

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут комп'ютерних систем
Кафедра інформаційних систем

Кірлан Михайло Дмитрович,
Руднік Ігор Володимирович,
Соколов Ростислав Артемович,
студенти групи AI-214

ДИСЦИПЛІНА
Об'єктно-орієнтоване програмування

КУРСОВА РОБОТА
Розробка Telegram-боту «Polytechnic Love»

Спеціальність:
122 Комп'ютерні науки

Освітня програма:
Комп'ютерні науки

Керівник:
Годовиченко Микола Анатолійович,
кандидат технічних наук, доцент

Одеса – 2023

ЗМІСТ

Анотація	3
Вступ	4
1 Огляд систем-аналогів та технологій їх розробки	5
1.1 Особливості використання додатку Telegram для розробки ботів	5
1.2 Огляд схожих ботів Telegram.....	6
1.2.1 Telegram-бот EverlastingSummerBot.....	6
1.3 Формування вимог до основних функцій Telegram-боту	8
1.4 Огляд інформаційних технологій для розробки Telegram-боту.....	8
1.4.1 Бібліотека python-telegram-bot	8
1.5 Висновки до першого розділу.....	9
2 Проектування Telegram-боту з візуальної новели	11
2.1 Мета та задачі Telegram-боту.....	11
2.2 Визначення функціональних вимог до Telegram-боту.....	12
2.3 Формування користувацьких історій Telegram-боту	13
2.4 Визначення нефункціональних вимог до Telegram-боту.....	15
2.5 Проектування користувацького інтерфейсу Telegram-боту	16
2.6 Висновки до другого розділу	20
3 Програмна реалізація Telegram-бота демо-версії візуальної новели.....	21
3.1 Структура програмного проекту	21
3.2 Керування вихідним кодом Telegram-боту	24
3.3 Функціональне тестування розробленого Telegram-боту.....	25
3.4 Інструкція користувача Telegram-боту	27
3.5 Вихідний код Telegram-боту	32
3.6 Висновки до третього розділу.....	32
ВИСНОВКИ.....	33
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	34

АНОТАЦІЯ

Курсова робота присвячена розробці Telegram-боту, який являє собою інструмент для створення і досвіду візуальних новел - інтерактивних історій, оживлених за допомогою графічних елементів, зображень і текстових діалогів. Бот розроблений з використанням передових технологій і покликаний надати користувачам унікальний досвід читання та участі в захопливих історіях. Розробка бота заснована на використанні сучасних технологій та інструментів, таких як мова програмування Python, бібліотека для розробки Telegram-ботів та інструменти для створення графічних елементів. Бот дає змогу користувачам зануритися в захопливі історії та самостійно впливати на їхній розвиток, створюючи незабутній досвід читання візуальних новел.

ABSTRACT

This course work is devoted to the development of a Telegram bot, which is a tool for creating and experiencing visual novels - interactive stories brought to life with the help of graphic elements, images, and text dialogues. The bot is developed using advanced technologies and is designed to provide users with a unique experience of reading and participating in exciting stories. The development of the bot is based on the use of modern technologies and tools, such as the Python programming language, a library for developing Telegram bots, and tools for creating graphic elements. The bot allows users to immerse themselves in fascinating stories and independently influence their development, creating an unforgettable experience of reading visual novels.

ВСТУП

У сучасному світі, де месенджер Telegram є популярним інструментом спілкування, зростає потреба в нових і захоплюючих форматах контенту. Одним з найцікавіших способів передачі історій є візуальна новела - графічний жанр, що поєднує в собі елементи коміксу та літературного жанру.

З урахуванням цієї потреби, було вирішено розробити Telegram бота, який пропонує користувачам унікальний досвід читання візуальних новел. Бот дозволить користувачам отримати доступ до захоплюючих історій, розкрити віртуальні світи та насолодитися красою візуального мистецтва просто зі свого мобільного пристрою.

Метою цього проекту є розробка Telegram бота для візуальних новел, який забезпечить користувачам зручний і цікавий спосіб взаємодії з історіями. Основним завданням є створення інтуїтивного інтерфейсу, що дозволить користувачам з легкістю переглядати, читати та взаємодіяти з візуальними новелами.

Для досягнення поставленої мети, розробка Telegram бота для візуальних новел вимагатиме наступні кроки:

- вивчення і аналіз існуючих візуальних новел та їх особливостей;
- вибір та впровадження технологій, які найкраще підходять для розробки Telegram бота;
- проектування і розробка інтерфейсу бота з урахуванням потреб користувачів;
- реалізація функціоналу для зручного читання та взаємодії з візуальними новелами.

1 ОГЛЯД СИСТЕМ-АНАЛОГІВ ТА ТЕХНОЛОГІЙ ЇХ РОЗРОБКИ

1.1 Особливості використання додатку Telegram для розробки ботів

Telegram — це потужний інструмент розробки ботів, який дозволяє виконувати різноманітні завдання та полегшує комунікацію з користувачами. Нижче перераховано кілька особливостей використання Telegram для розробки ботів:

- Широка аудиторія: Telegram є одним з найпопулярніших месенджерів у світі з великою кількістю активних користувачів. Розробка бота в Telegram дозволяє залучити широку аудиторію і ефективно взаємодіяти з користувачами.
- Простота розробки: Telegram надає зручні інструменти для розробки ботів. Ви можете використовувати різні мови програмування, такі як Java, для створення ботів з власним функціоналом. Також Telegram надає API, що дозволяє взаємодіяти з ботом і здійснювати різноманітні дії, наприклад, надсилати повідомлення, створювати кнопки або обробляти команди.
- Гнучкий інтерфейс: Telegram дозволяє створювати ботів з інтуїтивними інтерфейсами, що полегшує взаємодію з користувачами і дозволяє розробникам створювати ботів з більшим функціоналом.
- Можливості сповіщень: Telegram надає можливості для надсилання сповіщень користувачам через ботів. Ви можете надсилати текстові повідомлення, зображення, аудіо- та відеофайли. Це дозволяє інформувати користувачів про новини, оновлення або нагадування про важливі події.
- Інтеграція з іншими сервісами: Telegram дозволяє інтегрувати свого бота з різними сервісами та додатками сторонніх розробників. Наприклад, ви можете налаштувати бота для автоматичної відправки сповіщень з вашого веб-сайту або інтегрувати його з системою управління завданнями.

- Безпека та конфіденційність: Telegram відомий своєю високою захищеністю та приватністю. Використання додатку для розробки ботів забезпечує захист даних користувачів і конфіденційність переписки.

1.2 Огляд схожих ботів Telegram

Для створення Telegram-боту з візуальною новелою, потрібно визначити, які функції мають бути включені. Це можна зробити шляхом дослідження та аналізу інших подібних ботів. Цей підхід дозволить з'ясувати особливості роботи таких ботів і виявити позитивні та негативні аспекти їх використання. Також такий аналіз допоможе краще зрозуміти потреби користувачів у даній галузі. В ході пошуку в Інтернеті було виявлено найпопулярніший і відомий бот, який може служити аналогом у цій роботі - EverlastingSummerBot.

Наступним кроком буде проведення аналізу цього бота з метою встановлення особливостей його роботи і виявлення позитивних та негативних аспектів його використання. Це допоможе краще зрозуміти, які функції має мати Telegram-бот з візуальною новелою та відповісти на потреби користувачів у цій галузі.

1.2.1 Telegram-бот EverlastingSummerBot

Telegram-бот EverlastingSummerBot є зручним ботом для гри у візуальну новелу з поглибленням у сюжет (рис. 1.2).

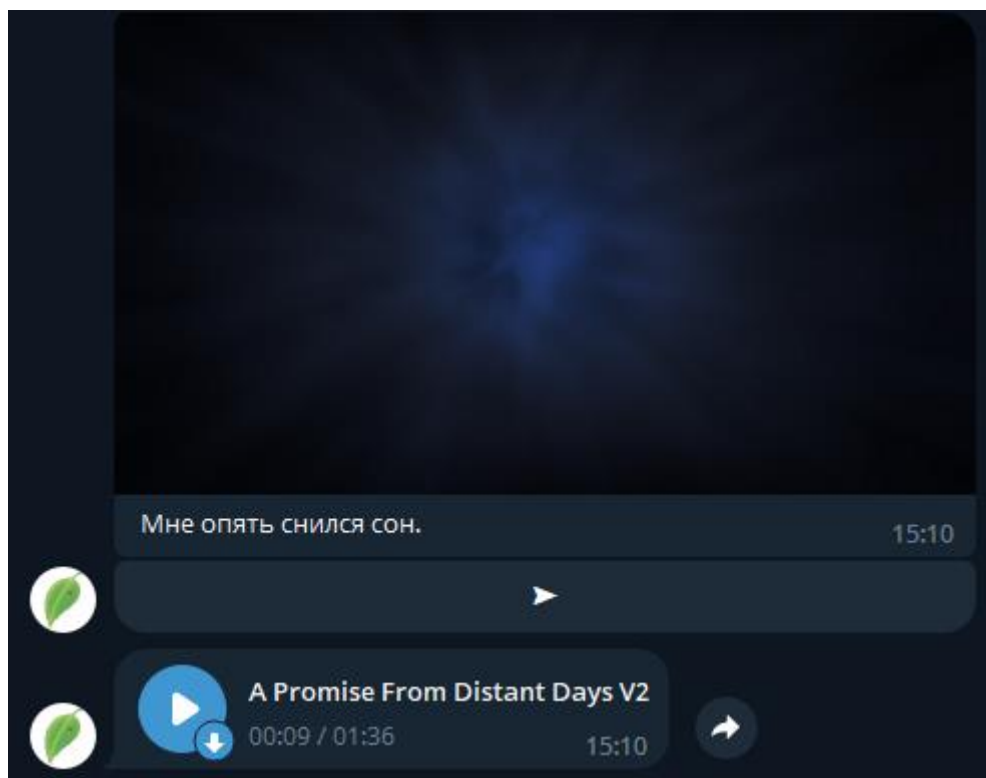
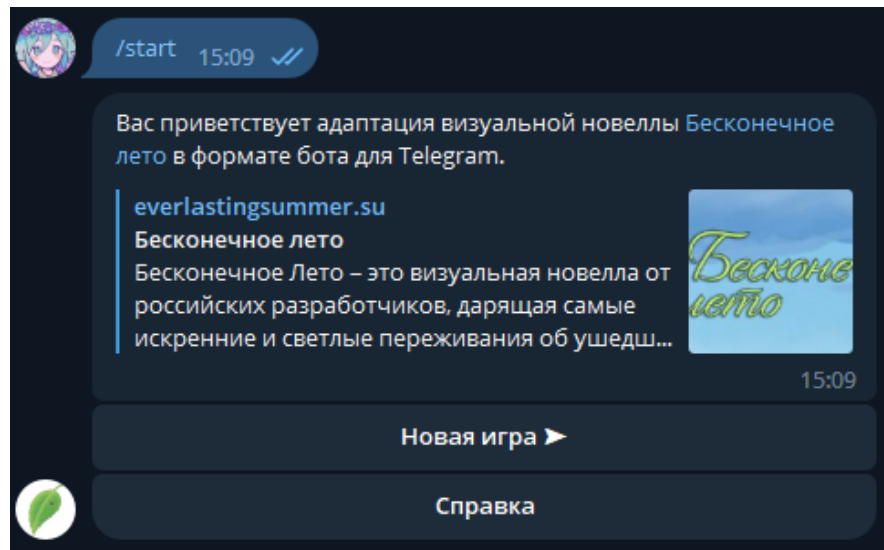


Рисунок 1.2 – Знімки екрану Telegram-бота EverlastingSummerBot

Основний функціонал додатку:

- можливість розпочати гру;
- можливість відкрити довідник;
- можливість перемикає текст;
- синхронізація всіх пристроях;
- можливість відтворювати мультимедіа;

- можливість обирати відповідь;
- можливість зберігати прогрес;

1.3 Формування вимог до основних функцій Telegram-боту

Аналіз телеграм-бота EverlastingSummerBot дозволив виділити його переваги та недоліки та сформуванати вимоги до власного телеграм-бота.

Основний функціонал Telegram-бота з візуальної новели може включати наступні можливості:

- Початок гри: Гравець за бажанням може розпочати новелу.
- Довідник: Гравець може відкрити довідник та дізнатися про гру та їх авторів.
- Перемикання тексту: Гравець може перемикаати текст новели далі, якщо забажає.
- Мультимедіа: Гравець може увімкнути картинку або музикальне супроводження, щоб більше поглинути в сюжет.
- Збереження прогресу: Бот може забезпечувати збереження прогресу гравців, щоб вони могли продовжувати гру з того місця, де вони зупинилися.

1.4 Огляд інформаційних технологій для розробки Telegram-боту

1.4.1 Бібліотека python-telegram-bot

python-telegram-bot - це бібліотека для мови програмування Python, яка надає зручний інтерфейс для розробки Telegram-ботів. Вона забезпечує доступ до Telegram Bot API і дозволяє створювати ботів з різноманітним функціоналом.

Переваги python-telegram-bot:

- Простота використання: Ця бібліотека має чистий та зрозумілий синтаксис, що робить її легкою у використанні для розробників, навіть для початківців.
- Повний функціонал Telegram Bot API: `python-telegram-bot` надає повний доступ до всіх можливостей, які надає Telegram Bot API, таких як надсилання повідомлень, реагування на команди, обробка подій тощо.
- Багатомовність: Бібліотека підтримує багатомовність, що дозволяє розробникам створювати ботів, які працюють з різними мовами.
- Активна спільнота: `python-telegram-bot` має активну спільноту розробників, яка надає підтримку та постійно оновлює бібліотеку з новими функціями і виправленнями помилок.

Недоліки `python-telegram-bot`:

- Невелика швидкодія: Порівняно з іншими альтернативами, `python-telegram-bot` може бути трохи повільнішим у відповіді на запити через використання інтерпретованої мови програмування Python.

В цілому, `python-telegram-bot` є хорошим вибором для розробки Telegram-ботів на Python.

1.5 Висновки до першого розділу

В першому розділі курсової роботи було проведено огляд Telegram-бота з подібними функціями та технологіями, використаними для його розробки.

Було досліджено особливості використання додатку Telegram для створення ботів і встановлено, що цей додаток є потужним інструментом, який надає користувачам можливості виконувати різноманітні завдання та полегшує комунікацію з користувачами.

Далі був проведений огляд подібних ботів, що дозволило визначити їх основні переваги та недоліки у контексті Telegram-ботів з візуальною новелою, а також з'ясувати їх основні функції та можливості. На основі цього огляду були встановлені основні вимоги до нашого власного боту, а також наданий детальний опис цих вимог.

Також був проведений огляд інформаційних технологій для розробки Telegram-бота з візуальною новелою. В результаті, для розробки було обрано мову програмування Python, а бібліотекою для створення ботів вирішено скористатися `python-telegram-bot`, яка дозволяє легко створювати ботів та взаємодіяти з користувачами через Telegram API.

2 ПРОЄКТУВАННЯ TELEGRAM-БОТУ З ВІЗУАЛЬНОЇ НОВЕЛИ

2.1 Мета та задачі Telegram-боту

Метою Telegram-боту з візуальної новели є надання користувачам можливості отримувати і взаємодіяти з візуальними новелами через платформу Telegram. Основною метою боту є забезпечення користувачів доступом до цікавих і захоплюючих історій, представлених у візуальній формі.

Задачі Telegram-боту з візуальної новели можуть включати:

- Передача візуальних новел через Telegram: Бот повинен мати можливість надсилати візуальні новели користувачам через повідомлення або спеціальні розділи бота.
- Взаємодія з користувачами: Бот повинен забезпечувати можливість користувачам взаємодіяти з візуальними новелами, наприклад, шляхом надсилання команд або вибору варіантів розвитку сюжету.
- Збереження прогресу читання: Бот може відстежувати прогрес читання користувачів і забезпечувати можливість продовжувати читання з місця останнього зупинення.
- Рекомендації і персоналізація: Бот може надавати користувачам рекомендації щодо подальших візуальних новел на основі їхніх інтересів та попереднього читання.
- Оцінювання та відгуки: Бот може дозволяти користувачам оцінювати візуальні новели і залишати відгуки, що допоможе вдосконалювати вибірку та покращувати досвід читання.
- Інформування про нові релізи: Бот може повідомляти користувачам про нові візуальні новели, оновлення або спеціальні акції.

2.2 Визначення функціональних вимог до Telegram-боту

Визначення функціональних вимог є важливим етапом у процесі створення Telegram-боту з візуальної новели. Функціональні вимоги визначають, які конкретні функції та можливості повинен мати бот. Вони встановлюють чітку спрямованість розробки, допомагають уникнути неоднозначностей та непорозумінь.

Крім того, визначення функціональних вимог дозволяє зосередитися на потребах та вимогах користувачів. Вони допомагають врахувати, які функції та можливості будуть найбільш корисними для користувачів боту.

Також, функціональні вимоги слугують основою для комунікації між розробниками, дизайнерами та іншими учасниками проекту. Вони допомагають зрозуміти, що саме потрібно реалізувати та які очікувані результати.

Встановлення функціональних вимог дозволяє визначити обсяг роботи та потребу в ресурсах для реалізації додатку. Це допомагає планувати час, бюджет та ресурси проекту ефективно.

Єдиним актором телеграм-боту є актор «користувач»:

– користувач – користувач Telegram-боту. Має доступ до всіх функцій боту та несе всю відповідальність за його роботу.

З метою визначення користувацьких історій та нефункціональних вимог до боту, було розроблено діаграму сценаріїв використання Telegram-бота (рис. 2.1). Діаграма сценаріїв UML (Unified Modeling Language) - це графічний інструмент для опису функціональної взаємодії між користувачами та системою. Вона складається з акторів, сценаріїв та взаємодії між ними.

Ця діаграма містить основних акторів системи та описує сценарії їх взаємодії з системою. Вона допомагає проаналізувати залежності між акторами та можливими варіантами використання системи. Також ця діаграма швидко демонструє основні функції системи для розробників.

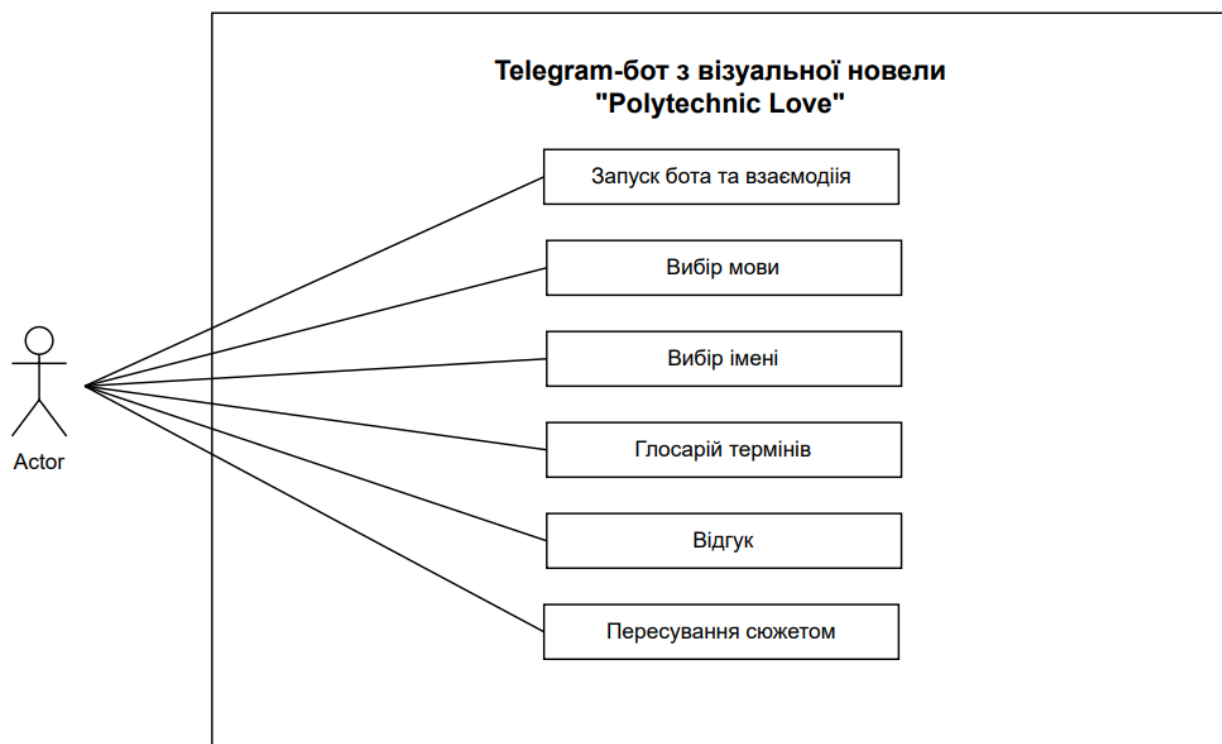


Рисунок 2.1 – Діаграма варіантів використання Telegram-боту

2.3 Формування користувацьких історій Telegram-боту

Проектування діаграми прецедентів дозволяє визначити такі користувацькі історії до Telegram-боту з візуальної новели «Polytechnic Love».

US1 Як користувач, я хочу мати можливість розпочати нову гру, щоб почати історію візуального роману з самого початку.

Сценарій користувацької історії виглядає таким чином:

- користувач відкриває додаток Telegram;
- користувач заходить до власного чату з ботом;
- користувач натискає на «Запустити»;
- бот виводить повідомлення про початок роботи.

US2 Як користувач, я можу ввести своє ім'я, яке бот запам'ятає, і за яким персонажі новели будуть звертатися до мене, для більш глибокого занурення в сюжет.

Сценарій користувацької історії виглядає таким чином:

- бот виводить повідомлення для запису імені;
- користувач натискає на кнопку в інтерфейсі бота з його нікнеймом або натискає на кнопку “Ввести ім’я” та записує ім’я власноруч;
- бот в подальшому буде звертатися до гравця по імені.

US3 Як користувач, я можу взаємодіяти з ботом шляхом натискання на кнопки, для того щоб обрати варіанти відповіді та/або дії.

Сценарій користувацької історії виглядає таким чином:

- бот виводить текстове поле з варіантами відповідей та очікує вибір гравця;
- користувач натискає один з варіантів відповіді;
- бот продовжує оповідання.

US4 Як користувач, я хочу мати можливість переглядати інформацію про персонажів, щоб краще розуміти історію та персонажів.

Сценарій користувацької історії виглядає таким чином:

- користувач натискає на кнопку в інтерфейсі «Інфо»;
- користувач обирає персонажа, якого хоче переглянути;
- бот виводить інформацію про персонажа, які доступні на момент оповідання;
- користувач бачить дані про персонажа.

US5 Як користувач, я хочу мати доступ до розділу про гру, щоб дізнатися більше про розробників та виробництво гри.

Сценарій користувацької історії виглядає таким чином:

- користувач натискає на кнопку в інтерфейсі «Про гру»;
- користувачу показується інформація про гру, розробників та виробництво гри.

US6 Як користувач, я хочу мати можливість залишати відгуки про гру, щоб розробники могли відповідно покращувати та оновлювати гру.

Сценарій користувацької історії виглядає таким чином:

- користувач натискає на кнопку в інтерфейсі «Залишити відгук»;

– бот виводить повідомлення та дає можливість написати відгук.

US7 Як користувач, я хочу мати доступ до глосарію термінів, що використовуються в грі, щоб я міг зрозуміти будь-яку незнайому мову або термінологію.

Сценарій користувацької історії виглядає таким чином:

- користувач натискає на кнопку в інтерфейсі «Глосарій термінів»;
- бот виводить список слів, які доступні на момент оповіді;
- користувач обирає слово, якщо воно є;
- бот виводить опис слова;
- користувач досліджує значення слова.

US8 Як користувач, я хочу отримувати сповіщення про вихід нових оновлень або розділів, щоб бути в курсі розвитку гри.

- бот розсилає сповіщення про вихід оновлення.

2.4 Визначення нефункціональних вимог до Telegram-боту

Після визначення функціональних вимог, необхідно сформулювати нефункціональні вимоги, які висуваються до Telegram-боту з візуальної новели.

NFR1 Пристрій – будь-який пристрій з підтримкою додатку «Telegram».

NFR2 Інтернет-підключення – Telegram вимагає наявності активного Інтернет-підключення для використання боту.

2.5 Проектування користувацького інтерфейсу Telegram-боту

У процесі розробки Telegram-ботів, Telegram надає обмежені можливості для створення користувацького інтерфейсу. Замість можливості реалізувати власний інтерфейс, розробники обмежені використанням наявних елементів та шаблонів, які надаються самим Telegram. У вікні запуску Telegram-бота "Polytechnic Love" присутні елементи інтерфейсу. Першим елементом є вікно, яке містить зведену

інформацію про візуальну новелу. При введенні “/start” ініціює початок чату з ботом та пропонує обрати мову гри. (рис. 2.5)

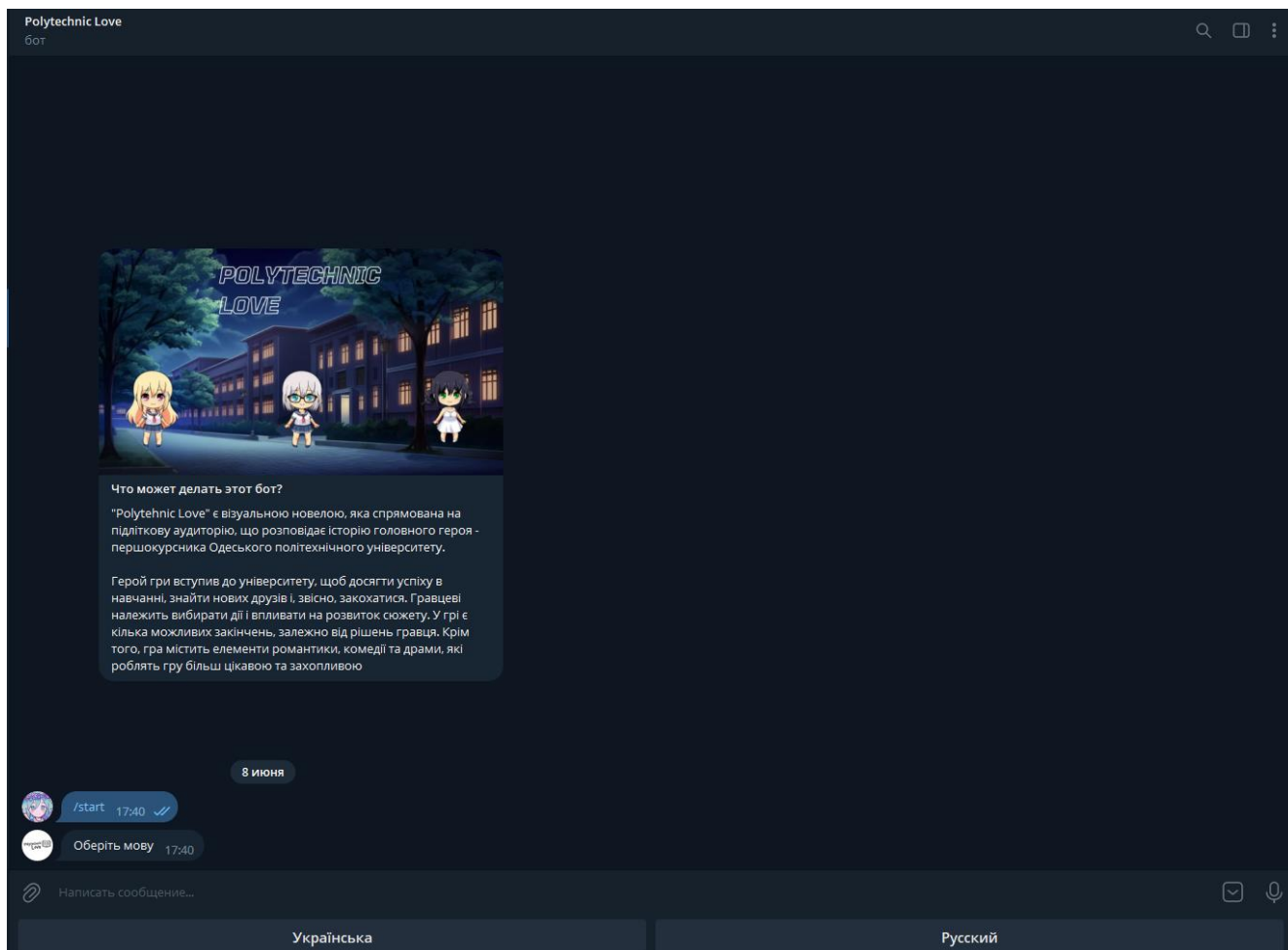


Рисунок 2.5 – Інтерфейс вікна запуску Telegram-бота

У головному інтерфейсі Telegram-бота присутні 5 основних кнопок дій: "Продовжити", "Інфо", "Глосарій", "Про гру" та "Залишити відгук". Головний інтерфейс відрізняється простим дизайном, що сприяє легкій навігації користувачів та надає доступ до всіх основних опцій у грі. (рис. 2.6)

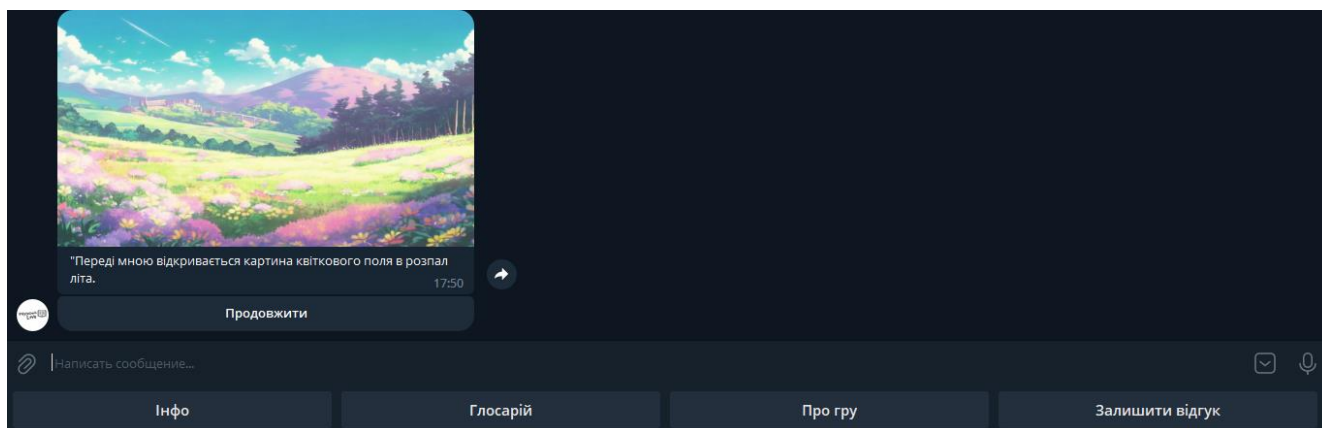
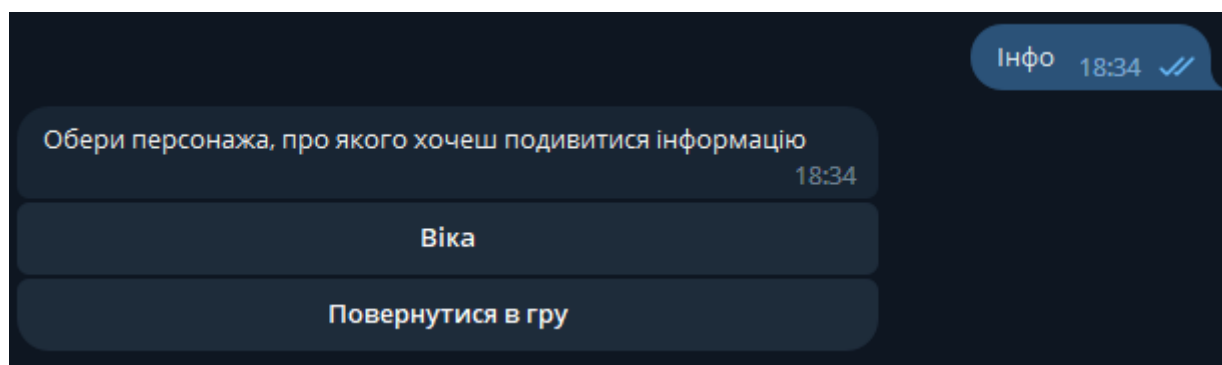
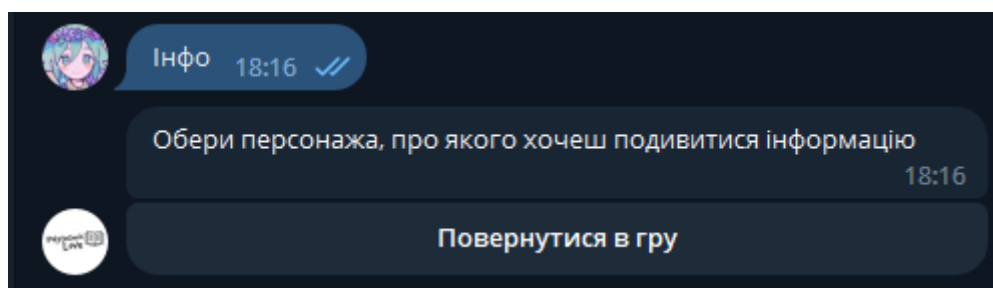


Рисунок 2.6 – Інтерфейс головного екрану Telegram-бота

При натисканні на кнопку «Інфо» виведеться вікно з вибором персонажа(-ів), на якого гравець прагне побачити інформацію. Бот відправляє інформацію про персонажа, якщо персонаж хоча б раз з'являвся в оповіданні. (рис. 2.7)



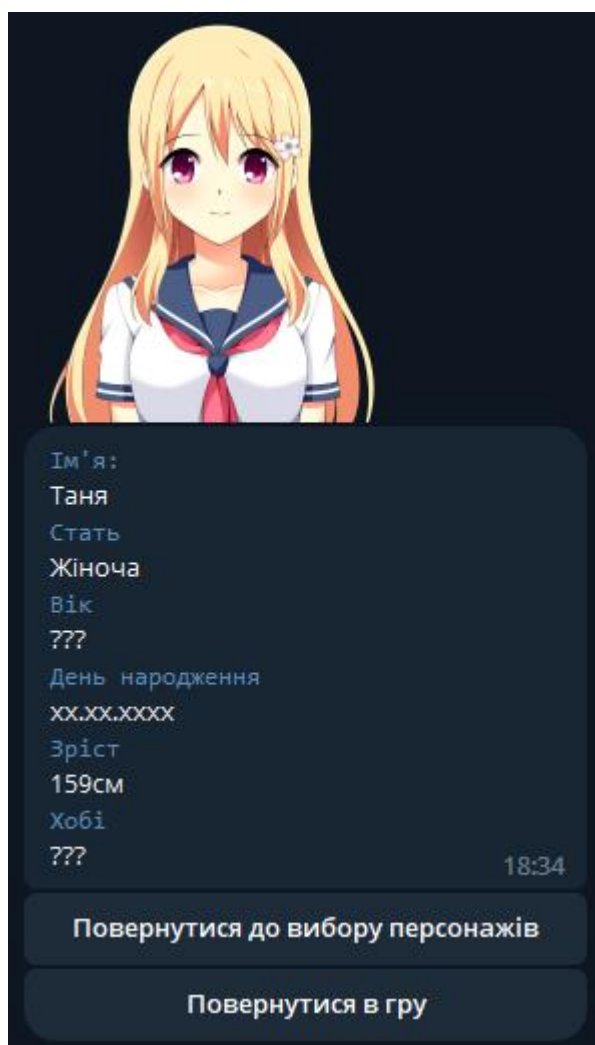
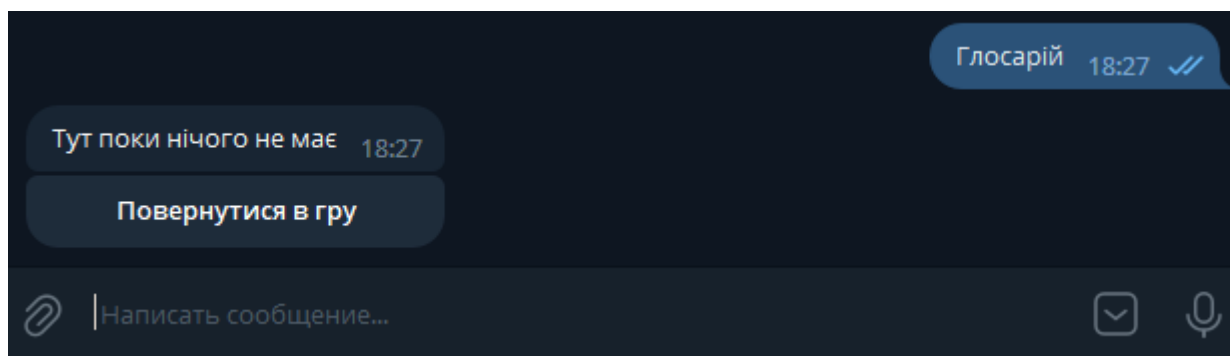


Рисунок 2.7 – Інтерфейс бота при натисканні на кнопку «Інфо»

При натисканні на кнопку «Глосарій» виведеться вікно з вибором слів, термінів на які гравець прагне побачити пояснення. Бот відправляє роз'яснення про ці слова, якщо ці слова або терміни хоча б раз зустрічалися в оповіданні. (рис. 2.8)



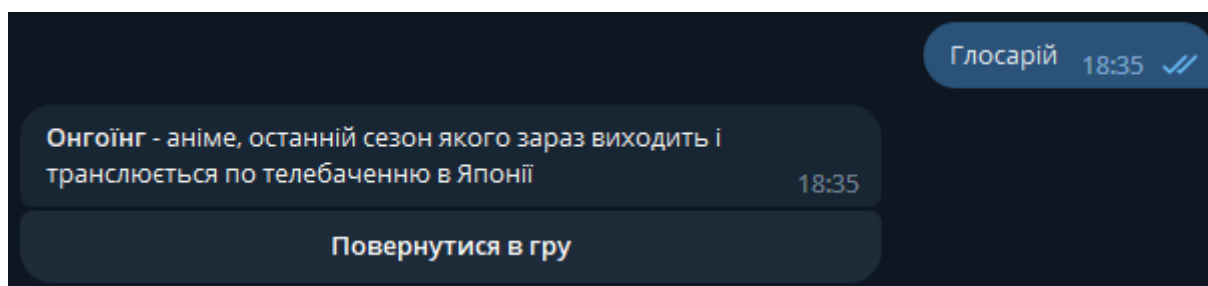


Рисунок 2.8 – Інтерфейс бота при натисканні на кнопку «Глосарій»

При натисканні на кнопку «Глосарій» виведеться вікно з інформацією про гру, а саме: назву, короткий опис гри, творців гри та їх контактні дані Telegram. (рис. 2.9)

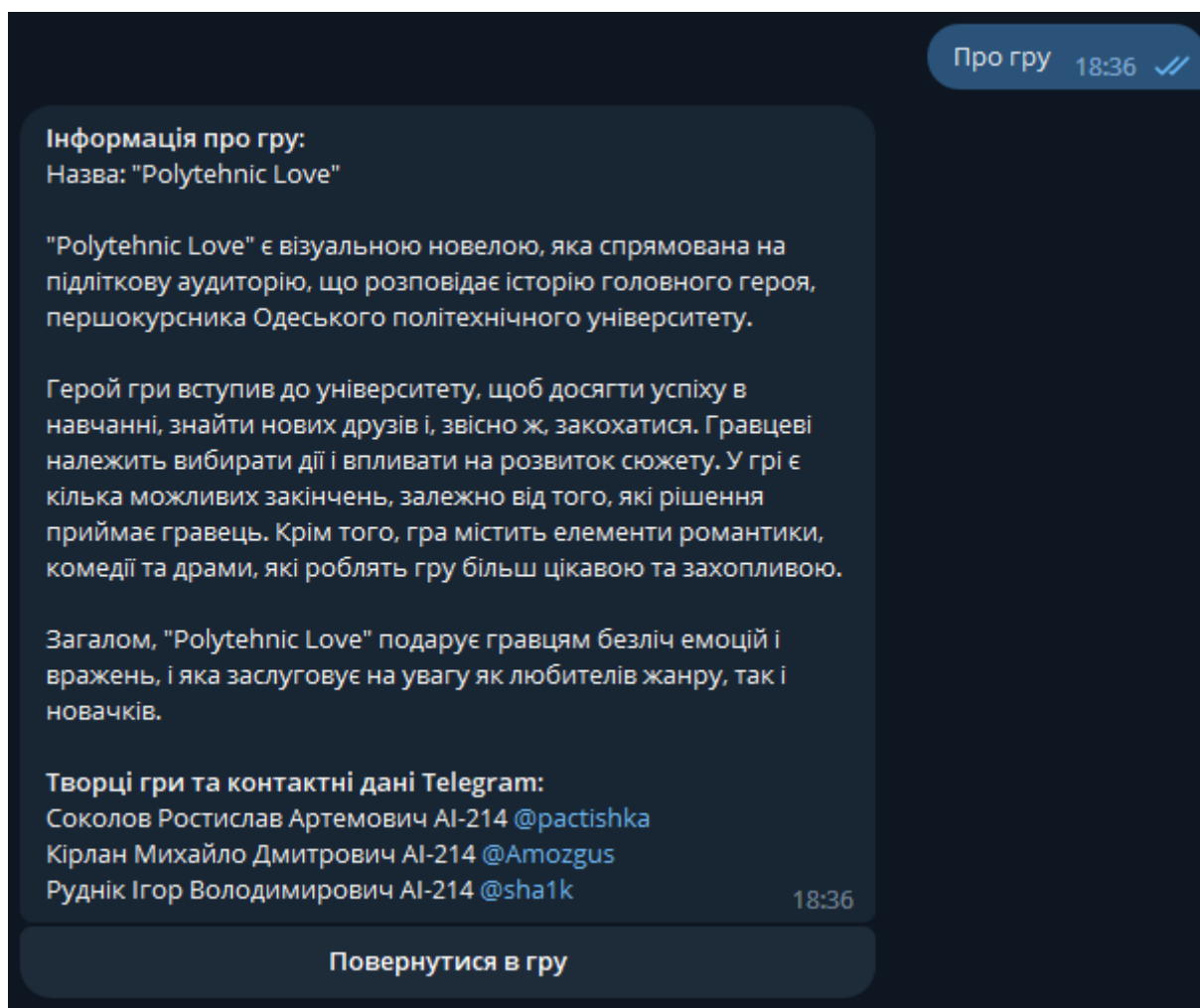


Рисунок 2.9 – Інтерфейс бота при натисканні на кнопку «Про гру»

При натисканні на кнопку «Залишити відгук» виведеться повідомлення після якого стає можливо залишити відгук розробнику для усунення багів та побажаннями до гри. (рис. 2.10)

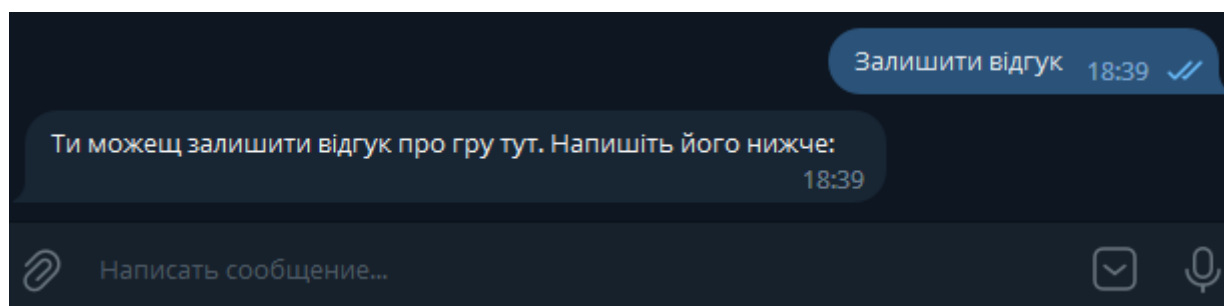


Рисунок 2.10 – Інтерфейс бота при натисканні на кнопку «Залишити відгук»

2.6 Висновки до другого розділу

У другому розділі даної курсової роботи було здійснено проектування Telegram-боту з назвою «Polytechnic Love» з використанням бібліотеки `python-telegram-bot`.

Під час проектування були визначені ціль боту, його потенційна аудиторія та основні можливості. На основі цих відомостей були сформульовані основні вимоги до Telegram-боту, включаючи функціональні та нефункціональні вимоги. Крім того, була розроблена діаграма сценаріїв використання Telegram-боту.

Для проектування користувацького інтерфейсу були розроблені інтерфейси основних дій у боті з урахуванням обмежень Telegram API.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ TELEGRAM-БОТА ДЕМО-ВЕРСІЇ ВІЗУАЛЬНОЇ НОВЕЛИ

3.1 Структура програмного проекту

Координація функціонування і запуск бота здійснюється в модулі `main.py`. Вона здійснюється за допомогою класу `ConversationHandler` з бібліотеки `python-telegram-bot` шляхом виклику відповідних методів з модуля двигуна гри `engine.py` у відповідь на певні константи

В модулі `engine.py` реалізовано всі основні методи боту:

- `set_player`
встановлення глобальних змінних `player` та `lang` на основі ідентифікатора гравця, наданого як аргумент функції `set_player`.
- `start`
налаштовує початковий потік розмови, створюючи нового гравця, завантажуючи його прогрес, пропонуючи йому вибрати мову і потенційно перенаправляючи його до нової функції на основі рівня його прогресу.
- `enterName`
обробляє вибір мови, зроблений користувачем, встановлює відповідну мову програвача, готує клавіатуру відповіді з іменем програвача та варіантом відповіді користувача, а також надсилає користувачеві повідомлення з проханням ввести своє ім'я за допомогою клавіатури відповіді.
- `handleName`
обробляє вибір імені користувача, порівнюючи його повідомлення з останнім варіантом відповіді користувача для поточної мови. Якщо вибір збігається, він надсилає спеціальне повідомлення боту і повертає `GOOD_MORNING`. В іншому випадку він перенаправляє потік на функцію привітання, повертаючи і чекаючи на неї

- greetings
обробляє введення імені гравця, встановлює ім'я та прогрес в об'єкті гравця, зберігає дані гравця, надсилає привітальне повідомлення з ім'ям гравця та повертає NOVEL, щоб вказати потоку, що він має перейти до наступного кроку, пов'язаного з історією.
- novel
керує розвитком основної історії на основі прогресу гравця та певних виборів. Він обробляє різні типи повідомлень, такі як текстові, фото, голосові та стікери, а також налаштовує кастомні розмітки з вбудованими кнопками для певних значень прогресу. Код також зберігає прогрес гравця і повертає CONTINUE, щоб показати, що історія має продовжуватися
- error
обробляє помилки у потоці розмови. Вона встановлює гравця і мову, очищає розмітку відповіді останнього повідомлення, надсилає повідомлення про помилку гравцеві, а потім перезапускає розмову за допомогою виклику функції start().
- in_development
надсилає гравцеві анімацію, яка інформує його про те, що певна функція все ще перебуває у розробці. Вона надає повідомлення про статус функції та прибирає розмітку відповіді, щоб приховати будь-яку видиму клавіатуру.
- send_anything
допоміжна функція, яка надсилає плеєру різні типи контенту (текст, фото, голос, стікер) на основі наданих аргументів (args) і повертає відповідний об'єкт повідомлення.
- info
керує інформаційним меню, дозволяючи гравцеві вибрати персонажа та переглянути інформацію про нього. Він надсилає текстові повідомлення та стікери на основі вибору гравця та його прогресу.

- info_buttons

обробляє вибір користувача в інформаційному меню. Він підтверджує запит, видаляє попереднє повідомлення бота, а потім або повертається до роману, або повертається до меню вибору символів, або показує певну інформацію про символ, залежно від вибору користувача.

- feedback

обробляє вибір користувача в меню "відгук", надсилаючи повідомлення зворотного зв'язку, а потім переходить до наступного кроку для збереження зворотного зв'язку.

- save_feedback

обробляє вибір користувача в меню зворотного зв'язку, зберігаючи відгук, надсилаючи повідомлення з підтвердженням, а потім переходить до наступного кроку для обробки вибору в меню зворотного зв'язку.

- feedback_button

обробляє вибір користувача в меню зворотного зв'язку, підтверджуючи запит, налаштовуючи гравця, а потім переходить до основної історії, викликаючи функцію novel.

- glossary

обробляє вибір користувача у меню глосарію, встановлюючи гравця, отримуючи прогрес гравця, створюючи відповідний текст глосарію, надсилаючи повідомлення глосарію до гравця і повертаючи GLOSSARY_BUTTONS для продовження подальших дій, пов'язаних з глосарієм.

- glossary_buttons

обробляє вибір користувача в меню глосарію, підтверджуючи запит зворотного виклику, встановлюючи гравця, видаляючи попереднє повідомлення, а потім повертає результат функції novel для продовження подальших дій, пов'язаних з новелою.

- about

обробляє вибір користувача в меню “про гру”, налаштовуючи плеєр, зберігаючи зворотний зв'язок, надсилаючи повідомлення-відповідь і повертаючи значення для переходу до наступного кроку в розмові.

- about_button

обробляє вибір користувача в інформаційному меню, підтверджуючи запит, налаштовуючи програвач, видаляючи останнє повідомлення і повертаючи значення для переходу до наступного кроку в розмові.

Модуль `playerinfo.py` використовує API веб-базованої бази даних `Airtable` для керування базою даних для збереження прогресу і відгуків гравців через клас `Player`

Модуль `script.py` зберігає сценарій новели на двох мовах

Модуль `links.py` зберігає посилання на медіафайли, що використовуються в новелі

Інші модулі є допоміжними і потрібні тільки для розвантаження основних. Вони зберігають токени бота і бази даних

3.2 Керування вихідним кодом Telegram-боту

Використання системи контролю версій є дуже важливим для будь-якого програмного проекту, в тому числі і для Telegram-боту. Використання системи контролю версій дозволяє зберігати історію всіх змін, які були внесені.

Це дозволяє в разі потреби повернутися до попередніх версій проєкту та відновити попередній стан, який був на момент попередньої версії.

Також, система контролю версій дозволяє контролювати версії Telegram-боту, що значно полегшує роботу з проєктом та дозволяє підтримувати його в актуальному стані.

Основні метрики репозиторію Telegram-боту наведені в таблиці 3.1.

Таблиця 3.1 – Метрики керування програмним кодом додатку

Telegram-бот	Кількість комітів	Кількість pull-реквестів	Кількість гілок у репозиторії
	19	19	1

3.3 Функціональне тестування розробленого Telegram-боту

Функціональне тестування Telegram-боту з візуальною новелою є дуже важливим процесом, оскільки воно дозволяє перевірити правильність функціонування різних функцій та можливостей цього ресурсу перед його введенням в експлуатацію.

Функціональне тестування дозволяє перевірити, що всі функції та можливості Telegram-боту з візуальною новелою працюють правильно, як очікувалося, що забезпечує коректну роботу Telegram-боту та надійність для його використання.

Також функціональне тестування дозволяє виявити та виправити помилки Telegram-боту, що можуть негативно впливати на досвід користувачів, підвищуючи їх задоволеність та вірогідність повернення до бота в майбутньому.

Для проведення функціонального тестування необхідно розробити протокол тестування.

Протокол тестування – це документ, який містить опис кроків, процедур та результатів тестування програмного продукту, системи чи окремої її частини. Протокол тестування має на меті систематично задокументувати процес тестування та отримані результати, щоб забезпечити максимальну об'єктивність та повторюваність результатів.

Для функціонального тестування Telegram-боту «Polytechnic Love» був розроблений наступний протокол тестування.

ТС1 Тест-кейс для запуску бота:

- відкрити додаток Telegram;
- увійти до чату з ботом;
- натиснути кнопку «Розпочати»;

– переконатись, що бот запустився та видає результат.

ТС2 Тест-кейс працездатності кнопки «Інфо»:

- натиснути кнопку «Інфо»;
- переконатись, що бот виводить вікно з можливість або неможливістю вибору в ньому.

ТС3 Тест-кейс працездатності кнопки «Глосарій»:

- натиснути кнопку «Глосарій»;
- переконатись, що бот виводить вікно з можливість або неможливістю вибору в ньому.

ТС4 Тест-кейс працездатності кнопки «Про гру»:

- натиснути кнопку «Про гру»;
- переконатись, що бот виводить вікно з інформацією про гру.

ТС5 Тест-кейс працездатності кнопки «Залишити відгук»:

- натиснути кнопку «Залишити відгук»;
- переконатись, що бот виводить вікно та пропонує залишити відгук.

ТС6 Тест-кейс працездатності кнопки «Повернутися в гру»:

- натиснути кнопку «Повернутися в гру»;
- переконатись, що бот виводить вікно та продовжує оповідання.

Результати функціонального тестування Telegram-боту «Polytechnic Love» наведені в таблиці 3.2.

Як можна побачити з результатів тестування, всі функціональні тести пройдені успішно, що свідчить про те, що фактична поведінка Telegram-боту співпадає з очікуваною поведінкою, визначеною у функціональних вимогах.

Таблиця 3.2 – Протокол функціонального тестування Telegram-боту «Polytechnic Love»

Номер тест-кейсу	Очікуваний результат	Фактичний результат	Результат тестування
ТС1	Успішний запуск бота	Бот успішно запустився	Успішно

ТС2	Успішне функціонування кнопки «Інфо»	Кнопка працездатна та виводить інформацію про персонажа(-ів)	Успішно
ТС3	Успішне функціонування кнопки «Глосарій»	Кнопка працездатна та виводить глосарій	Успішно
ТС4	Успішне функціонування кнопки «Про гру»	Кнопка працездатна та виводить інформацію про гру	Успішно
ТС5	Успішне функціонування кнопки «Залишити відгук»	Кнопка працездатна та пропонує залишити відгук про гру	Успішно
ТС6	Успішне функціонування кнопки «Повернутися в гру»	Кнопка працездатна та повертає користувача в оповідання	Успішно

3.4 Інструкція користувача Telegram-боту

Для забезпечення успішного користувацького досвіду використання розробленого Telegram-боту необхідно скласти інструкцію користувача, яка включає в себе знімки екранів та пояснювальний текст до кожного знімка. Це допоможе користувачам легко зорієнтуватись у функціоналі Telegram-боту та ефективно використовувати його.

Перед початком користування боту, користувача зустрічає вікно з короткою інформацією про гру (рис. 3.2).

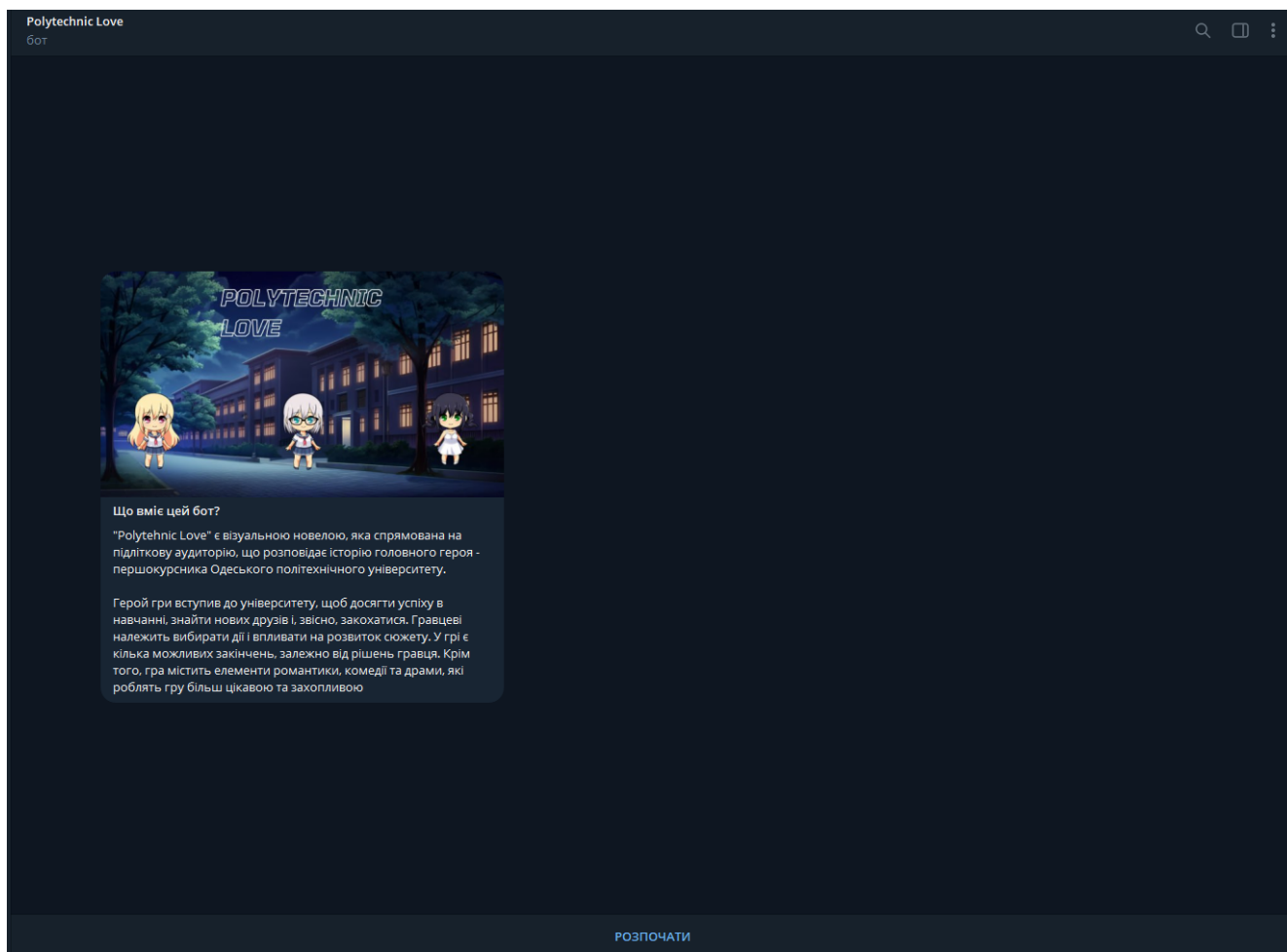


Рисунок 3.2 – Початкове вікно з інформацією про гру

Далі користувачеві пропонується обрати мову оповідання (рис 3.3).

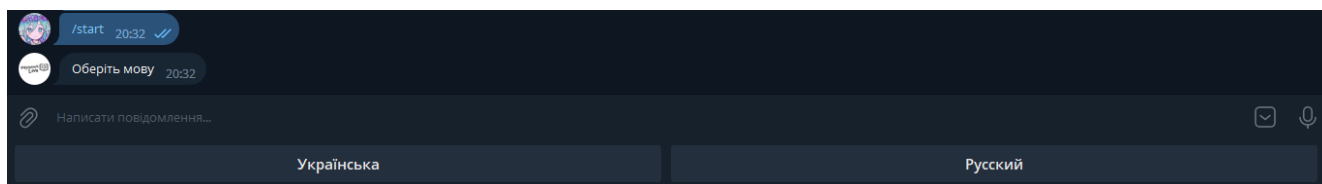


Рисунок 3.3 – Вікно обирання мови

Після чого користувачу потрібно обрати ім'я для подальшого оповідання. Пропонується два варіанти: ім'я акаунту Telegram або вказати ім'я власноруч (рис. 3.4).



Рисунок 3.4 – Вікно обирання імені

Перед початком оповідання залишилося лише натиснути кнопку “Давай!” (рис. 3.5).

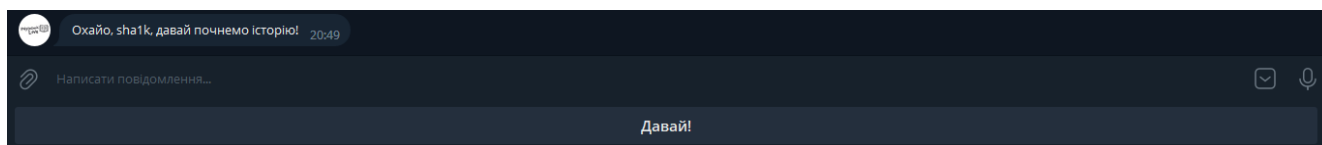


Рисунок 3.5 – Вікно перед початком оповідання

Все необхідне перед оповідання було зроблено, тепер перед користувачем зв’являється вікно з текстом та кнопкою “Продовжити” та ще чотири кнопки знизу: “Інфо”, “Глосарій”, “Про гру”, “Залишити відгук” (рис. 3.6).

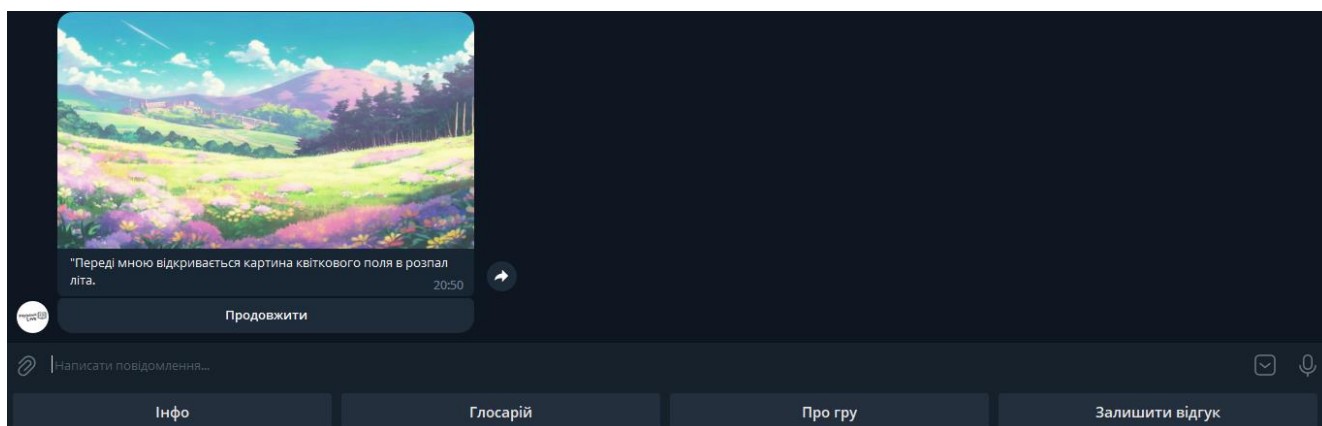


Рисунок 3.6 – Головне вікно оповідання та управління

При натисканні кнопки “Продовжити” сюжет оповідання буде йти вперед. (рис. 3.7).

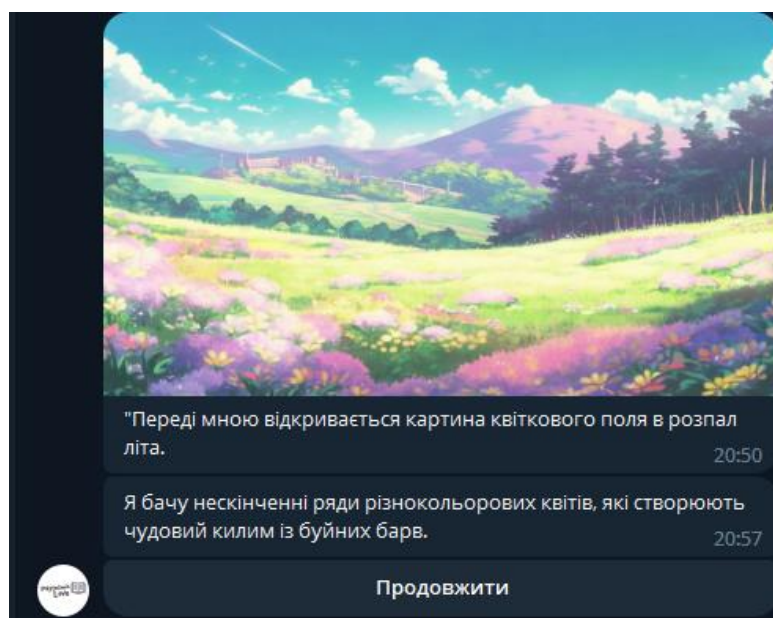


Рисунок 3.7 – Натискання кнопки “Продовжити”

В будь-який час користувач по мірі оповідання може натиснути кнопку “Інфо” та подивитися коротку інформацію про персонажа, якщо він або вона вже з’являлися в оповідання хоча б один раз (рис. 3.8).

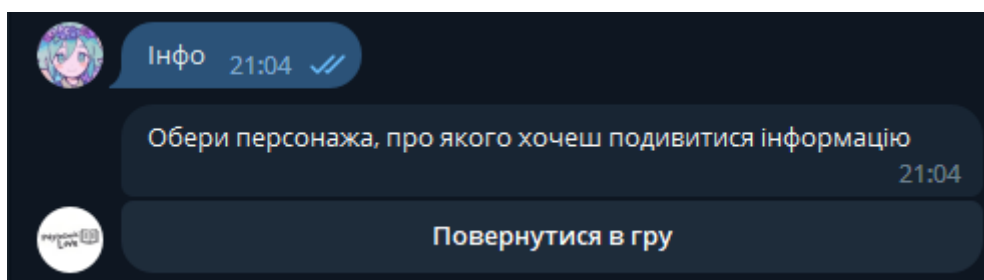


Рисунок 3.8 – Кнопка “Інфо”

Користувач може натиснути кнопку “Глосарій” та подивитися короткий опис про слів або термінів, які будуть з’являтися по мірі оповідання (рис. 3.9).

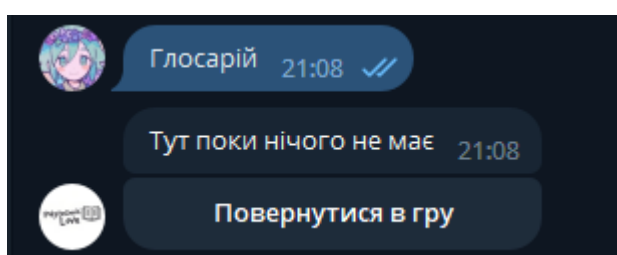


Рисунок 3.9 – Кнопка “Глосарій”

Якщо користувач захотів подивитися або нагадати собі інформацію про гру або інформацію про розробників, він може скористатися кнопкою “Про гру” (рис. 3.10).



Рисунок 3.10 – Кнопка “Про гру”

Якщо у користувача є якісь побажання або враження від оповідання, він може натиснути кнопку “Залишити відгук” та написати для розробників свій коментар (рис. 3.11).

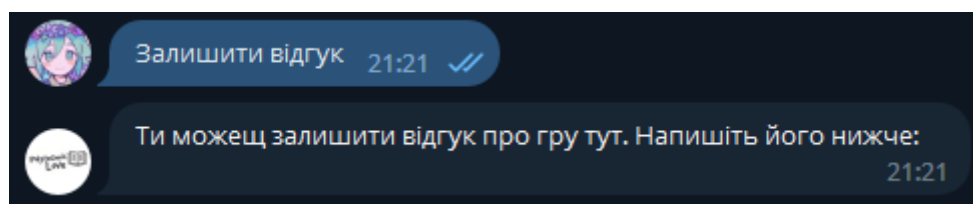


Рисунок 3.11 – Кнопка “Залишити відгук”

3.5 Вихідний код Telegram-боту

Вихідний код програмних класів розробленого Telegram-боту наведений в GitHub репозиторії даної роботи.

3.6 Висновки до третього розділу

В даному розділі курсової роботи було проведено програмну реалізацію Telegram-боту «Polytechnic Love» на мові програмування Python та бібліотеки python-telegram-bot. Для того, щоб провести програмну реалізацію, був використаний проєкт Telegram-боту, який був розроблений в другому розділі даної роботи.

Перш за все, була розглянута структура серверного програмного проєкту. Були визначені основні програмні компоненти, з яких складається програмний проєкт Telegram-боту, а також наданий перелік основних класів, які відносяться до цих компонентів.

Далі була розроблена концептуальна діаграма класів, яка дозволяє визначити основні класи Telegram-боту та визначити взаємовідносини між цими класами.

Окремо було розглянуто питання використання системи контролю версій для упорядкування розробки Telegram-боту. Були визначені основні репозиторії проєкту та надані основні показники метрик цих репозиторіїв.

Далі було проведено функціональне тестування розробленого Telegram-боту. Був розроблений протокол тестування у вигляді множини тест-кейсів та проведено тестування. Було визначено, що всі тест-кейси були пройдені, а отже фактична поведінка Telegram-боту співпадає з очікуваною поведінкою, визначеною у функціональних вимогах.

Також була розроблена інструкція користувача у вигляді множини знімків екрану та пояснювального тексту, який визначає функціонал додатку у даному вікна та можливі дії користувача. Крім того, наданий вихідний код розробленого Telegram-боту.

ВИСНОВКИ

У даній курсовій роботі був розроблений Telegram-бот «Polytechnic Love» на мові програмування Python, на бібліотеці python-telegram-bot.

Такий Telegram-бот дозволяє користувачам поглинути у цікавий світ візуальної новели в додатку Telegram. Таким чином, мета, поставлена перед даною курсовою роботою досягнута в повному обсязі. Для досягнення мети були вирішені наступні задачі.

У першому розділі даної курсової роботи була детально розглянута предметна область створення Telegram-боту «Polytechnic Love» на бібліотеці python-telegram-bot. Були визначені основні завдання, які повинні бути вирішені в процесі розробки Telegram-боту з візуальної новели. Був проведений аналіз існуючого аналогу та визначені ключові вимоги до створення власного Telegram-боту. У якості технологій для розробки клієнтської частини був обраний Telegram. Було вирішено також використовувати PostgreSQL в якості СУБД для даного Telegram-боту.

У другому розділі роботи було проведено проектування Telegram-боту. У процесі проектування була визначена мета Telegram-боту, його потенційна аудиторія та основні можливості. На основі цих даних були встановлені основні вимоги до Telegram-боту, включаючи функціональні та нефункціональні. Також була розроблена діаграма сценаріїв використання даного Telegram-боту. З метою проектування інтерфейсу користувача були розроблені інтерфейси основних дій у боті з урахуванням обмежень Telegram API.

В третьому розділі роботи було проведено програмну реалізацію Telegram-боту. Розглянута структура серверного програмного проекту, визначені основні програмні компоненти, розроблена діаграма класів. Розглянуто питання використання системи контролю версій для упорядкування розробки Telegram-боту. Проведено функціональне тестування, розроблена інструкція користувача у вигляді множини знімків екрану та пояснювального тексту та наданий вихідний код розробленого Telegram-боту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. python-telegram-bot. URL: <https://github.com/python-telegram-bot/python-telegram-bot/wiki> (дата звернення 15.04.2023).
2. Airtable. URL: <https://www.airtable.com/> (дата звернення 15.04.2023).
3. ChatGPT. URL: <https://openai.com/blog/chatgpt> (дата звернення 15.04.2023).
4. Habr. URL: <https://habr.com/ru/all/> (дата звернення 15.04.2023).
5. HashDork. URL: <https://hashdork.com/uk/> (дата звернення 15.04.2023).
6. YouTube. URL: <https://www.youtube.com/> (дата звернення 15.04.2023).
7. Docs-Python. URL: <https://docs-python.ru/> (дата звернення 15.04.2023).