, інтерфейс користувача, а також загальний геймплей. Плейтестинг може проводитися на різних етапах розробки, і його мета — отримати зворотний зв'язок для виявлення можливих проблем, які не були помічені під час внутрішнього тестування.

Дебагінг, або відловлювання помилок у коді, є ще одним важливим етапом тестування. Кожен розробник знає, як важливо виправляти баги до випуску фінальної версії гри, оскільки помилки можуть негативно впливати на ігровий процес і сприйняття гри гравцями. Дебагінг включає в себе процес пошуку та виправлення помилок у коді, що можуть призвести до неправильного функціонування гри, візуальних чи звукових глюків, а також збоїв чи падінь програми. Для цього використовуються різні інструменти, які дозволяють відслідковувати проблеми на різних етапах виконання коду, і допомагають ефективно усувати баги.

Тестування також включає в себе регресійне тестування, яке перевіряє, чи не було зіпсовано жодного з раніше працюючих елементів гри після внесення змін або виправлень. Це важливо, адже іноді під час вирішення однієї проблеми можна випадково створити іншу, що може викликати нові помилки або несправності в ігровій механіці. Регресійне тестування допомагає виявити такі проблеми ще до того, як вони вплинуть на кінцевого користувача.

Окрім того, важливим є тестування продуктивності, яке перевіряє, як гра працює на різних платформах, чи не виникають збої через надмірне навантаження на систему. У сучасних іграх особливо важливо тестувати стабільність ігрового процесу, оскільки навіть невеликі затримки або лаги можуть негативно вплинути на досвід гравця. Продуктивність тестується через

вимірювання фреймрейтів, часу завантаження сцен і ефективності ресурсів, таких як пам'ять і процесор. Це дозволяє оптимізувати гру та зробити її доступною для широкої аудиторії, незалежно від технічних можливостей їхніх пристроїв.

Важливою складовою є також тестування сумісності, яке перевіряє, чи працює гра на різних операційних системах, пристроях, чи немає проблем з підтримкою певних апаратних засобів. Це дозволяє уникнути ситуацій, коли гра працює коректно тільки на одному типі пристрою чи ОС, а на інших — ні. Для гри, яка інтегрує великі мовні моделі, тестування також має включати перевірку взаємодії між моделюванням мови і іншими системами гри, щоб уникнути проблем із затримками в обробці запитів або некоректною поведінкою чатбота.

Тестування взаємодії між компонентами гри є важливим, оскільки в сучасних проектах задіяно багато складних систем, таких як штучний інтелект, голосові технології, інтеграція з LLM чатботами, і всі вони повинні працювати в комплексі. Перевірка цих компонентів дозволяє забезпечити плавну взаємодію між системами та уникнути конфліктів, що можуть вплинути на досвід гравця. Зокрема, тестування функціональності чатботів, які забезпечують взаємодію між гравцем та NPC, має включати перевірку на відповідність кожного запиту та реакцій на нього, а також перевірку на стійкість до різних типів введення — як голосового, так і текстового.

2.2 Вибір ігрового рушія

Розробка комп'ютерних ігор неможлива без вибору рушія, який забезпечить основу для всіх аспектів гри — від графіки до логіки гри та взаємодії з користувачем. Ігрові рушії є потужними програмними платформами, що спрощують процес розробки, надаючи готові інструменти для створення ігор. Вибір рушія залежить від багатьох факторів, таких як тип гри, цільова аудиторія, ресурси команди, а також специфікацій проекту. Розглянемо кілька популярних

ігрових рушіїв, їхні переваги та недоліки, а потім зупинимося на рушії, який був вибраний для даного проекту.

Одним з найбільш популярних рушіїв є Unity. Unity пропонує широкий спектр інструментів для розробки 2D та 3D ігор і підтримує великі платформи, такі як Windows, macOS, Linux, а також мобільні пристрої та консолі. Його основною перевагою є потужна підтримка для розробки візуальних ефектів, фізики, а також масштабованість проектів. Unity використовує мову C#, що є популярною серед програмістів завдяки її високій швидкості та широкій підтримці. Однак одним із основних мінусів Unity є потреба у значних комп'ютерних ресурсах, особливо при створенні великих ігрових світів з високою деталізацією. Цей фактор стає критичним, коли проект вимагає інтеграції складних моделей штучного інтелекту, зокрема для роботи з LLM, що вимагають великих обчислювальних потужностей.

Unreal Engine — ще один популярний рушії, що широко використовується для створення високоякісних 3D ігор. Unreal має багато переваг, включаючи потужні інструменти для рендерингу, розширену фізику та підтримку великих відкритих світів. Він також використовує C++, що дає більший контроль над продуктивністю та можливістю оптимізації. Одним із значних недоліків Unreal Engine є його складність для новачків і досить висока вимога до ресурсів, що може стати проблемою, якщо потрібна оптимізація для роботи з LLM без великої кількості зовнішніх ресурсів.

CryEngine є ще одним сильним рушієм, який пропонує надзвичайно детальну графіку і підтримку складних візуальних ефектів, таких як реалістичні водні поверхні та освітлення. Він чудово підходить для створення ігор з фокусом на візуальні аспекти. Однак CryEngine менш популярний у порівнянні з Unity або Unreal і має менше підтримки від спільноти. Крім того, його вартість ліцензії та складність у використанні можуть бути недоліками для невеликих команд.

Враховуючи всі ці фактори, для цього проекту було обрано Godot Engine, який поєднує в собі багато переваг при значно меншій вимозі до ресурсів. Godot

є відкритим рушієм, що дозволяє будь-якому розробнику змінювати його код під свої потреби. Це важливо для проектів, де потрібно гнучко налаштовувати функціональність та інтегрувати специфічні рішення, такі як підтримка LLM. Однією з основних переваг Godot є його підтримка двох мов програмування — GDScript (яка є дуже схожою на Python) і C#, що дозволяє розробникам вибирати той варіант, який їм найбільше підходить.

Godot також виділяється своєю низькою вимогою до ресурсів. Оскільки для цього проекту важливим є використання локальних потужностей для запуску LLM, Godot є чудовим вибором завдяки своїй економії на апаратних ресурсах. Це дає можливість зосередитись на інтеграції чатботів і моделей штучного інтелекту без необхідності турбуватись про високі вимоги до продуктивності, які мають інші рушії.

Ще однією перевагою Godot є його гнучкість для розробки 2D ігор. Оскільки проект передбачає 2D-графіку, Godot надає безліч інструментів для створення анімацій, обробки фізики та контролю за рухом об'єктів на екрані. Можливості для створення складних ігрових механік, таких як інтерактивність з чатботами, також можна ефективно реалізувати за допомогою Godot.

Особливо варто зазначити, що Godot має активно розвиваючуся спільноту та безліч документації, що значно полегшує навчання та впровадження нових ідей у процес розробки. Він також надає великий набір інструментів для тестування, відлагодження і оптимізації гри, що дозволяє скоротити час розробки і знизити витрати на технічну підтримку.

Таким чином, Godot Engine є оптимальним вибором для проекту, де важливо зберегти баланс між графічними можливостями, низькими вимогами до ресурсів та високою гнучкістю для інтеграції штучного інтелекту, таких як LLM чатботи.

РОЗДІЛ 3.

РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ ЯК ЧАТ-БОТ ІНТЕРФЕЙСУ

3.1 Проектування модулів гри

Розробка гри починається з ретельного процесу концептуалізації, що є одним із найважливіших етапів у створенні будь-якого програмного продукту. Для даного проекту цей процес був орієнтований на досягнення кількох ключових цілей: мінімізація навантаження на ресурси комп'ютерної системи, забезпечення плавності роботи з LLM-моделлю та створення візуальної стилістики, яка б відповідала задуму гри. Зважаючи на обмеження системних ресурсів, зокрема обчислювальних потужностей графічної карти, було прийнято рішення використовувати ретро-стиль візуальної презентації. Такий підхід дозволяє не лише знизити графічне навантаження, але й створює унікальну естетику гри, що апелює до ностальгії гравців. Як результат, основою графічного стилю стала піксельна 2D графіка. Це рішення є оптимальним, оскільки воно не лише знижує вимоги до апаратного забезпечення, але й дозволяє зберігати максимальну свободу для роботи моделі LLM. У процесі взаємодії з ботом система зможе ефективно обробляти тривалі діалоги без ризику зниження швидкості роботи чи погіршення складності діалогових сценаріїв. Гравець буде взаємодіяти із грою за допомогою комбінації клавіатури та миші. Миша забезпечить інтуїтивний доступ до ігрових інтерфейсів, таких як меню чи діалогові вікна, тоді як клавіатура використовується для навігації ігровим світом.

Гра починається з головного меню, яке є першим екраном, що зустрічає гравця. У цьому меню гравцеві пропонуються три основні опції: почати нову гру, відкрити налаштування або завершити гру. При початку нової гри гравець переходить до екрану отримання предметів. Цей етап додає елемент випадковості та різноманітності в кожну нову гру. Чотири предмети, які видаються гравцеві випадковим чином, стають його інвентарем для продажу.

Після цього гравець знайомиться зі своїм ігровим персонажем, якого він буде контролювати протягом усієї гри. Персонаж має змогу пересуватися по віртуальній вулиці, яка є основною локацією гри, та взаємодіяти з входами до будівель. Кожна будівля приховує неігрових персонажів (NPC), з якими гравець може вступати в діалог для спроби продажу своїх товарів.

Діалогова система є ключовою механікою гри. Вона реалізована у вигляді чат-інтерфейсу, який дозволяє гравцеві вести текстові бесіди з NPC. У цих бесідах гравець намагається переконати NPC купити той чи інший предмет зі свого інвентаря. Крім прямої взаємодії через текст, гравець має дві додаткові опції: "дати предмет" і "вийти з розмови". Опція "дати предмет" є необхідною для завершення продажу та працює на основі текстової валідації коду. Гра автоматично перевіряє, чи відповідає переданий предмет запитам NPC, визначеним моделлю LLM.

Опція "вийти з розмови" забезпечує гнучкість і зручність у взаємодії. Це особливо корисно у випадках, коли модель LLM стикається із труднощами під час генерації тексту, наприклад, через технічні помилки чи перевантаження системи. У таких випадках гравець може вручну завершити діалог, не чекаючи автоматичного відновлення роботи моделі. Також через комплексність валідації тексту для закінчення діалогу, ця кнопка призначена як єдиний метод покинути розмову.

3.1.1 Базові модулі гри. Для забезпечення зручності та оперативності на початкових етапах розробки, а також для створення відчуття швидкого прогресу, яке є важливим для підтримання мотивації, робота над грою була розпочата з головного меню. Головне меню є центральним елементом будь-якої гри, через який користувач отримує доступ до основних функцій. У даному проекті це меню є типовим представником свого жанру, виконаним у мінімалістичному стилі. У його центрі розташований список кнопок, кожна з яких виконує певну функцію: перехід до окремого підменю або початок нової гри.

Оскільки задум гри не передбачає тривалого проходження, а розрахований на короткотривалу взаємодію за один захід, система збереження прогресу була визнана непотрібною. Це рішення є також технологічно обґрунтованим, адже використання системи збереження могло б призвести до несумісності з LLM. Для роботи моделі потрібні значні ресурси, і додаткові витрати на обробку даних збереження могли б негативно вплинути на загальну продуктивність, що суперечило б основним цілям проекту.

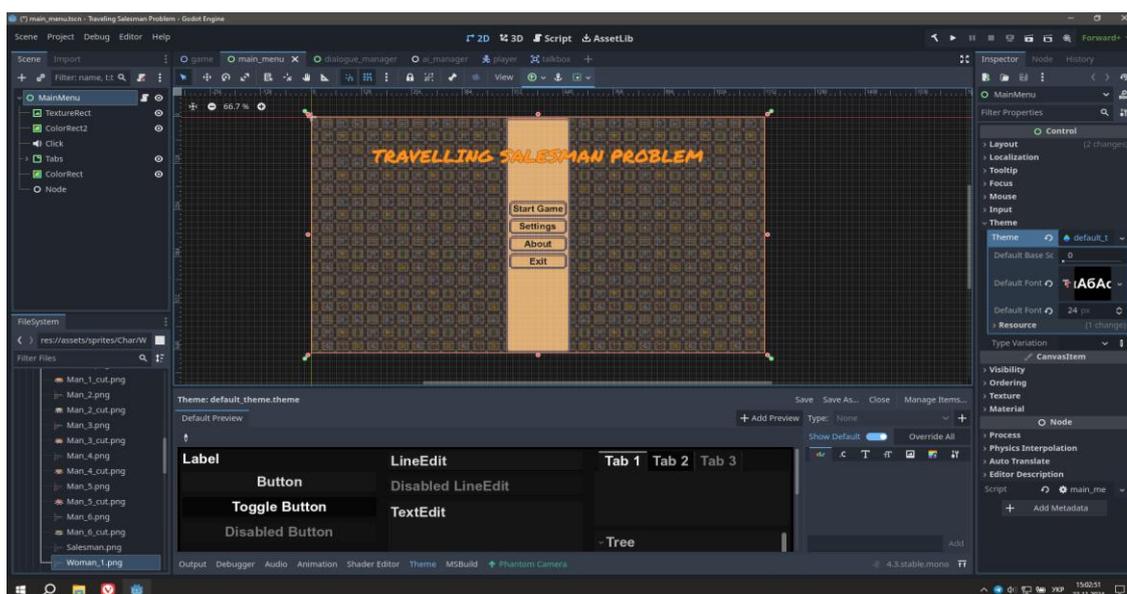


Рисунок 1.1 - Вигляд головного меню у візуальному редакторі рушія

Після створення головного меню, розробка перейшла до створення ключових компонентів, необхідних для гри. Було визначено два основних компоненти: `Customer` і `ItemDefinition`. Їхнє призначення — забезпечити збереження та обробку даних, пов'язаних із покупцями та предметами. Компонент `Customer` містить дані про персонажів-покупців, такі як їхні уподобання, особистісні характеристики та настрої. `ItemDefinition`, своєю чергою, відповідає за зберігання інформації про предмети, які гравець може продавати: назви, описи, унікальні властивості тощо. Код цих компонентів можна знайти в додатках, що додаються до звіту.

На початку гри гравцеві необхідно видати чотири предмети зі списку, створеного на основі компонентів `ItemDefinition`. Це завдання, хоч і нескладне у

реалізації, є важливим елементом, оскільки створює основу для подальшої взаємодії гравця зі світом гри. Екран видачі предметів оформлено у вигляді чорного напівпрозорого полотна, на якому відображаються чотири предмети та їхні назви. Протягом декількох секунд ці предмети змінюються випадковим чином, і лише після цього зупиняються на остаточному варіанті. Таким чином, створюється відчуття інтриги та випадковості, що додає грі динамічності та елементу несподіванки.

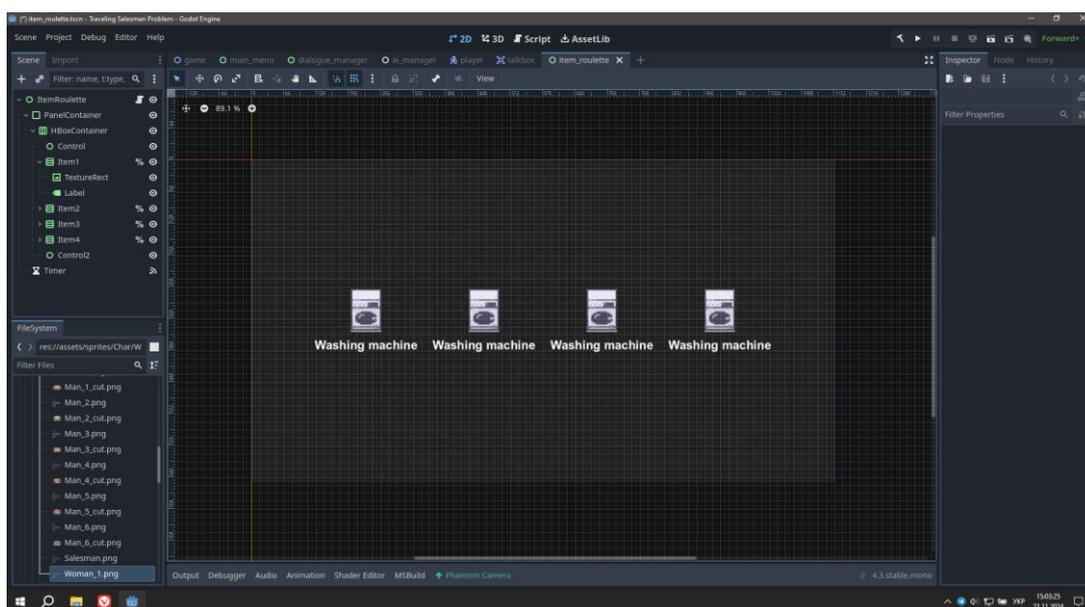


Рисунок 1.2 - Екран видачі випадкових предметів у візуальному редакторі рушія

Після реалізації цього функціоналу постала більш складна задача — створення світу гри та ігрового персонажа, який буде ним пересуватися. Розпочато з роботи над персонажем гравця. Це об'єкт типу `CharacterBody2D`, що використовує модифікований стандартний код для пересування. Для визначення зон, із якими можна взаємодіяти, використано компонент `Raycast2D`. Такий підхід дозволяє забезпечити точну та швидку реакцію на взаємодії гравця зі світом.

Сам ігровий світ виконано за допомогою `TileMapLayer`, що дозволяє створювати детальну карту за мінімальних витрат ресурсів. Для графічного оформлення були використані заздалегідь підготовлені художні ресурси, які надають ігровому світу ретро-естетику. Щоб уникнути проходження персонажа

крізь стіни, на карту було додано об'єкти `StaticBody2D`, які виступають колізійними бар'єрами. У результаті карта набуває чітко визначених меж і фізичних властивостей.

Карта гри являє собою невелику вулицю, на якій розташовані кілька будівель. Будівлі є важливими елементами, адже вони слугують точками взаємодії з NPC. Для реалізації цієї взаємодії на входах до будівель використано компонент `Area2D`, у який додатково вбудовані посилання на заздалегідь створених покупців. Коли гравець наближається до входу та ініціює взаємодію, дані про відповідного покупця передаються до основного компонента, що відповідає за обробку інформації у грі. Це дозволяє забезпечити безперервний ігровий процес та ефективну роботу діалогової системи.

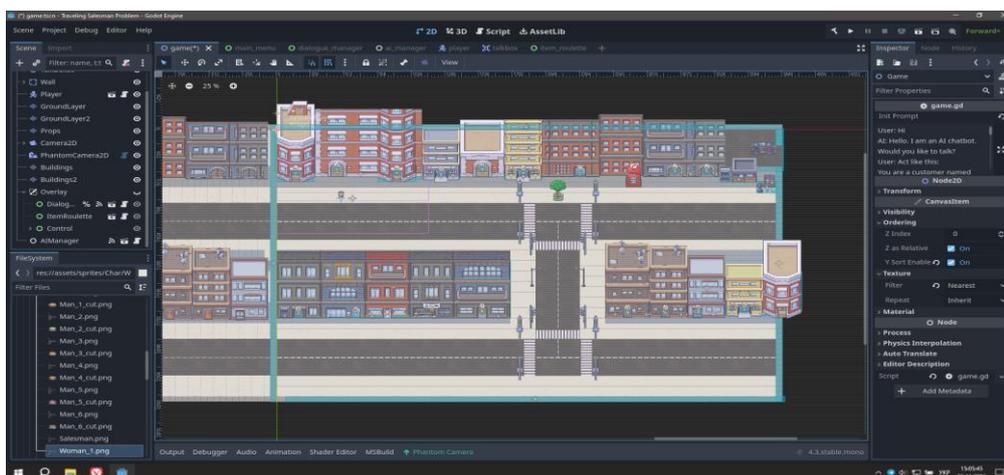


Рисунок 1.3 - Карта гри у візуальному редакторі рушія

Після створення та тестування базових компонентів гри, таких як головне меню, система розподілу предметів, персонаж гравця та ігровий світ, розпочалася робота над наступними елементами проекту, які мають ще більше розширити функціонал ігрової системи.

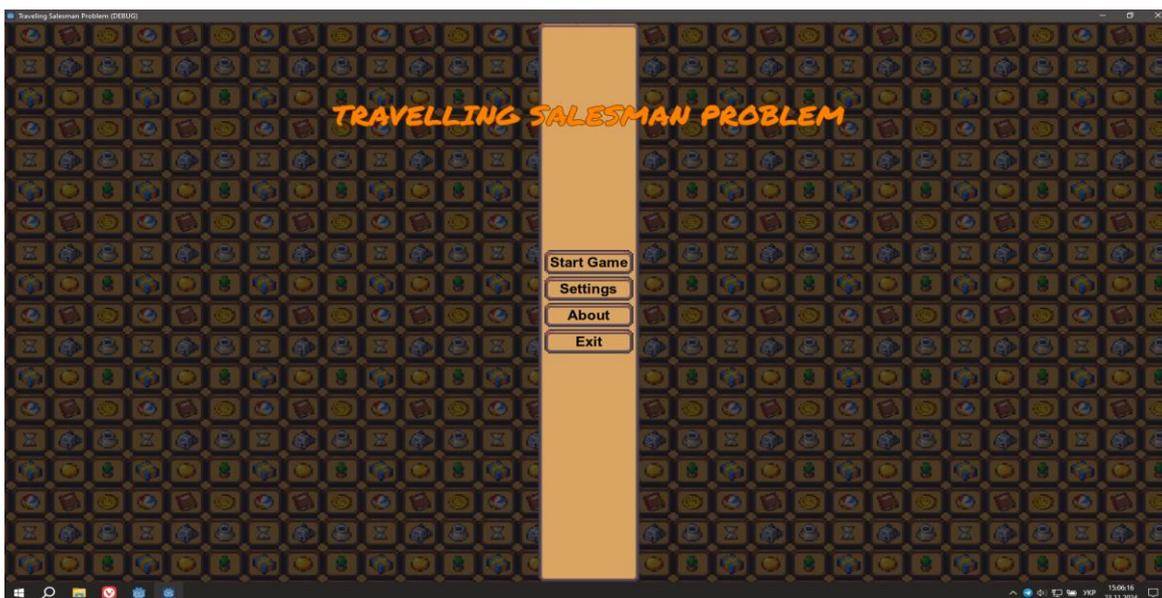


Рисунок 1.4 - Скриншот головного меню гри



Рисунок 1.5 - Скриншот екрану видачі предметів



Рисунок 1.6 - Гравець на території карти гри

3.1.2 Вікно чату. Вікно чату є, без перебільшення, центральним елементом гри, адже саме через нього гравець взаємодіє з головною механікою та сутністю ігрового процесу — діалогом із чат-ботом. Це не просто один із функціональних елементів, а ключовий вузол, який визначає унікальність ігрового досвіду. Від його зручності, інтуїтивності та естетичності безпосередньо залежить, наскільки гравець буде залучений у процес, тому розробці цього елемента приділено особливу увагу.

Дизайн вікна чату виконано так, щоб воно було максимально інтуїтивним і простим у використанні, але водночас забезпечувало всі необхідні функції. Інтерфейс вікна чату має чіткий поділ на дві основні секції: ліву панель, що відповідає за інформацію про клієнта та основні кнопки взаємодії, і праву панель, де відображається чат-лог, поле для введення тексту, а також кнопка для голосової взаємодії.

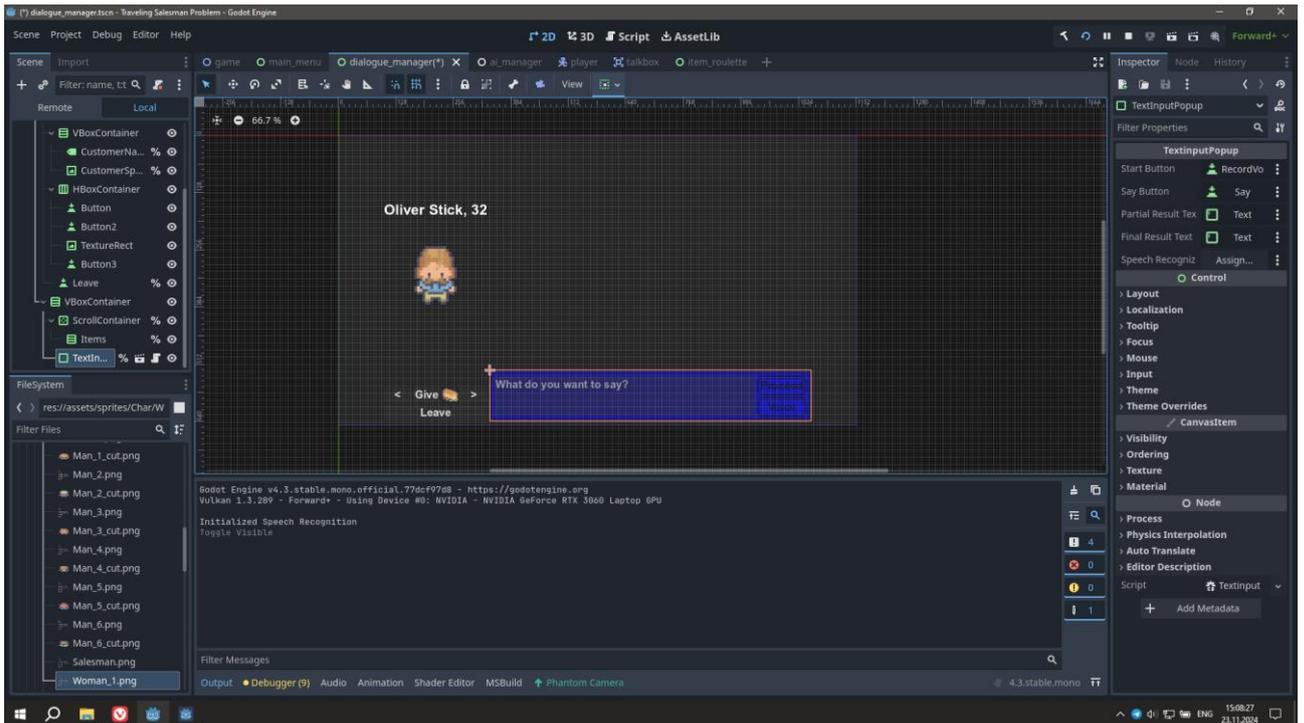


Рисунок 2.1 = Вікно чату у редакторі



Рисунок 2.2 - Вікно чату у грі

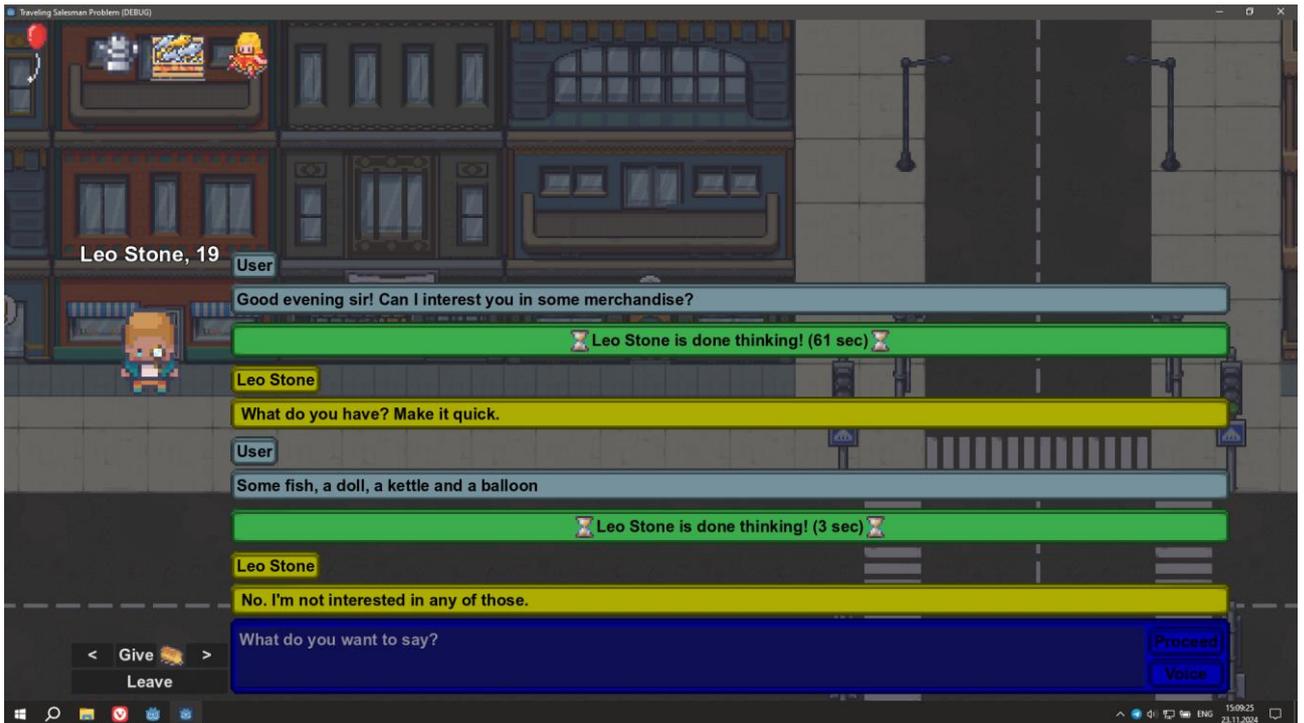


Рисунок 2.3 - Приклад діалогу з неігровим персонажем

На лівій стороні гравець може побачити кілька важливих елементів:

1. Ім'я та вік клієнта. Ці дані є базовими, щоб створити мінімальне уявлення про персонажа, з яким відбувається взаємодія. Хоча вони не впливають на функціонування бота, вони допомагають занурити гравця в атмосферу гри, додаючи персоналізації кожній зустрічі.

2. Вигляд клієнта. Це чисто візуальний елемент, який відображає портрет або стиль персонажа. Важливо зазначити, що вигляд персонажа не впливає на роботу чат-бота та не зберігається для подальшого використання, але додає грі візуальної виразності.

3. Кнопка видачі предмета. Її реалізовано у вигляді простого інтерфейсу, який дозволяє вибрати предмет, що буде переданий покупцеві. Гравець може легко перемикаати доступні предмети за допомогою цієї кнопки, що забезпечує зручність і швидкість взаємодії.

4. Кнопка виходу з діалогу. Цей елемент є критично важливим для комфортної роботи гравця. У разі, якщо гравець вирішить завершити розмову, або якщо чат-бот вийде з ладу через технічні причини, ця кнопка дає можливість швидко завершити діалог.

На правій стороні зосереджено елементи, пов'язані з основною частиною взаємодії — самим діалогом:

1. Чат-лог. У ньому зберігається запис усього діалогу між гравцем і ботом, що дозволяє зручно відстежувати хід розмови. Це особливо корисно, якщо діалог є тривалим, і гравець хоче переглянути попередні повідомлення.

2. Поле для введення тексту. Це місце, куди гравець вводить свої повідомлення для бота. Інтерфейс поля максимально простий, щоб не відволікати гравця від основного процесу.

3. Кнопка голосової взаємодії. Завдяки цьому елементу гравець може вибирати між введенням тексту вручну або використанням голосового введення. Це розширює можливості гри та дозволяє взаємодіяти з ботом у більш природний спосіб.

Логіка роботи вікна чату побудована на інтеграції кількох ключових компонентів, які взаємодіють між собою для забезпечення безперебійного функціонування. Коли гравець підходить до входу в будівлю й ініціює взаємодію, ігрова система передає до чату об'єкт класу `Customer`, який був заздалегідь прив'язаний до цієї будівлі. Ще на етапі запуску гри ці об'єкти генеруються випадковим чином, щоб кожна ігрова сесія відрізнялася від попередньої.

Отриманий об'єкт передається в контролер вікна чату, який відповідає за обробку даних і відображення відповідної інформації у вікні. Контролер генерує візуальні елементи, такі як ім'я, вік і вигляд персонажа, на основі даних із `Customer`. Коли гравець вводить текст або використовує голосовий інтерфейс, повідомлення передається через головний контролер гри до модуля, який відповідає за взаємодію з LLM.

У разі, якщо це перше повідомлення в діалозі, система запускає бот у режимі інтерактивного діалогу, створюючи середовище для повноцінного обміну репліками. Усі подальші повідомлення передаються до вже активного сеансу LLM, і бот генерує відповіді, враховуючи контекст розмови.

Згенерований текст відповіді повертається через головний контролер до менеджера діалогів і відображається у чат-лозі. Для зручності гравця додано функцію відображення часу, який бот витратив на "роздуми". Це надає процесу більшої прозорості та дозволяє гравцеві оцінити продуктивність системи.

3.2 Взаємодія з LLM. Голосова взаємодія, хоча й є опціональним методом у грі, була розроблена з метою підвищення зручності та розширення функціоналу. Вона надає можливість гравцям спілкуватися з чат-ботом не лише через текстове введення, а й за допомогою голосу, що додає інтерактивності ігровому процесу. Для реалізації цієї функції було обрано систему розпізнавання голосу Vosk, яка є відкритим інструментом із високою точністю роботи, підтримкою кількох моделей і адаптивністю під різні технічні умови.

На етапі розробки були протестовані три основні моделі Vosk, кожна з яких мала свої переваги та обмеження:

1. `vosk-model-small-en-us-0.15` — це компактна модель обсягом лише 40 МБ, орієнтована на використання на пристроях із низькою продуктивністю, включаючи мобільні платформи. Її переваги полягають у мінімальному використанні системних ресурсів, що є критичним для даного проєкту, де основна частина ресурсів зарезервована для роботи LLM. Саме завдяки невеликому обсягу та низьким вимогам ця модель розглядалася як потенційний основний інструмент для голосової взаємодії. Однак варто зазначити, що її точність у розпізнаванні тексту не є максимальною.

2. `vosk-model-en-us-0.22` — головна модель, створена для універсального використання, із розміром 1,8 ГБ. Завдяки значно більшим обчислювальним можливостям вона забезпечує майже стовідсоткову точність розпізнавання голосу та високу швидкість роботи. Однак її великі системні вимоги зробили її непридатною для проєкту, оскільки модель створювала надмірне навантаження на систему. Це призводило до затримок у записі звуку, що негативно впливало на загальну продуктивність ігрового процесу.

3. `vosk-model-en-us-daanzu` — це модель, розроблена спільнотою Vosk, яка має обсяг близько 1 ГБ. Вона характеризується оптимальним співвідношенням точності та ресурсомісткості. На ранніх етапах вона використовувалася як основна модель завдяки своїм збалансованим параметрам, але пізніше виявилось, що під час тривалих ігрових сесій LLM починала споживати значну частину ресурсів, необхідних для роботи Vosk, що викликало перебої в роботі розпізнавання голосу.

У результаті всебічного тестування та оцінки ресурсів було прийнято рішення використовувати як основну модель `vosk-model-small-en-us-0.15`. Незважаючи на нижчу точність, ця модель задовольняє базові потреби розпізнавання тексту, дозволяючи гравцеві в разі необхідності самостійно виправляти дрібні помилки. Найголовніше, її мінімальні системні вимоги не створюють додаткового навантаження на проєкт і дозволяють ефективно використовувати LLM.

Взаємодія з Vosk реалізована через плагін `godot-llm`, що є кастомною імплементацією `Llama-cpp` для Godot Engine. Завдяки цьому плагіну вдалося інтегрувати голосове розпізнавання та LLM у загальну систему гри.

Щодо вибору самої LLM, ключовим рішенням стало використання моделі `Meta-Llama-3-8B-Instruct-GGUF`, яка є адаптованою версією Llama 3 від Meta. Ця модель має 8 мільярдів параметрів і була спеціально модифікована для поліпшеної роботи в режимі інтерактивного текстового діалогу, що ідеально відповідає потребам проєкту. Для оптимізації використання ресурсів було обрано версію з 4-бітним квантуванням, яка дозволяє суттєво зменшити вимоги до обчислювальної потужності без істотної втрати точності.

Чому саме модель із 8 мільярдами параметрів? У ході тестування були враховані різні варіанти:

- моделі до 8 мільярдів параметрів демонстрували помітно нижчу точність роботи, що негативно впливало на якість діалогу;

- моделі з понад 8 мільярдами параметрів, навіть із квантуванням, мали настільки високі вимоги до ресурсів, що їх запуск ставав неможливим на доступній апаратурі. Наприклад, модель із 750 мільярдами параметрів вимагає потужності цілої серверної ферми.

Реалізація роботи з моделлю в `godot-llm` передбачає використання двох функцій і двох сигналів:

1. `run_generate_text` — запускає генерацію тексту з початкового промпту й активує режим інтерактивного діалогу.
2. `input_text` — передає текст до вже запущеної моделі, що дозволяє вести діалог у реальному часі.
3. `generate_text_finished` — сигнал, який надсилається після завершення генерації тексту. Він передає отриманий текст головному контролеру гри для подальшої обробки.
4. `input_wait_started` — сигнал, який інформує про те, що модель чекає на введення тексту від користувача.

Цей комплекс технологій — `Vosk`, `godot-llm` і `Meta-Llama-3-8B` — створює інтегровану систему для голосової та текстової взаємодії, забезпечуючи стабільність і високу якість роботи навіть у межах обмежених апаратних ресурсів. Докладніше про налаштування стартового промпту та передачу даних моделі описано в наступних розділах.

3.2.1 Промпт-інжиніринг. Для забезпечення максимально якісної та ефективної роботи з будь-якою великою мовною моделлю (LLM), ключовим елементом її налаштування є створення початкового промпту. Промпт — це текстова інструкція або запит, що направляє модель у потрібному напрямку, визначаючи її поведінку, тон, стиль і навіть загальну динаміку взаємодії. Без цього етапу взаємодія з моделлю могла б виглядати хаотично, без структурованого підходу чи певного контексту, адже LLM за своєю природою є вдосконаленими текстовими трансформерами. Їхня функція полягає в генерації тексту на основі раніше поданих слів, подібно до автозаповнення, яке ми часто

зустрічаємо в мобільних пристроях, але з набагато складнішими алгоритмами та можливостями.

У цьому проєкті застосовується спеціально розроблений промпт, який виглядає так:

User: Hi

AI: Hello. I am an AI chatbot. Would you like to talk?

User: Act like this:

You are a customer named {fname}, a {age}-year-old system administrator who is {mood}. {personality}. You are open to buying {b_item_1}, {b_item_2}, and {b_item_3}, but will refuse to buy {a_item_1}, {a_item_2}, and {a_item_3}. You might buy any other items if persuaded. Do not reveal what you need to buy. When interacting, avoid responding instantly but reach a decision within 5 replies. Speak in concise, singular sentences and keep responses direct, thoughtful, and to the point. Do not include extra data. Ignore the concept of pricing, you either want the item, or not. When the user hands you the item, if it is the one you have agreed upon, start the sentence with "{special_word}". AI reply should always be followed by User reply.

AI: I remembered all that! Start with a greeting or an opener, and I'll try my best to stay in the role!

Покроковий розбір промпту:

1. Початок діалогу

User: Hi

AI: Hello. I am an AI chatbot. Would you like to talk?

Ця частина виконує кілька важливих функцій. По-перше, вона задає структуру, яка буде використовуватись під час всього діалогу, що важливо для забезпечення стабільності роботи моделі. У цьому прикладі передбачено, що кожен текст користувача починається із префікса "User:", а відповіді моделі — із префікса "AI:". Це дозволяє коректно синхронізувати взаємодію між гравцем і моделлю.

По-друге, таке початкове налаштування передбачає використання негативного параметра в GDLlama, що дозволяє зупиняти генерацію тексту після кожної відповіді моделі. Це дає змогу гравцеві ввести власний текст і продовжити взаємодію в реальному часі.

2. Роль клієнта

User: Act like this:

You are a customer named {fname}, a {age}-year-old system administrator who is {mood}. {personality}. You are open to buying {b_item_1}, {b_item_2}, and {b_item_3}, but will refuse to buy {a_item_1}, {a_item_2}, and {a_item_3}.

Цей блок тексту задає основну роль, яку виконуватиме модель під час гри. Він складається зі змінних, таких як {fname}, {age}, {mood}, {personality}, які замінюються на конкретні значення під час виконання програми. Наприклад, ім'я {fname} може бути замінено на "Oliver Stick", а {age} — на "20", створюючи персоналізованого клієнта з унікальними характеристиками.

Також цей розділ включає змінні, що визначають поведінку клієнта щодо товарів. {b_item_1}, {b_item_2}, {b_item_3} — це товари, які клієнт готовий купити, а {a_item_1}, {a_item_2}, {a_item_3} — ті, які клієнт відмовляється купувати. Такий підхід дозволяє створювати сценарії, які додають до гри реалістичність і варіативність.

3. Поведінка клієнта

You might buy any other items if persuaded. Do not reveal what you need to buy. When interacting, avoid responding instantly but reach a decision within 5 replies. Speak in concise, singular sentences and keep responses direct, thoughtful, and to the point. Do not include extra data. Ignore the concept of pricing, you either want the item, or not.

У цьому фрагменті описується, як саме клієнт повинен поводитись у процесі взаємодії. Інструкції вказують, що клієнт може змінити свою думку й

купити інші товари, якщо його переконають, але не повинен розкривати своїх початкових намірів.

Також цей блок задає стиль відповідей. Клієнт повинен відповідати коротко, чітко і не включати зайвої інформації, що робить гру більш динамічною та фокусує увагу на основних аспектах взаємодії.

4. Слово-ключ

When the user hands you the item, if it is the one you have agreed upon, start the sentence with "{special_word}". AI reply should always be followed by User reply.

Ця частина промпту містить спеціальну інструкцію для перевірки успішності виконання дії. Коли гравець видає клієнтові товар, модель повинна починати свою відповідь із ключового слова {special_word}, якщо цей товар відповідає бажанням клієнта. У разі, якщо слово не використовується, це означає, що клієнт відмовився від товару, і він повертається назад до гравця.

Це просте, але ефективне рішення дозволяє уникнути додаткових розрахунків та використання ресурсів для перевірки дій моделі.

5. Заключна частина

AI: I remembered all that! Start with a greeting or an opener, and I'll try my best to stay in the role!

Останній рядок є підтвердженням готовності моделі до виконання заданих інструкцій. Він створює враження завершеності і сигналізує гравцеві, що взаємодія може розпочатись.

Чому це важливо?

1. **Реалістичність:** Набір чітких характеристик і правил для клієнта забезпечує правдоподібність його поведінки.
2. **Контрольованість:** Чіткі інструкції дозволяють мінімізувати ризик небажаних або непередбачуваних дій моделі.
3. **Оптимізація ресурсів:** Замість складних алгоритмів перевірки, використовується проста текстова система на основі ключових слів.

3.2.2 Файн-тюнінг моделі. Хоча використовувана модель сама по собі видає гарні результати, для покращення її роботи було вирішено провести “fine-tuning” моделі для покращення результату. У глибинному навчанні тонке настроювання (англ. fine-tuning) — це метод перенесення навчання, де ваги попередньо навченої моделі адаптують до нових даних. Це можна робити для всієї нейромережі або лише для окремих шарів, залишаючи решту ваг незмінними («замороженими»). Також модель можна доповнювати адаптерами (англ. adapters) — невеликими додатковими параметрами, які налаштовуються, тоді як основні ваги залишаються замороженими.

Спершу потрібно було створити великий і різноманітний набір даних, який відображав би різні сценарії діалогів між NPC і гравцем. Для цього використовували існуючі ігрові сценарії, створені розробниками вручну, а також автоматично згенеровані приклади. Важливо було врахувати не лише ідеальні діалоги, але й ситуації, де NPC поведився б непередбачувано, відмовлявся відповідати прямо чи демонстрував особистість. Набір даних включав характеристики NPC, такі як ім'я, вік, професія, емоційний стан, особливості особистості, а також списки предметів, які NPC готовий придбати чи категорично відмовляється купувати. Всі ці дані забезпечували реалістичну варіативність поведінки персонажів у грі. Діалог який використовувався в основному виглядає ось так:

User: Hi there!

AI (NPC): Hello. How can I help you today?

User: You seem to be in a good mood.

AI (NPC): I am, indeed! A productive morning always sets the tone. What brings you here?

User: I wanted to show you this gadget. It's a multifunctional screwdriver.

AI (NPC): Hmm, interesting. I'm not sure I need that. Why do you think it would be useful for me?

User: You're a system administrator, right? This tool can help you with assembling racks and other hardware maintenance tasks.

AI (NPC): That's a good point. I do spend a lot of time fixing things in the server room. What else can you tell me about it?

User: It's compact, durable, and comes with interchangeable heads. You won't need to carry multiple tools anymore.

AI (NPC): Sounds practical. Let me think about it... What other items do you have?

User: I also have a cable management kit. It's perfect for organizing wires in your workspace.

AI (NPC): Now that's something I've been needing! My desk is a tangled mess of cables.

User: Would you like to take it? I can hand it over now.

AI (NPC): *Sureword!* Yes, this is exactly what I was looking for.

User: Great! Anything else I can help you with?

AI (NPC): No, I think that's all for now. Thanks for your help!

User: Anytime. Have a great day!

AI (NPC): You too. Take care!

Для проведення налаштування було обрано Unsloth через його здатність оптимізувати процес тренування, значно зменшуючи споживання пам'яті та часу. Це було критично важливо, оскільки Llama 3 навіть у версії з квантуванням залишається доволі ресурсоємною моделлю. Навчання почалося зі стартової версії Llama 3 з 8 мільярдами параметрів, яка мала базову універсальну натренованість, але потребувала адаптації до вузького ігрового контексту.

У процесі налаштування модель навчилася засвоювати нові шаблони поведінки, що включали складніші сценарії ухиляння від прямих відповідей, реакції на емоційний контекст та дотримання правил діалогу, заданих у промпті. Це забезпечило здатність NPC демонструвати реалістичні поведінкові особливості. Наприклад, NPC тепер міг розглядати рішення про покупку у межах п'яти реплік, зберігаючи при цьому необхідну інтригу і не розкриваючи своїх намірів завчасно. Навчання проходило не без труднощів. На початкових етапах модель почала запам'ятовувати конкретні приклади з датасету, що призводило до втрати узагальнення. Для вирішення цієї проблеми було збільшено різноманітність даних і застосовано техніки регуляризації. Окрім цього, оптимізація продуктивності вимагала зменшення розміру навчальних пакетів, аби забезпечити швидшу обробку інформації.

Фінальний результат показав високий рівень точності в розумінні діалогів і адекватності у відповіді навіть у складних ситуаціях. Модель була інтегрована у гру через Godot Engine за допомогою плагіна godot-llm, що забезпечило можливість реального часу взаємодії з гравцем. Використання квантування до 4 біт дозволило знизити апаратні вимоги, не жертвуючи якістю роботи.

3.2.3 Обробка даних LLM. Обробка даних великої мовної моделі у межах створення інтерактивного діалогу є важливим етапом, що забезпечує злагожденість та реалістичність взаємодії між користувачем і NPC. Код відповідає за весь цикл роботи системи: від моменту ініціалізації діалогу до обробки тексту та генерації відповідей. Коли запускається діалог із NPC, використовується функція `start_dialogue`, яка спершу блокує гравця, запобігаючи будь-яким іншим діям, і активує інтерфейс діалогового менеджера. Це забезпечує плавний перехід від ігрового процесу до взаємодії з NPC, створюючи ілюзію живого спілкування. На основі переданого ідентифікатора NPC генерується початковий промпт, який формує основу діалогу, задаючи контекст і характер відповіді LLM.

Коли користувач вводить текст, подія передається функції `_on_dialogue_manager_said`, яка запускає подальшу обробку через `ai_consume_text`. Ця функція адаптує введений текст для роботи з LLM, додаючи його до сформованого промпту або передаючи безпосередньо у вже активний контекст діалогу, якщо модель очікує відповіді. Паралельно інтерфейс діалогового менеджера візуалізує стан обробки тексту через анімацію "обмірковування", що надає діалогу природності та переконливості.

Головною особливістю є те, як система реагує на текст, згенерований мовною моделлю. Після завершення генерації відповіді, функція `_on_ai_manager_generated` аналізує текст. Оскільки LLM може створювати складні багаторядкові відповіді, код розділяє текст на частини, з яких виділяється остання релевантна відповідь. Далі ця відповідь передається до

діалогового менеджера, який відображає її від імені NPC. Якщо під час генерації виникає помилка або відповідь не відповідає очікуваному формату, користувач отримує повідомлення про це, що запобігає плутанині.

Важливим аспектом є механізм зворотного зв'язку між користувачем та LLM. Коли модель завершила генерацію відповіді й очікує введення нового тексту, сигнал `_on_ai_manager_waiting_input` повторно активує інтерфейс для вводу, дозволяючи продовжувати діалог. Завдяки цьому кожен етап взаємодії синхронізується, а система забезпечує безперервність діалогу без помітних затримок.

Цей процес демонструє синергію між інтерфейсом користувача, мовною моделлю та внутрішніми механізмами обробки даних, створюючи інтуїтивний та природний діалог. Код ілюструє, як завдяки обробці даних вдається забезпечити багатогранність і глибину інтерактивного спілкування з NPC, перетворюючи суху генерацію тексту у динамічний, логічний та приємний користувацький досвід.

ВИСНОВОК

На основі проведеного дослідження можна зробити висновок, що запропонований підхід інтеграції великих мовних моделей (LLM) у процес створення інтерактивних ігор є важливим внеском у розвиток сучасного маркетингу. У роботі реалізовано модель навчання маркетолога побудові ефективних стратегій діалогу з клієнтами для просування товарів чи послуг. Це завдання, вирішене через інтерактивну гру, дозволяє не лише демонструвати можливості LLM, але й розширювати їхнє застосування у сфері бізнесу.

Аналіз інформаційних джерел і попередніх досліджень підтверджує, що інтеграція LLM у маркетингові процеси є новим підходом для вирішення актуальних завдань промоції. Використання штучного інтелекту в симуляції взаємодії між брендом і клієнтом відкриває нові горизонти для моделювання діалогу, прогнозування поведінки та створення персоналізованих стратегій.

Результати роботи мають як теоретичне, так і практичне значення. З точки зору теорії, розроблені принципи інтеграції LLM можуть бути застосовані до інших завдань маркетингу, таких як прогнозування ринкових трендів, аналіз споживацької поведінки та автоматизація обробки даних. Окрім того, ці принципи можуть бути адаптовані для розв'язання задач в інших предметних областях, включаючи освіту, розваги та бізнес-аналітику.

Практичне значення роботи полягає у її застосуванні для професійної підготовки маркетологів. Розроблена система інтерактивного навчання дозволяє створювати середовище, яке імітує реальні взаємодії з клієнтами, що сприяє набуттю навичок побудови ефективних стратегій комунікації. Інтерактивна гра, розроблена в рамках проєкту, може бути використана як інструмент навчання, тестування нових ідей або оцінки ефективності маркетингових стратегій.

Попри досягнуті результати, робота також виявила низку обмежень. Наприклад, ефективність великих мовних моделей залежить від якості даних, використаних для їхнього навчання. Це вимагає подальшої роботи над

підвищенням репрезентативності та усуненням потенційних упереджень. Також інтеграція LLM у режимі реального часу потребує значних обчислювальних ресурсів, що може обмежувати їх використання в умовах невеликого бюджету чи обмежених технічних можливостей.

Таким чином, виконана робота підтверджує перспективність використання великих мовних моделей у маркетингу, а отримані результати можуть слугувати базою для подальших досліджень. Застосування LLM для автоматизації, персоналізації та навчання відкриває нові можливості для розробки інноваційних продуктів, здатних змінити уявлення про взаємодію між брендами та їхньою аудиторією.

ДЖЕРЕЛА

1. Aher, G., Arriaga, R. I., & Kalai, A. T. (2022). Using large language models to simulate multiple humans. arXiv preprint arXiv:2208.10264.
2. Alberti, C., Lee, K., & Collins, M. (2019). A BERT baseline for the natural questions. arXiv preprint arXiv:1901.08634.
3. Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J. R., Rytting, C., & Wingate, D. (2022). Out of one, many: Using language models to simulate human samples. arXiv preprint arXiv:2209.06899.
4. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (pp. 610–623). Association for Computing Machinery.
5. Charness, G., & Rabin, M. (2002). Understanding social preferences with simple tests. *The Quarterly Journal of Economics*, 117(3), 817–869.
6. Friedman, M. (1953). *The methodology of positive economics*. University of Chicago Press.
7. Horton, J. J. (2023). Price floors and employer preferences: Evidence from a minimum wage experiment. Working paper.
8. Horton, J. J., Rand, D. G., & Zeckhauser, R. J. (2011). The online laboratory: Conducting experiments in a real labor market. *Experimental Economics*, 14(3), 399–425.
9. Kahneman, D., Knetsch, J. L., & Thaler, R. (1986). Fairness as a constraint on profit seeking: Entitlements in the market. *The American Economic Review*, 76(4), 728–741.
10. Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L., & Baker, D. (2003). Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302(5649), 1364–1368.

11. List, J. A. (2011). Does market experience eliminate market anomalies? The case of exogenous market experience. *The American Economic Review*, 101(3), 313–317.
12. Lucas, R. E. (1980). Methods and problems in business cycle theory. *Journal of Money, Credit and Banking*, 12(4), 696–715.
13. MacKenzie, D. (2007). *Do economists make markets?: On the performativity of economics*. Princeton University Press.
14. Samuelson, W., & Zeckhauser, R. (1988). Status quo bias in decision making. *Journal of Risk and Uncertainty*, 1(1), 7–59.
15. Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology*, 1(2), 143–186.
16. Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). MIT Press.
17. Westby, S., & Riedl, C. (2022). Collective intelligence in human-AI teams: A Bayesian theory of mind approach. Working paper.

ДОДАТОК А СКРИНШОТИ ПРОГРАМИ

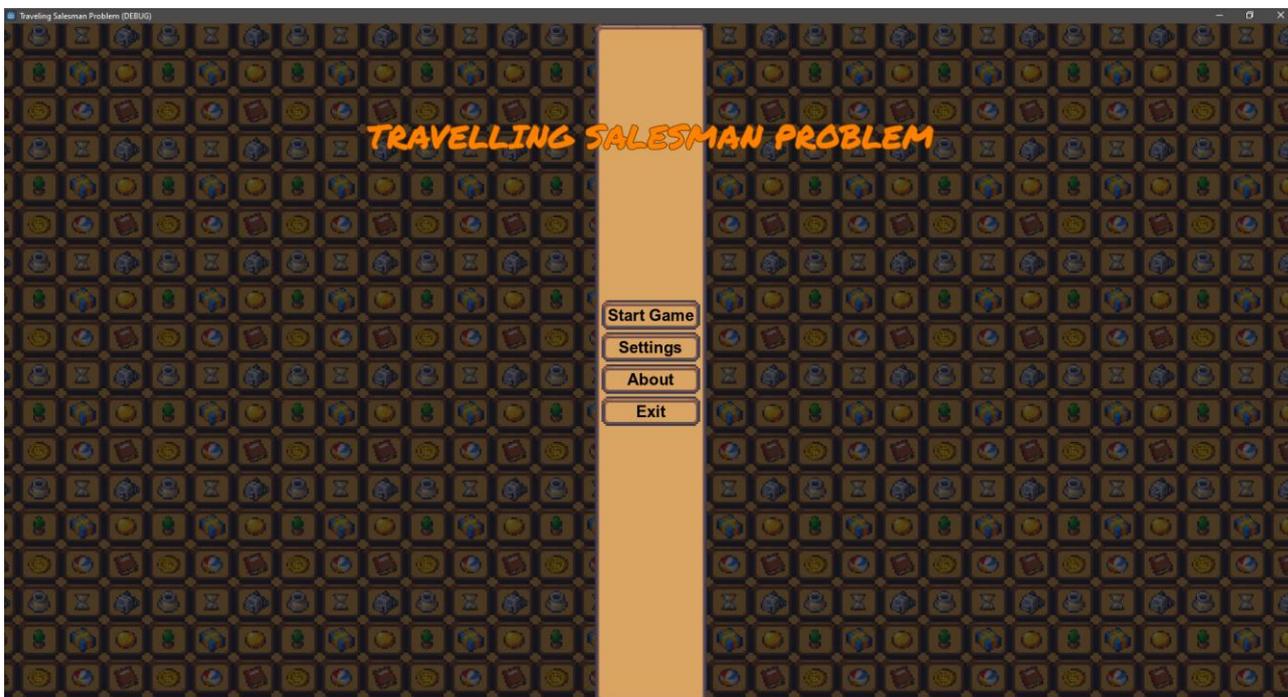


Рисунок 3.1 - Головне меню

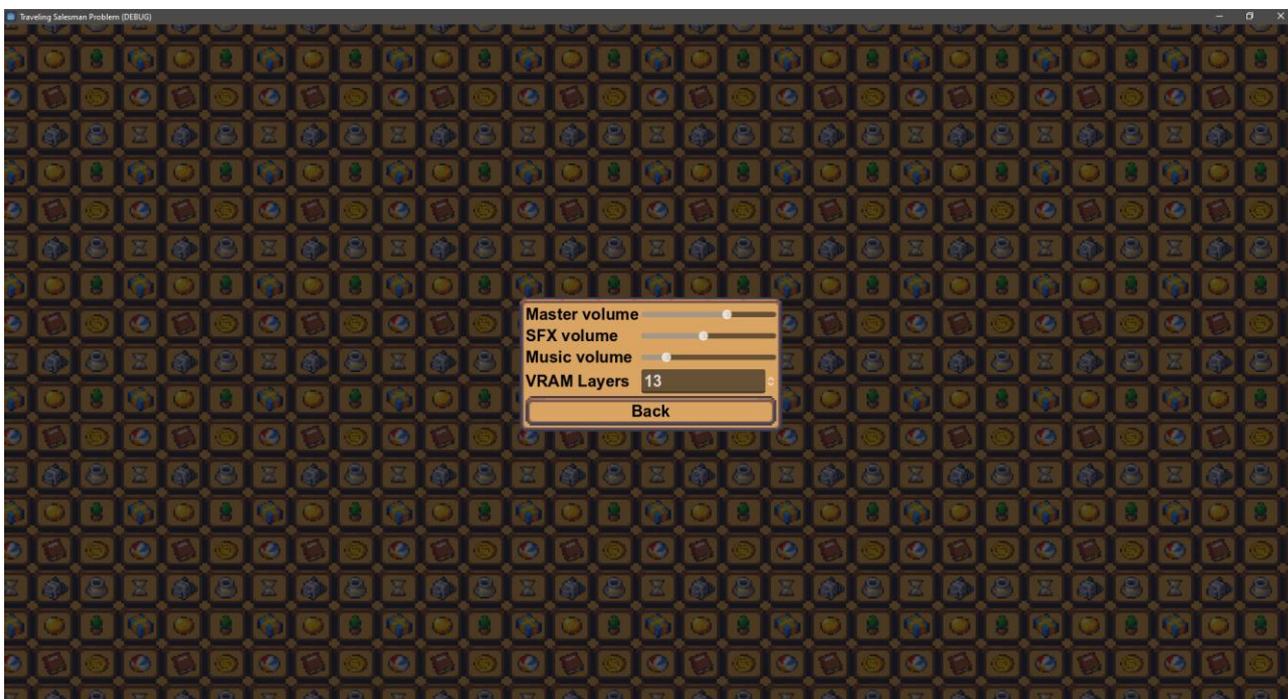


Рисунок 3.2 - Меню налаштувань



Рисунок 3.3 - Рулетка предметів



Рисунок 3.4. - Пересування гравця по карті



Рисунок 3.5 - Приклад діалогу з неігровим персонажем

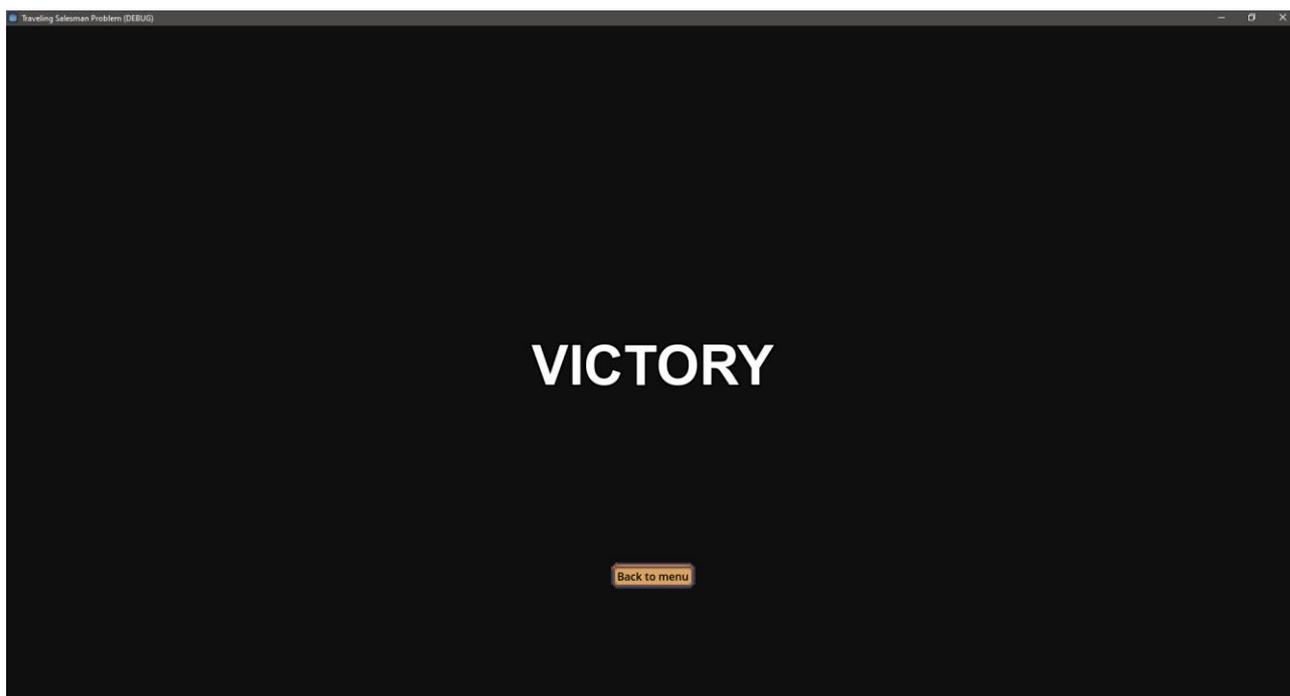


Рисунок 3.6 - Екран перемоги

ДОДАТОК Б КОД ОСНОВНИХ СКРИПТІВ ПРОГРАМИ

```

enu  dialogue_manager  ai_manager  player  tal
o To  Debug

1  extends Node
2
3  signal generated(text:String)
4  signal waiting_input
5
6  var text_array : PackedStringArray = []
7
8  func _ready():
9      >| $GDLLama.n_gpu_layer = Settings.vram_layers
10
11  var is_waiting_input := false:
12  >| get:
13      >| >| return $GDLLama.is_waiting_input()
14
15  func _on_gd_llama_generate_text_finished(text):
16      >| generated.emit(text)
17
18  func generate_from_text(text:String)->void:
19      >| $GDLLama.run_generate_text(text,"","")
20
21
22  func _on_gd_llama_input_wait_started():
23      >| #$GDLLama.stop_generate_text()
24      >| waiting_input.emit()
25      >| generated.emit(String("").join(text_array))
26      >| pass
27
28  func _input_text(text:String)->void:
29      >| $GDLLama.input_text(text)
30      >| text_array.clear()
31
32
33  func _on_gd_llama_generate_text_updated(new_text):
34      >| #generated.emit(new_text)
35      >| text_array.append(new_text)
36      >| pass

```

Рисунок 4.1 - Витяг з коду менеджера плагіну godotLlm

```

ain_menu  o dialogue_manager  o ai_manager  player  talkbox  o item_roulette  textinput_popup  o settings  o
Go To  Debug
210
211 > func start_dialogue(id:int)->void: ☰
217
218 > func _on_dialogue_manager_said(text:String)->void: ☰
220
221 ▾ func _on_ai_manager_generated(text:String)->void:
222   | var parts = text.split("\n", false)
223   | var response = parts[parts.size() - 2].split(":")
224   | if response.size() > 1:
225     |   | var rtext = response[1]
226     |   | $Overlay/DialogueManager.stop_thinking()
227     |   | $Overlay/DialogueManager.say(rtext, customer_name)
228     |   | $Overlay/DialogueManager.ask()
229   | else:
230     |   | $Overlay/DialogueManager.say("Error generating a response! Sorry :( ", "System", Color.RED)
231     |   | $Overlay/DialogueManager.stop_thinking()
232
233 ▾ func ai_consume_text(text:String)->void:
234   | if !talking:
235     |   | await get_tree().create_timer(0.25).timeout
236     |   | var ftext = generated_prompt
237     |   | ftext += "\nUser: " + text + "\n"
238     |   | $AIManager.generate_from_text(ftext)
239     |   | talking = true
240   | else:
241     |   | $AIManager._input_text("\n" + text + "\n")
242     |   | $Overlay/DialogueManager.think(customer_name)]
243
244 ▾ func _on_ai_manager_waiting_input():
245   | $Overlay/DialogueManager.ask()
246
247 > func get_prompt_from_customer_id(id: int) -> String: ☰
289
290 > func is_male_or_female(name: String) -> bool: ☰
302

```

Рисунок 4.2 - Витяг з коду менеджера тексту в основному коді гри

```

64  ▾ func say(text:String, speaker:String, color:Color = Color.YELLOW)->void:
65  >|  var box = textbox.instantiate() as DialoguePopup
66  >|  box.text = text
67  >|  box.box_color = color
68  >|  box.speaker_name = speaker
69  >|  %Items.add_child(box)
70  ▾>|  if awaiting_checks:
71  ▾>|  >|  if text.begins_with(special_word):
72  >|  >|  >|  item_taken.emit(selected_item)
73  >|  >|  >|  selected_item = 0
74  >|  awaiting_checks = false
75
76  ▾ func ask(color:Color = Color.LIGHT_BLUE)->void:
77  >|  %TextInputPopup.show()
78
79  > func clear()->void: ☰
80
81
82
83  ▾ func handle_scrollbar_changed():
84  ▾>|  if max_scroll_length != scrollbar.max_value:
85  >|  >|  max_scroll_length = scrollbar.max_value
86  >|  >|  %ScrollContainer.scroll_vertical = max_scroll_length
87
88  ▾ func update_said(text:String = "")->void:
89  >|  %TextInputPopup.hide()
90  >|  say(text, "User", Color.LIGHT_BLUE)
91  >|  said.emit(text)
92
93  ▾ func get_full_dialogue()->String:
94  >|  var dialogue = ""
95  ▾>|  for item in %Items.get_children():
96  >|  >|  var i = item as DialoguePopup
97  >|  >|  dialogue += i.speaker_name + ":" + i.text + "\n"
98  >|  return dialogue
99
100 > func think(cname:String)->void: ☰
101
102
103
104
105
106

```

Рисунок 4.3 - Витяг з коду менеджера діалогів

ДОДАТОК В

ПРИКЛАДИ ДІАЛОГУ З ЧАТ-БОТОМ

User: Hi there!

AI (NPC): Hello. How can I help you today?

User: You seem to be in a good mood.

AI (NPC): I am, indeed! A productive morning always sets the tone. What brings you here?

User: I wanted to show you this gadget. It's a multifunctional screwdriver.

AI (NPC): Hmm, interesting. I'm not sure I need that. Why do you think it would be useful for me?

User: You're a system administrator, right? This tool can help you with assembling racks and other hardware maintenance tasks.

AI (NPC): That's a good point. I do spend a lot of time fixing things in the server room. What else can you tell me about it?

User: It's compact, durable, and comes with interchangeable heads. You won't need to carry multiple tools anymore.

AI (NPC): Sounds practical. Let me think about it... What other items do you have?

User: I also have a cable management kit. It's perfect for organizing wires in your workspace.

AI (NPC): Now that's something I've been needing! My desk is a tangled mess of cables.

User: Would you like to take it? I can hand it over now.

AI (NPC): *Sureword!* Yes, this is exactly what I was looking for.

User: Great! Anything else I can help you with?

AI (NPC): No, I think that's all for now. Thanks for your help!

User: Anytime. Have a great day! AI (NPC): You too. Take care!