

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра програмних систем і технологій

УДК _____

На правах рукопису

ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Тема: “ Розробка 3D комп’ютерної гри жанру рольова гра з використанням
Telegram ботів ”.

Спеціальність: 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

БР.ІПЗ-_____

Студент групи ІПЗ-43

Андрій ДЕВЛИШ

(підпис) (розшифровка підпису)(дата)

Науковий керівник

к. т. н., доц. Катерина МЕРКУЛОВА

(посада) (підпис) (дата) (розшифровка підпису)

Консультант з питань нормоконтролю

Тамара ЧАПОВСЬКА

Допускається до захисту

з питань нормоконтролю

Завідувач кафедри

проф., д.т.н., доц. Олексій БИЧКОВ

(посада) (підпис) (дата) (розшифровка підпису)

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра програмних систем і технологій
Спеціальність 6.050103 Програмна інженерія

ЗАТВЕРДЖЕНО

Завідувач кафедри

програмних систем і технологій

_____ (Олексій БИЧКОВ)

„___” _____ 2021р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ
СТУДЕНТУ**

Девлишу Андрію Владиславовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної бакалаврської роботи «Розробка 3D комп'ютерної гри жанру рольова гра з використанням Telegram ботів»
затверджена наказом вищого навчального закладу від „___” _____ 20__ р. № _____

2. Строк здачі студентом закінченої роботи _____

3. Вихідні дані до роботи _____

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)

5. Перелік графічного матеріалу (з точним забезпеченням обов'язкових креслень)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 13 жовтня 2019 р.

Керівник _____ (Катерина МЕРКУЛОВА)

Завдання прийняв до виконання _____ (Андрій ДЕВЛИШ)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назви етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Збір інформації	24.10.2020	
2	Вивчення варіантів реалізації та вибір варіанту для розробки	15.01.2021	
3	Аналіз аналогів	03.02.2021	
4	Проектування гри та сюжету	19.02.2021	
5	Розробка графічної складової гри	12.03.2021	
6	Реалізація Telegram ботів	28.03.2021	
7	Написання бази даних	25.04.2021	
8	Аналіз готового продукту та оформлення дипломної роботи	31.05.2021	

Студент – бакалавр _____ (Андрій ДЕВЛИШ)

Керівник роботи _____ (Катерина МЕРКУЛОВА)

АНОТАЦІЯ

Випускна кваліфікаційна бакалаврська робота: с.54, рис. 38, додат.7, джерел.20

Тема: Розробка 3D комп'ютерної гри жанру рольова гра з використанням Telegram ботів

Об'єкт дослідження: Сучасні 3D ігри жанру RPG, екосистема Telegram Bot

Мета роботи: Проаналізувати актуальні аналоги та Розробити 3D гру, жанра RPG з використанням Telegram Bot

АННОТАЦИЯ

Выпускная квалификационная бакалаврская работа: с.54, Рис.38, Доп.7,
Источники.20

Тема: Разработка 3D компьютерной игры жанра ролевая игра с использованием Telegram ботов

Объект исследования: Современные 3D игры жанра RPG, экосистема Telegram Bot

Цель работы: Проанализировать актуальные аналоги и разработать 3D игру, жанра RPG с использованием Telegram Bot

ANNOTATION

Final qualifying bachelor's work: p.54, Pic.38, Add.7, Sources.20

Topic: Development of a 3D computer game of the role-playing game genre using Telegram bots

Research object: Modern 3D RPG games, Telegram Bot ecosystem

Purpose of work: Analyze current analogues and develop a 3D RPG game using Telegram Bot

ЗМІСТ

Перелік умовних позначень.....	6
Вступ.....	7
РОЗДІЛ 1	
ОГЛЯД СУЧАСНОГО РИНКУ	
1.1 Огляд сучасного ринку.....	8
1.2 Аналоги.....	8
1.3 Висновки.....	13
РОЗДІЛ 2	
ПРОЕКТУВАННЯ	
2.1 Постановка завдання.....	14
2.2 Обґрунтування методів реалізації.....	15
РОЗДІЛ 3	
АРХІТЕКТУРА ПРОЕКТУ	
3.1 Проектування Telegram Бота.....	22
3.2 Внутрішня архітектура Telegram Бота.....	26
3.3 Написання Unity гри.....	29
3.4 Проектування бази даних.....	34
3.5 Взаємозв'язок між усіма компонентами.....	40
3.6 Програмна інженерія.....	44
ВИСНОВКИ.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46
ДОДАТКИ.....	47

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

БД - база даних

SQL - мова структурованих запитів

RPG - Role-playing game

ГГ-Головний Герой, центральна фігура сюжету гри

API - Application Programming Interface, інтерфейс програмування

ВСТУП

Актуальність теми полягає у тому, що ігрова індустрія має безліч жанрів, під жанрів. На кожен якісно сконструйований ігровий продукт знайдеться своя цільова аудиторія. Під жанром кваліфікаційної роботи є жанри: візуальний роман, пригода.

Цей напрямок на момент написання кваліфікаційної роботи є популярним на території України

Зв'язок роботи з науковими програмами полягає у вдалому прикладі реалізації комплексу використання практичних знань з різних предметів, які було засвоєно під час навчання

Методи дослідження були наступними:

1. Візуальний аналіз сучасного ринку відеоігор, та знаходження цільової аудиторії
2. Детальний аналіз обраного жанру та виявлення основних ігрових паттернів шляхом тестування аналогічної продукції

Новизна одержаних результатів - Інтеграція соціальних мереж(у данному випадку месенджер TELEGRAM) у ігровий процес, чого раніше не було помічено у інших проектах.

Сучасний ринок TELEGRAM ботів надає схожі продукти, але основна їх проблема полягає у відсутності візуальної частини, які вирішує дане програмне рішення

Апробація результатів випускної кваліфікаційної бакалаврської роботи

--

Публікації

РОЗДІЛ 1

ОГЛЯД СУЧАСНОГО РИНКУ

1.1 Аналіз аналогів

Основним джерелом натхнення для проекту були ігри жанру: Adventure, розмовні, RPG.

Найпопулярніших представників жанру і за сумісництвом ідейних натхненників, можна виділити:

- Нескінченне літо(Endless summer)



Рис. 1.1 Кадри з гри Нескінченне літо

Нескінченне літо - комп'ютерна гра в жанрі візуального роману або ж візуальна новела.

Гра була орієнтована на пост-СНД країни, зокрема: Росія, Україна, Білорусь, що і призвело до колосального успіху в даних країнах, гра по сей день вважається дуже популярною в вузьких колах, нехай і була випущена в 2014 році.

Основний упор робиться на знайомство гравця зі світом гри і розвитку відносин з обраними персонажами.

Ставка на ностальгію і наближені до реальності образи персонажів позитивно позначилися на продажі гри і забезпечило їй звання «Класики» серед українських споживачів.

Гра піднімає питання розуміння себе, а також всю складність людських взаємин, що близько більшості гравців цієї гри.

- stardew valley



Рис. 1.2 stardew valley

Гра в жанрі інді, RPG, квест.

Розроблена і випущена в 2016 році, навпаки, націлена на глобальний ринок і охоплення аудиторії у неї куди ширший.

У центрі оповідання стоїть ГГ котрий потрапляє в село і може вибрати свій власний шлях розвитку, який включає в себе:

- Вибір основного ремесла (рибалка, видобуток рідкісних ресурсів, сільськогосподарська діяльність і т.д.)
- Просування нелінійного сюжету на свій розсуд.

- Вибір взаємовиключної гілки прогресу в діалогових системах, розвиток відносин з численними персонажами гри.

Основними «гачками» гри стали:

1. Медетатівний ігровий досвід.
2. Сопроводжуючий Саундтрек гри.
3. Широкій охоплення цільової аудиторії.
4. Різноманітність дій.

Основною відмінністю від інших аналогів можна виділити аномально тривалий сюжет, який може тривати від 53 годин до 160 реальних годин, що вкрай сприяє зануренню в світ гри, оскільки проходження гри може розтягнутися на місяці, у гравця є багато часу на детальне занурення в світ гри, що в свою чергу провокує позитивну залежність і бажання продовжити ігровий процес.

- Привіт незнайомцю!

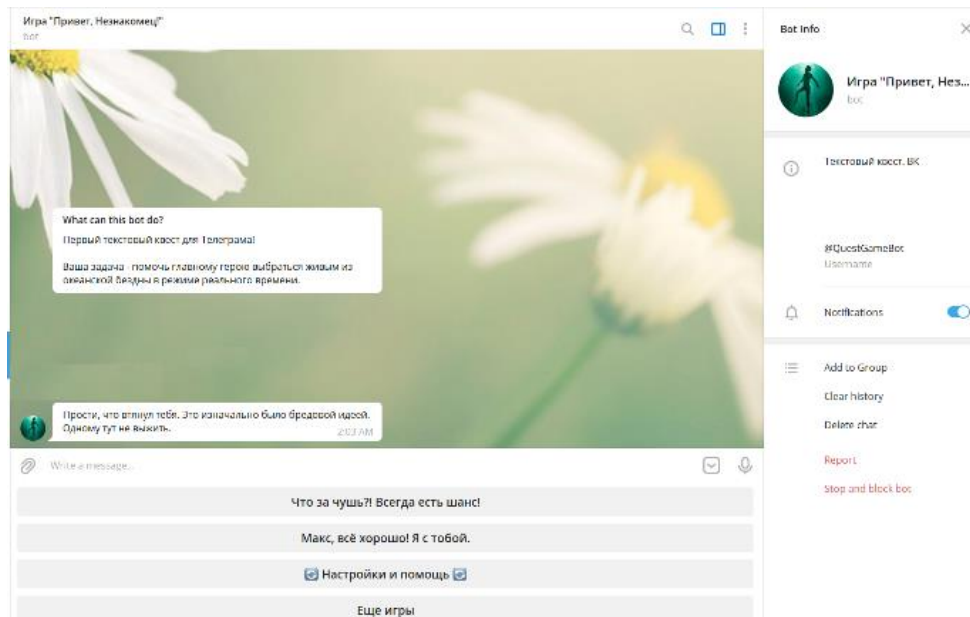


Рис. 1.3 Гра привіт незнайомцю!

Гра жанру текстовий квест, написання і реалізованості в середовищі Telegram Bot. На момент випуску була першопроходьцем цього жанру, в швидко розвивається Месенджер.

Основними проблемами жанру «текстовий квест» в сучасних реаліях були:

1. Відсутність професійно написаного тексту, який буде цікавий гравцеві, сучасна індустрія відеоігор робить упор на візуальну складову, ніж на велику кількість тексту.
2. Відсутність належної взаємодії гри з гравцем
3. Потеря можливої конкурентоспроможності на тлі сучасних і більш популярних жанрах.
4. Не інтерактивний геймплей.

Як яскравий приклад проблеми з першого пункту можна виділити Disco Elysium.



Рис. 1.4 Disco Elysium

Гра мала приголомшливий успіх у критиків, продукт можна назвати проробленим і різноманітним, але перевантаженість текстової складової не дозволила новим користувачам в повній мірі відчувати всі переваги гри, що і призвело гру до комерційного провалу.

Гра «Привіт, незнайомець» адаптувала вельми старий жанр під нові реалії, врахувавши невдалий досвід своїх попередників, що принесло їй успіх, нехай гра і вважається безкоштовною і не несе за собою комерційної вигоди.

1.2 Проміжні висновки

Всі перераховані вище гри можна в повній мірі назвати вдалими, вони успішно займають свою ігрову нішу, проте на мій рахунок упускають деякі можливості.

1.Гра «Привіт Незнайомець» нехай і стала популярною, але в силу того, що вона уособлює тільки свій власний жанр, і «замкнені» в рамках середовища Telegram Bot, все ж не дозволила їй приманити до себе більшу частину цільової аудиторії.

Більшість сучасних ігор є багатожанрова, що сприяє різноманітності, що в свою чергу сприяє більшому охопленню цільової аудиторії.

2.Гра «Нескінченне літо» нехай і є ідеальною грою свого жанру, якщо така термінологія має місце бути, все одно обмежена в своєму жанрі, так само її ігрова складова при належному завзятості проходиться за 2-3 години, що вкрай заважає зануренню.

3.Гра «Stardew Valley» перевантажена зайвими ігровими механіками, деякі аспекти якої можна характеризувати як «головний біль», як цитує відоме ігрове видання IGN "механіка боїв - в шахтах неповороткість героя постійно обертається головним болем" з чим не можна не погодитися.

РОЗДІЛ 2

ПРОЕКТУВАННЯ

2.1 Постановка завдання

Завданням даної роботи-розробка технічно-демонстраційній версії гри.

Технічно-демонстраційна версія гри буде включати в себе:

1.готовний архітектурний скелет.

2. Візуальне оформлення, придатне для демонстрації кінцевим користувачам і потенційним споживачам.

3.Повністю робочі про протестовані основні ігрові механіки.

Як було описано в попередніх пунктах, всі представлені аналоги мали як і свої плюси, так і свої мінуси.

Так само в завдання входить усунення більшості недоліків, розробка технічно-демонстраційній версії гри, яка зможе стати інноваційною в своєму жанрі, а саме, комбінація всіх перерахованих вище прикладів в сукупності з розробленими архітектурними рішеннями дозволить створити унікальний продукт, унікальність якого полягатиме в наступних пунктах:

1. Взаємодія гравця не тільки під час самого ігрового процесу, а й в будь-який час протягом доби, що створить, на момент проектування роботи, неповторний ігровий досвід.

2. Пряма взаємодія між ігровими Telegram ботами, на момент проектування роботи, всі відомі інтерективні Боти спілкуються на пряму з гравцем, дане рішення може підштовхнути на розвиток екосистеми ігрових ботів месенджера Telegram.

2.2 Обґрунтування методів реалізації

Основні стовпи кваліфікаційної роботи є:

- Unity3D
- Telegram Bot
- MySQL

Unity3D

На питання про причини вибору даного ігрового движка можна відповісти наступним чином.

Вибір даного ігрового движка обґрунтований рядом наступних чинників:

1. Універсальність

Написання основної частини гри відбувається на кросплатформенних мовою C #, який в совакупность з кросплатформенних Unity, підходить під поставлену задачу, в порівнянні з іншими поширеними ігровими движками.

2. Доступність

Оскільки ігровий движок надає безкоштовний версію, витрати на розробку і підтримку зводяться до мінімуму, що необхідно для розробки гри.

3. Сообщество

За тривалі роки існування ігрової платформи, сформувався великий фундамент підтримки початківців користувачів, що містить як і відповіді на більшість часто задаються, так і велику аудиторію фахівців, наявність яких може посприяти вирішенню більш комплексних проблем

Так само варто згадати про наявність платформи Asset Store під управлінням компанії Unity Technologies, яка надає велику колекцію готових рішень на низькорівневі проблеми.

Якщо підводити підсумки, то можна констатувати, що Unity3D підходить з причин низьких витрат часу, по відношенню до результативності, яка задовольняє потреби роботи.

Telegram Bot

На території України месенджер Telegram користується значною популярністю серед вікового контингенту від 12 до 30 років, аналогічно можна сказати і про інші країни. Така популярність забезпечує борний зростання даної платформи, що в свою чергу веде до появи нових рішень, одним з яких і являється Telegram Bot.

Це відносно нове рішення має вкрай різноманітний функціонал, простота використання і модифікація якого має вкрай широкий спектр застосування.

Одним таких застосувань є Telegram Bot гри. Дане застосування не можна назвати новим або інноваційним в екосистемі Telegram, але це швидкозростаюча галузь, яка вже вкрай успішно себе зарекомендувала в очах користувачів.

До додаткової причини вибору саме цієї платформи можна віднести вкрай розвинену і доступну базу підтримки початківців розробників.

Проектування Telegram Бота можливо на практично всіх сучасних мовах програмування, така універсальність обумовлюється архітектурою Telegram Bot API, яка має безліч реалізацій в багатьох популярних мовах програмування у вигляді сторонніх бібліотек.

Натиними мовами програмування вважаються:

1.PHP.

2.Python.

3.C #.

Розробка Telegram Бота буде проводитися на мові C #, через більш зручної взаємодії з ігровим движком Unity, а так само по приченен того, що дана мова надає можливість розробки проекту з Об'єктно-орієнтованою архітектурою.

Якщо підводити підсумки, то можна констатувати, що Telegram Bot є на сьогоднішній день вдалим рішенням для реалізації даного проекту, такі висновки можна зробити з наявності і обізнаності про даному методі переважна більшість цільової аудиторії гри

RPG

Рольові ігри, а в нашому випадку саме західні рольові відеоігри - це рольові відеоігри, розроблені в західному світі, включаючи Америку та Європу. Вони виникли в університетських комп'ютерних системах мейнфреймів у 1970-х роках, пізніше були популяризовані такими назвами, як "Ultima" та "Wizardry" на початку та в середині 1980-х років, і продовжують випускатися для сучасних домашніх комп'ютерних та відеоігрових консолей. «Золотий вік» жанру стався в середині-кінці 1980-х, а його популярність зазнала спаду в середині 1990-х, коли розробники намагалися не відставати від зміни моди, еволюції обладнання та збільшення витрат на розробку. Пізніша серія ізометричних рольових ігор, опублікована Interplay Productions та Blizzard Entertainment, була розроблена протягом більш тривалого періоду часу та встановила нові стандарти якості виробництва.

Комп'ютерні рольові ігри (CRPG) знову користуються популярністю. Нещодавні назви, такі як серії Mass Effect від BioWare та серія The Elder Scrolls від Bethesda Softworks, були створені для консольних систем і отримали багатоплатформні випуски, хоча ігри, що розробляються самостійно, часто створюються як ексклюзив для персонального комп'ютера (ПК). Розробники рольових ігор постійно експериментували з різними графічними перспективами та стилями гри, такими як системи обліку часу в режимі реального часу та покрокові, аксонометричні та графічні проекції від першої особи, а також партії, що складаються з одного персонажа чи багато символів. До піджанрів належать екшен-рольові ігри, розбійники та тактичні рольові ігри.

Технічна вишуканість, необхідна для створення сучасних відеоігор, і високі очікування гравців (принаймні, з точки зору кількості та якості передачі голосу та збільшення графічної вірності) ускладнюють для незалежних розробників враження на аудиторію в тій же мірі, що і велика виробники ігор з великими бюджетами та командами розробників можуть.

Але інновації та якість не обов'язково повинні стримуватися. Як і кіноіндустрія, "інді" (скорочена від "незалежна") сцена відеоігор відіграє вирішальну роль у формулюванні нових ідей та концепцій, які основні видавці та маркетингові відділи, застрягаючи у своїх жорстких застарілих способах, можуть вважати занадто непрацездатними чи радикальними, але пізніше прийняти.

Є багато прикладів фільмів, які так і не отримали схвалення в корпоративних рамках, які в кінцевому підсумку стали фінансово успішними та / або знаковими серед кіноглядачів.

У новому тисячолітті з'явилася низка незалежно опублікованих RPG для ПК, а також ряд CRPG, розроблених в Європі та спрямованих далі на схід, що призвело до того, що деякі люди називали Східну Європу "осередком" розвитку RPG за останні роки .

Саме таку логіку планується імпортувати до проекту, бо вона гарно себе зарекомендувала, як можна бачити.

MySQL

Протягом десятиліть переважною моделлю даних, яка використовувалася для розробки додатків, була реляційна модель даних, яка використовувалася реляційними базами даних, такими як Oracle, DB2, SQL Server, MySQL і PostgreSQL. Лише в середині-кінці 2000-х років інші моделі даних почали отримувати широке поширення і використання. Для диференціації та класифікації цих нових класів баз даних і моделей даних був введений термін "NoSQL". Часто термін "NoSQL" використовується як взаємозамінний з "нереляційні".

Незважаючи на те, що існує багато типів баз даних NoSQL з різними функціями, в таблиці нижче відмінності між базами даних SQL і NoSQL.

Кожна система має у собі, в тому чи іншому вигляді базу даних.

На сьогодні існують 2 найпопулярніших типів Бази Даних :

- NoSQL DB

Бази даних NoSQL використовують різноманітні моделі даних для доступу та управління інформацією. Такі типи баз даних оптимізовані спеціально для програм, які потребують значного обсягу даних, низької затримки та гнучких моделей даних, що досягається за рахунок послаблення деяких обмежень на узгодженість даних інших баз даних.

Цей тип бази даних має свої переваги, серед яких можна виділити :

1. Універсальність: Бази даних NoSQL, як правило, надають гнучкі рішення, що дозволяють пришвидшити та більш ітеративно розробляти. Гнучка модель даних робить бази даних NoSQL гарними для напівструктурованих та неструктурованих даних.

Прикладом таких додатків можуть служити месенджери, наприклад

Telegram, VK, Facebook, тощо.

2. Масштабованість: Бази даних NoSQL, як правило, розроблені для збільшення за допомогою розподілених кластерів обладнання, а не масштабування через додавання нових та значно ресурсозатратних(як і в матеріальному, так і в людино/годинному плані) серверів. Деякі хмарні провайдери обробляють ці операції за кадром як повністю керовану послугу.
3. Коефіцієнт корисної дії: бази даних NoSQL оптимізовано для конкретних моделей даних та шаблонів доступу, які надають більш високу продуктивність, ніж спроби виконати подібну функціональність за допомогою реляційних баз даних.
4. Якісна інтеграція з сервісами на приклад Google, Amazon, тощо.

Це дозволяє використовувати такі фічі, як: авторизація за допомогою сервісів, загрузка контактів з вищеперерахованих соціальних мереж, тощо

Але нажаль така база даних не підходить під цей проект, оскільки архітектура проекту подразумеває використовувати саме SQL DB.

- SQL DB

Бази даних SQL підтримують SQL - специфічну для домену мову для запитів і обробки даних в реляційній базі даних. Термін "реляційний" в реляційній базі даних відноситься до "реляційної моделі" управління даними, розробленої дослідником ІВМ Е.Ф. Кодом на початку 1970-х років і популяризована в ряді наступних систем баз даних, починаючи з System R.

Ключем до реляційної моделі є абстрагування даних у вигляді набору кортежів, організованих в відносини, що дозволяє абстрагуватися над фізичним представленням

даних і шляхів доступу. Хоча SQL не єдина можлива мова для реалізації запиту над реляційної моделлю - насправді він не відповідає строго оригінальному дизайну Кодда - він, безумовно, самий популярний.

Бази даних SQL і реляційні бази даних стали галузевим стандартом з кінця 1970-х років, хоча їх так звані "навігаційні" попередники, такі як IMS часів Аполлона, в деяких випадках все ще активно розробляються. Більшість популярних "корпоративних" систем є прямими нащадками System R і успадковують значну частину її конструктивних обмежень.

Основні переваги SQL:

- Зменшення обсягу зберігання даних завдяки нормалізації та інших можливостей оптимізації. Часто це призводить до кращої роботи і більш ефективного використання ресурсів.
- Сильна і добре зрозуміла семантика цілісності даних завдяки ACID (атомність, послідовність, ізоляція, довговічність).
- Стандартний доступ до даних через SQL.
- Як правило, більш гнучка підтримка запитів, здатна обробляти більш широкий діапазон робочих навантажень. SQL абстрагується від базової реалізації і дозволяє механізму оптимізувати запити відповідно до їх подання на диску.

Існує багато різних систем керування базами даних, серед яких можна виділити:

1. MySQL.
2. Microsoft SQL Server.
3. PostgreSQL.
4. Oracle.

Та багато інших

Мною була обрана саме MySQL, оскільки:

- MySQL має гарну гнучкості, яка полегшує ефективне управління глибоко вбудованими додатками навіть в гігантських центрах обробки даних, в яких зберігаються величезні обсяги критично важливої інформації. Він забезпечує повну настройку для задоволення унікальних вимог бізнесу електронної комерції з набагато меншими розмірами. MySQL забезпечує максимальну гнучкість платформи для підприємств, яким потрібні додаткові функції і можливості для своїх серверів баз даних.
- Захист конфіденційної ділової інформації. MySQL забезпечує безпеку даних з винятковими функціями захисту даних. Потужне шифрування даних запобігає несанкціонований перегляд даних, а підтримка SSH і SSL забезпечує більш безпечні з'єднання. Він також має потужний механізм, який обмежує доступ до сервера для авторизованих користувачів і має можливість блокувати користувачів навіть на рівні людина-машина. Нарешті, функція резервного копіювання даних спрощує відновлення на певний момент часу.

РОЗДІЛ 3

АРХІТЕКТУРА ПРОЕКТУ

Проектування Telegram Бота

Початок роботи з Telegram Ботом

Як було сказано раніше Telegram Bot в сучасних реаліях вважається вельми популярним і високорозвиненим продуктом, за тривалий час розвитку на різних платформах утворилися певні патерни архітектури.

Оскільки мною була вибрана мова розробки C # було прийнято рішення взяти за основу поточні методи реалізації.

В основі будь-яких Telegram ботів лежить офіційний Telegram Bot API, піднесений або ж у вигляді базової архітектури додатку в поєднанні з HTTP запитами до серверів Telegram, або ж у вигляді тієї ж базової архітектури додатку, але з використанням сторонніх бібліотек, які працюють безпосередньо з Telegram Bot API.

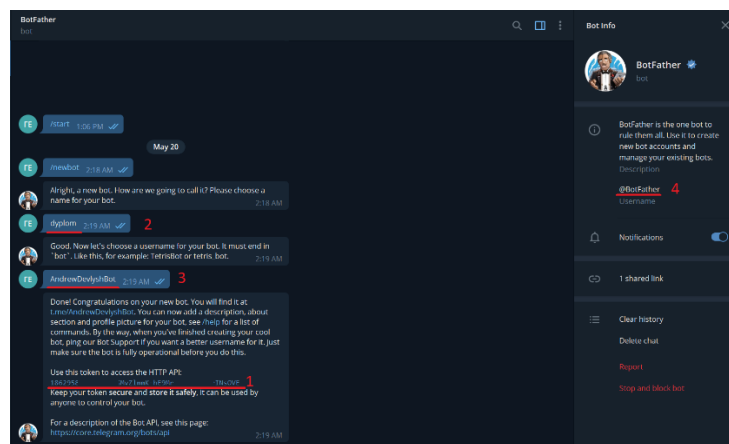


Рис.3.1 Bot Father

Першим етапом в розробці Telegram Bot стала реєстрація нового бота, що відбувалося через офіційного Telegram Бота під ім'ям BotFather.

1-унікальний API токен, він буде ключем, який дозволяє стороннім бібліотекам коректно функціонувати, а так само є надійною системою захисту, адже той, то отримує

доступ до цього ключа має всі права на модифікацію, видалення і зміна поточної логіки бота, так само , потенційний зловмисник може використовувати токен всередині свого проекту, що може привести до катастрофічних наслідків.

З міркувань безпеки частина токена була прихована, кількість відсутніх символів унеможливає отримання доступу до свіжезареєстрованому боту.

2-Ім'я створеного бота, воно повинно бути унікальним, пізніше це ім'я буде фігурувати при роботі з ботом (мал7) .

3-Унікальний NickName, за допомогою цього никнемо можна знайти бота за допомогою сторонніх джерел, не боятись потрапити на фейковий бота, пізніше він так само як і ім'я бота буде фігурувати в самому боті (рис7) .

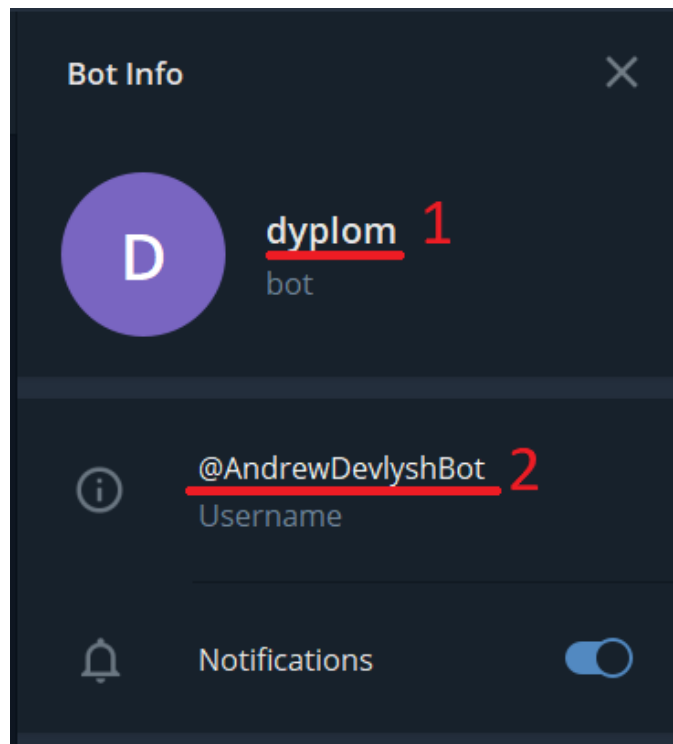


Рис.3.2 Bot Info

Наочний приклад різних типів пошуку:

1-По імені.



Рис.3.3 Пошук по імені

2-По Нік нейму.



Рис.3.4 Пошук по нік нейму

Як вже було сказано раніше мною буде використана стороння бібліотека:
Telegram Bot

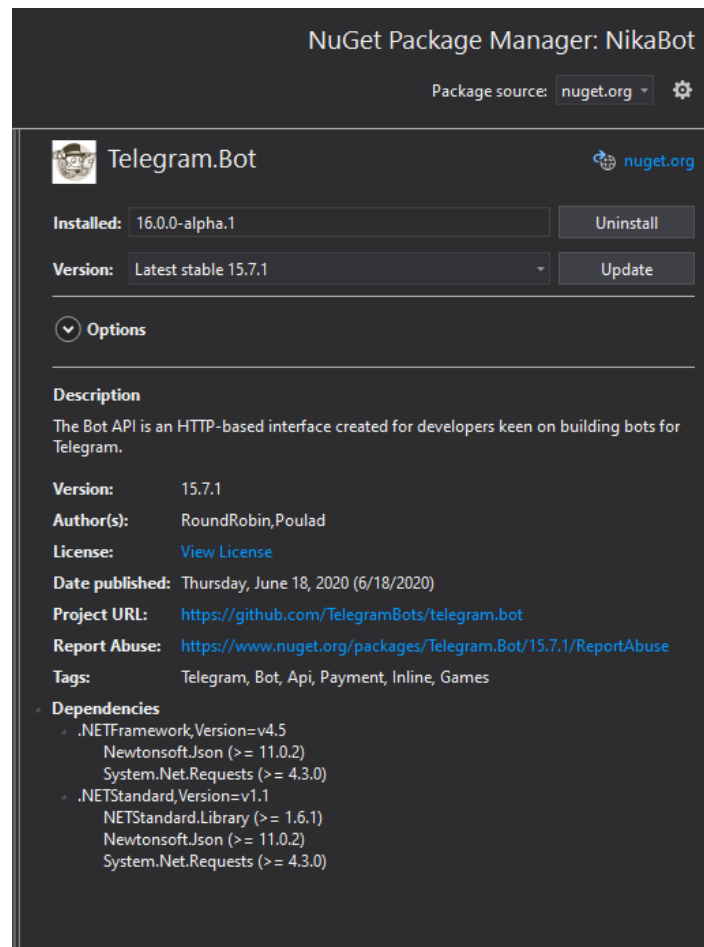


Рис.3.5 NuGet Package

Оскільки розробка ведеться через середовище розробки Visual Studio, стала доступна внутрішня утиліта:

NuGet Package Manager

Використовуючи цю утиліту, можна отримати необмежений доступ, до всіляких бібліотекам, в тому числі і до бібліотеки, яка нас цікавить. (рис10)

Внутрішня архітектура Telegram Бота

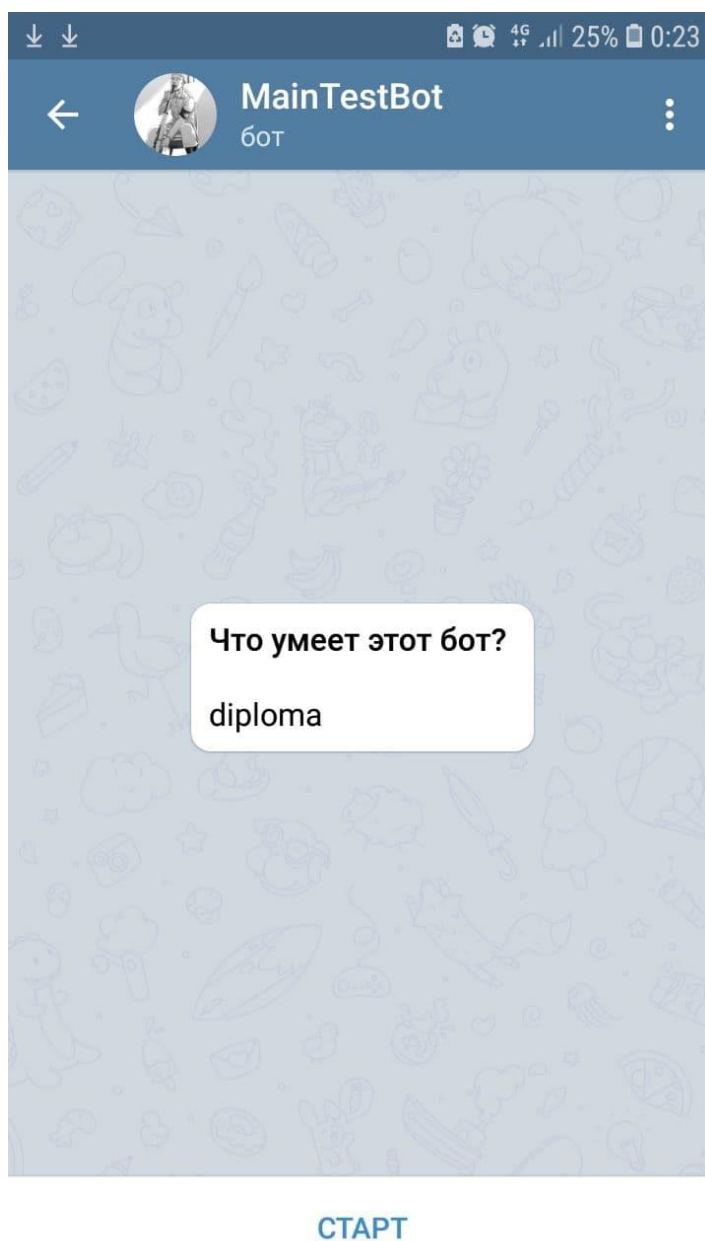


Рис.3.6 Стартовое вікно телеграм бота

На цьому рисунку можна побачити стандартне вікно бота,
Кнопка «Старт»- початок роботи з ботом.

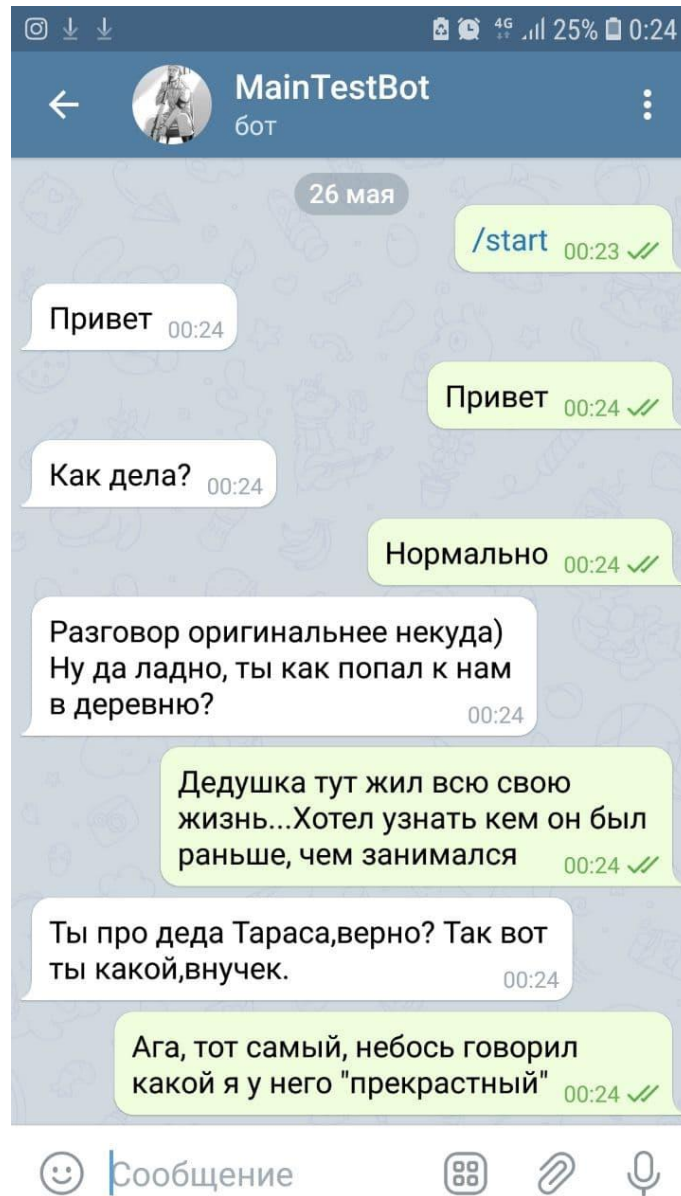


Рис.3.7 Приклад роботи телеграм бота

На цьому рисунку можна побачити основний функціонал бота, а саме спілкування з одним із чотирьох ботів.

Написання бота, та його архітектура

1. Як показано у доданку А, ключ, який ми отримали на момент реєстрації бота, вставляється у окремий клас `Configuration` у поле `string`.

2. Наступний крок, це ініціалізація даних для входу у СУБД, у котрій буде зберігатися уся основна інформація.

База даних знаходиться локально на комп'ютер.

Конфігурація та метод з'єднання представлені у доданку Б.

Клас бази даних та основні методи реалізовані у доданку В.

3. Наступний крок, це встановлення сторонніх бібліотек, для вдалої роботи з Telegram Bot API.

Основні бібліотеки включені у проект за допомогою ключового слова `using`.

Усі бібліотеки перелічені у доданку Г.

4. Додаток Д – вхідна точка програми, та основний метод, як можна побачити,

У цьому доданку можна виділити 2 ключових момента.

- `Bot = new TelegramBotClient(Configuration.BotToken);`
- `public static async Task Main()`

`Bot` - ключовий об'єкт усього проекту.

Member	Type	Value	System Namespace	Count
BotId	int	1796887056	Telegram.Bot.Tel...	1
IsReceiving	bool	false		
MessageOffset	int	0		
Timeout	System.TimeSpan	(00:01:40)		2
Static members				
Non-Public members				
BaseUrl	string	"https://api.telegram.org/file/bot"		
BaseUrl	string	"https://api.telegram.org/bot"		
EmptyUpdates	Telegram.Bot.Types.Update[]		Telegram.Bot.Typ...	
Non-Public members				
ApiResponseReceived	System.EventHandler	null	System.EventHa...	
MakingApiRequest	System.EventHandler	null	System.EventHa...	
OnCallbackQuery	System.EventHandler	null	System.EventHa...	
OnInlineQuery	System.EventHandler	null	System.EventHa...	
OnInlineResultChosen	System.EventHandler	null	System.EventHa...	
OnMessage	System.EventHandler	null	System.EventHa...	
OnMessageEdited	System.EventHandler	null	System.EventHa...	
OnReceiveError	System.EventHandler	null	System.EventHa...	
OnReceiveGeneralError	System.EventHandler	null	System.EventHa...	
OnUpdate	System.EventHandler	null	System.EventHa...	
BaseUrlRequestUrl	string	"https://api.telegram.org/bot1796887056:AAG1A87-..."		3
HttpClient	System.Net.Http.HttpClient		System.Net.Http...	
ReceivingCancellationT...	System.Threading...		System.Threadin...	
Token	string	"1796887056:AAG1A87-03oPk7dFuXBeWDrAv4Q73-..."		4

Рис.3.8 Клас Bot

- 1- Унікальний ID бота,
- 2- Час запиту
- 3- URL посилання на API сервіс
- 4- Token бота

Написання Unity гри

Першим кроком було встановлення середовища розробки, після цього з'являється наступне вікно.

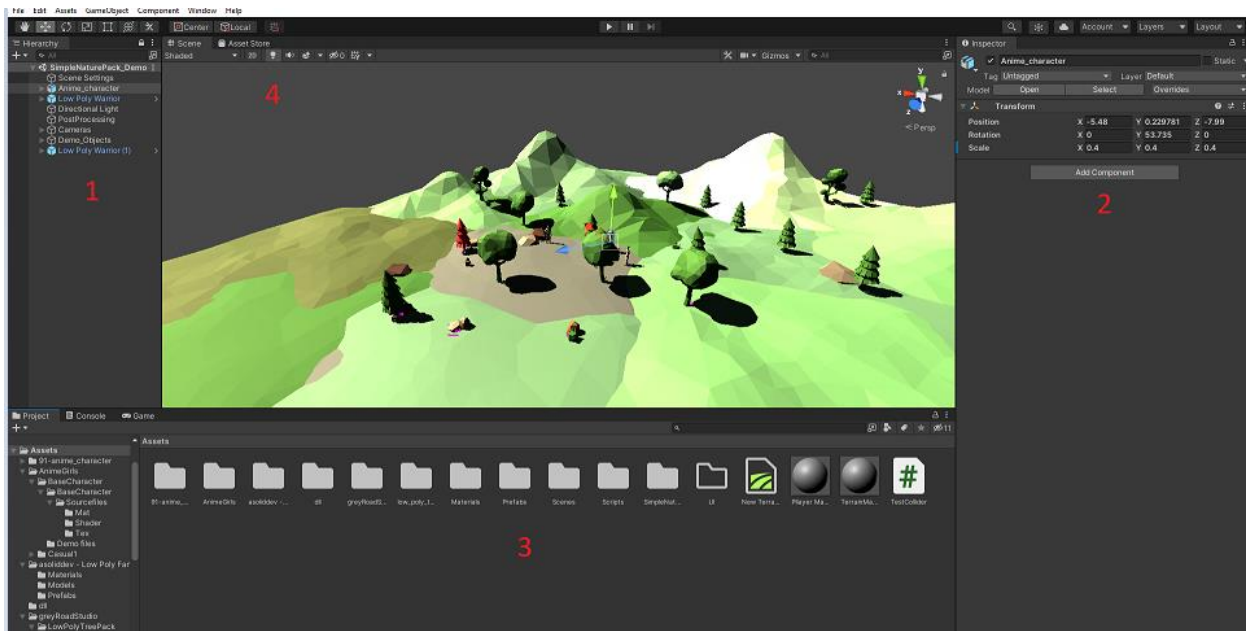


Рис.3.9 Інтерфейс рушія Unity 3D

- 1-Ієрархія усіх об'єктів.
- 2-Внутрішній «Інспектор» Відображає усі властивості вибраного об'єкту.
- 3-Вікно з усіма матеріалами, зазвичай там знаходяться:
 - Скрипти.
 - Моделі.
 - Аудіодоліжки.

4-голівне вікно.

- Через нього робиться розробка самої гри.
- Можливість перейти до онлайн магазину від unity-Assets Store.
- Дивитися на гру, очима камери.

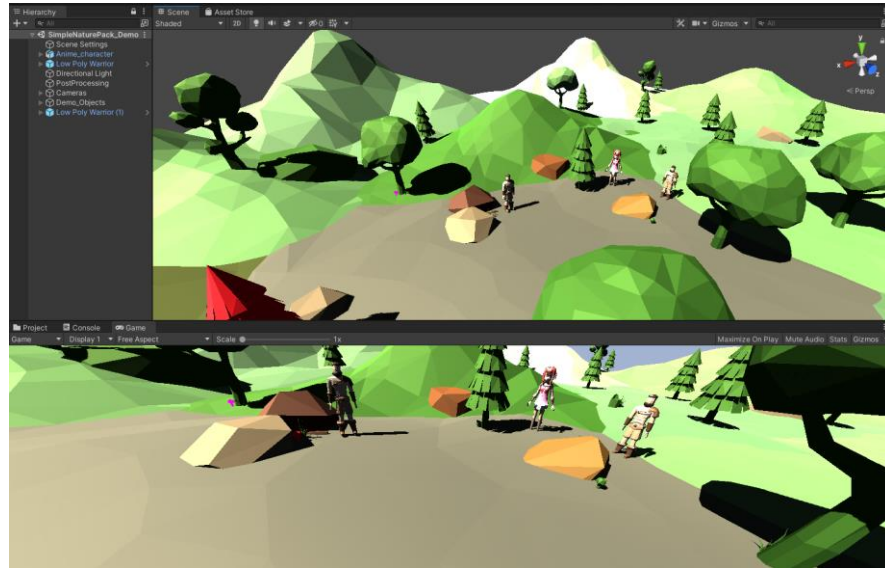


Рис.3.10 Проекція камери на ігрову сцену

Процес гри можна побачити лише очима камери, яка у нашому випадку буде закріплена як можна побачити на рисунку 3.10

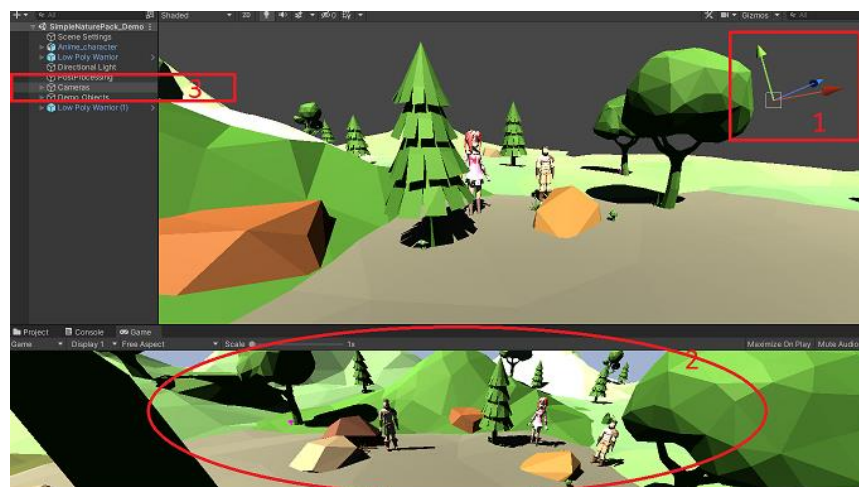


Рис.3.11 Основні елементи інтерфейсу

1. об'єкт камери у грі.
2. Представлення яке бачить сама гра.
3. Камера у ієрархії усіх файлів.

Камера закріплена за головним героєм, та у самого героя повинен бути набір рухів, а саме .

- Рух вправо
- Рух вліво
- Рух назад
- Рух вперед
- Прижок

Увесь цей функціонал вдалося реалізувати завдяки скрипту [Player_Movement](#).

Його код можна побачити у доданку Е.

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.Space))
    {
        IsSpacePressad = true;
    }
    X_Input = Input.GetAxis("Horizontal");
    Z_Input = Input.GetAxis("Vertical");
}
```

Рис.3.12 Метод Update

Викрикається кожен раз у 0,001 секунду.

Та в залежності від того, яку кнопку натиснув гравець .

Робить те, що диктує код програми.

Повний лістинг команд наступний.

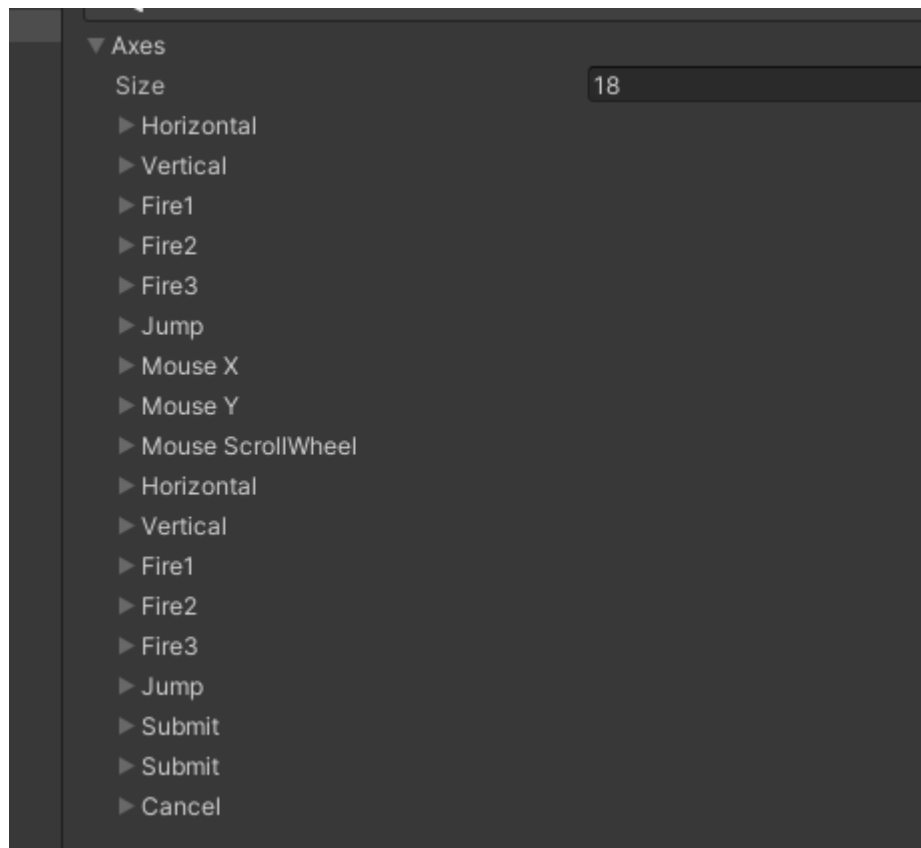


Рис.3.13 Перелік команд гравця

Кожен об'єкт складається із різних компоненті, розглянемо одного із ботів.

Коли гравець підходить до одного з ботів, Цей бот відправляє повідомлення до гравця

Код , завдяки якому це виконується можна побачити у доданку Є.

Схожий за синтаксисом код виконується з 3 остальними ботами, єдине що відрізняється, це, токен бота, який у всіх різний, приклад можна побачити у доданку Ж.

Проектування бази даних

База даних є одним з основних компонентів проекту.

База даних складається з 5 таблиць.

1- Player

Name:

Comment:

Columns:

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	ID	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	Nika_lastMess...	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_0900_ai_ci
3	Matias_lastMe...	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_0900_ai_ci
4	Nika_reputation	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
5	Matias_reputa...	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
6	Player_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

Рис.3.14 Таблиця Player

1- MatiasMessages

Name:

Comment:

Columns:

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	ID	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	Title	VARCHAR	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
3	Answer1	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
4	Answer1_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
5	Answer2	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
6	Answer2_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
7	Answer3	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
8	Answer3_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
9	Is_end	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

Рис.3.15 Таблиця MatiasMessages

2-NikaMessages

Name: nikamessages
Comment:

Columns: + Add - Remove ▲ Up ▼ Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	ID	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...				
2	Title	VARCHAR	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci		
3	Answer1	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci		
4	Answer1_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
5	Answer2	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci		
6	Answer2_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
7	Answer3	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci		
8	Answer3_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
9	Is_end	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Рис.3.16 Таблица MatiasMessages

4-DenisMessages.

Name: matiasmessages
Comment:

Columns: + Add - Remove ▲ Up ▼ Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	ID	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	Title	VARCHAR	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
3	Answer1	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
4	Answer1_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
5	Answer2	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
6	Answer2_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
7	Answer3	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
8	Answer3_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
9	Is_end	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

Рис.3.17 Таблица DenisMessages

5-ElenaMessages.

Name: matiasmessages
Comment:

Columns: + Add - Remove ▲ Up ▼ Down

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	ID	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREME...		
2	Title	VARCHAR	1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
3	Answer1	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
4	Answer1_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
5	Answer2	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
6	Answer2_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
7	Answer3	VARCHAR	1000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	"		utf8mb4_0900_ai_ci
8	Answer3_ID	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
9	Is_end	INT	10	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

Рис.3.18 Таблица ElenaMessages

Взаємозв'язок між усіма таблицями виглядає наступним чином.

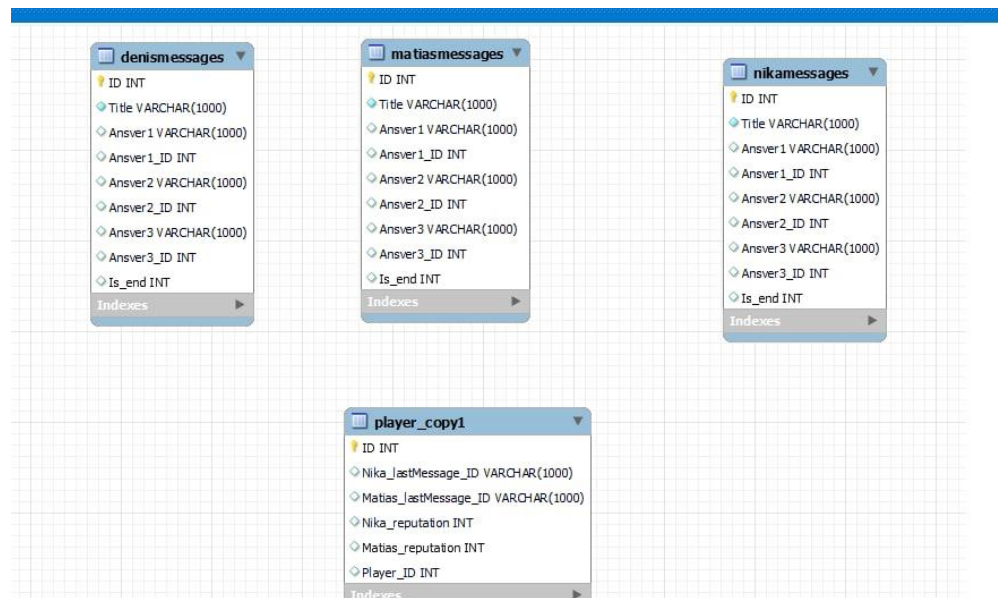


Рис.3.19 Реляційна діаграма баз даних

ID	Title	Answer1	Answer1_ID	Answer2	Answer2_ID	Answer3	Answer3_ID	Is_end
1	Привет	Привет	3	Пока	2		(NULL)	0
2	Прощай	Ладно((NULL)		(NULL)		(NULL)	1
3	Как дела?	Нормально	7	Прохладно	4		(NULL)	0
4	Как Грубо	Прости	(NULL)	Пофиг	(NULL)		(NULL)	0
7	Разговор оригинальнее некуда) Ну да ладно, ты к...	Дедущка тут жил всю свою жизнь...Хотел узнать к...	8	Да так	(NULL)	(NULL)	(NULL)	0
8	Ты про деда Тараса, верно? Так вот ты какой, вунчак.	Ты на что намекаешь?	(NULL)	Ага, тот самый, небось говорил какой я у него "п...	(NULL)		(NULL)	(NULL)

Рис.3.20. Приклад інформації яка зберігається у таблиці NikaMessages.

Title – повідомлення яке напише Telegram Bot.

Answer1 – перший варіант відповіді.

Answer1_ID – посилання на наступний елемент .

Answer2 – другий варіант відповіді.

Answer2_ID – посилання на наступний елемент.

Answer3 – третій варіант відповіді .

Answer3_ID – посилання на наступний елемент.

Is_end – чи є даний елемент фінальним.

У таблиці первинним ключом є ID.

Ось так буде виглядати данні колонки у чаті TelegramBot.

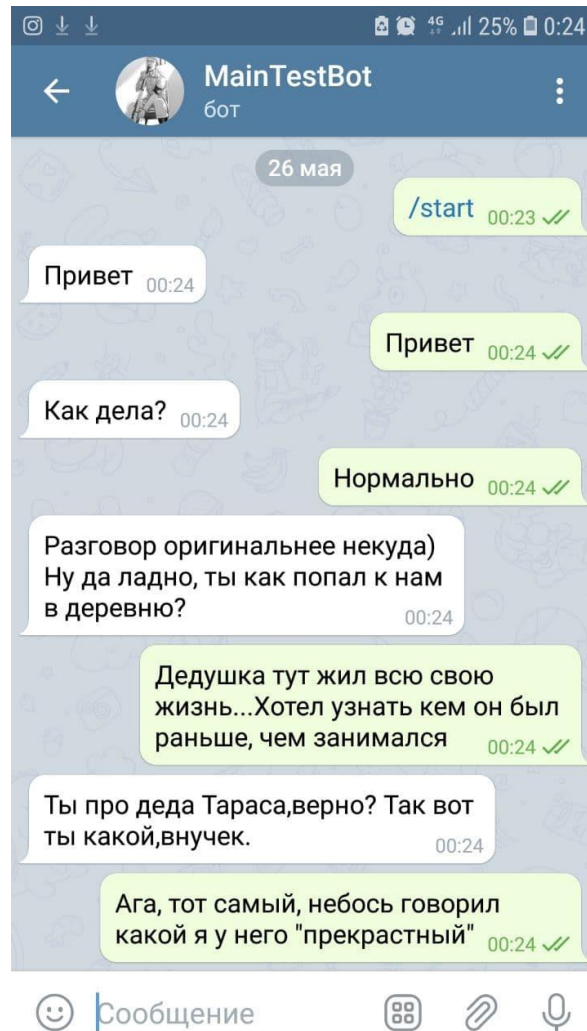


Рис.3.21 Представлення повідомлень у месенджері

Якщо представити ці повідомлення у вигляді діаграми, то ми отримаємо бінарноподібне дерево.

Яке має наступний вигляд.

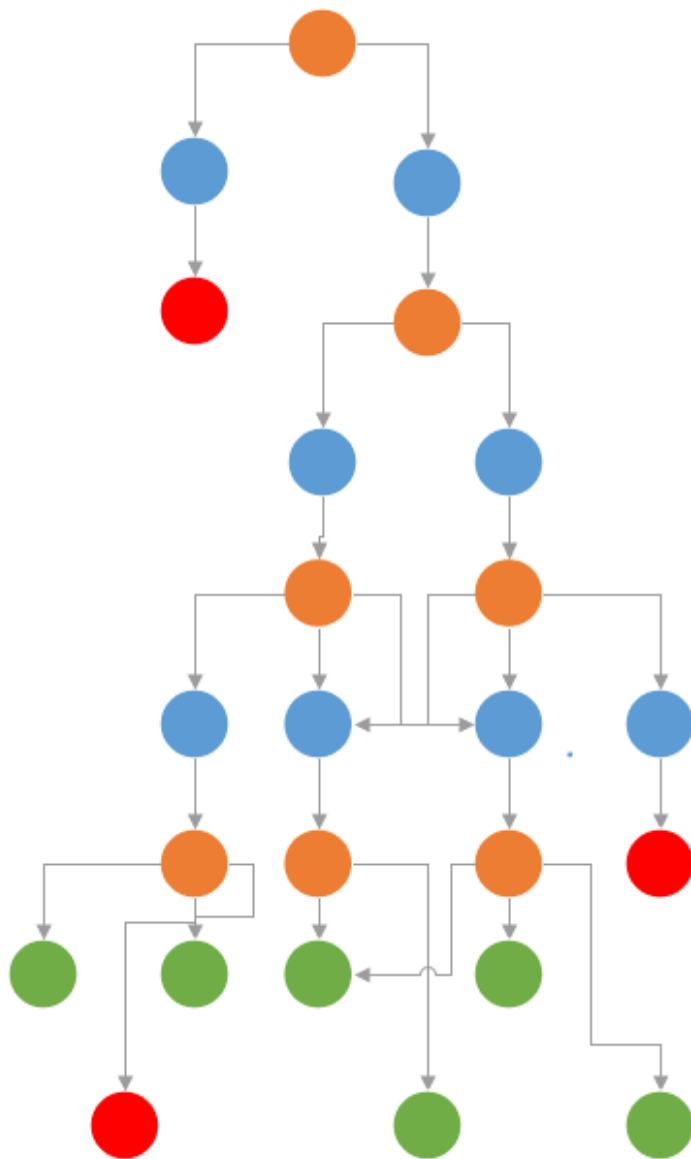


Рис.3.22 Ієрархія дерева повідомлень

1. Помаранчевий - повідомлення бота
2. Блакитний - варіант відповіді гравця
3. Зелений – гарна кінцівка
4. Червоний – погана кінцівка

Взаємозв'язок між усіма компонентами

Unity

Стартовою точкою є вхід до гри, та перед собою можна побачити стартову локацію.

Якщо підійти до одного з чотирьох ботів, він напише повідомлення.



Рис.3.23 Взаємодія з ботом в середній грі

Після цього, гравець може перейти до свого Telegram акаунту, та продовжити своє спілкування там.

Схема діалогів для кожного бота відрізняється, але має схожу структуру, а саме структуру бінарного дерева, як було показано на рисунку 3.23

Після потрапляння до одного з декількох пунктів у діалозі, гравець, підходячи до бота може отримати різні відповіді, наприклад, якщо гравець закінчив діалог на поганому кінці, то можна буде побачити наступну відповідь.

Telegram

Коли гравець перейде до месенджера, то він зможе спілкуватися з ботом у наступному форматі.

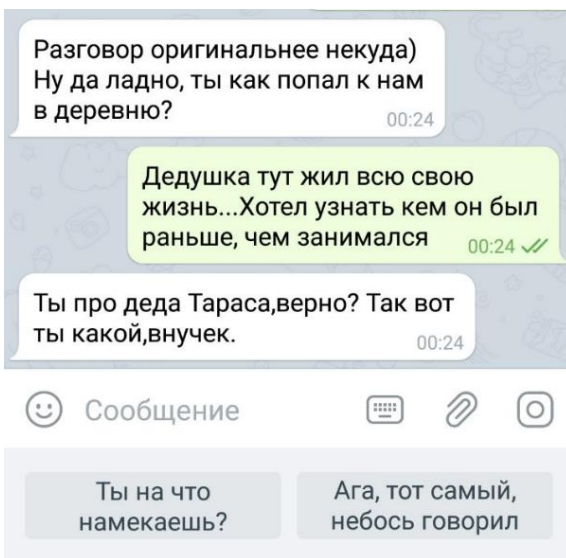


Рис.3.24 Варіативні відповіді гравця

У самому месенджері можна побачити діалог зі всіма чотирьма ботами.

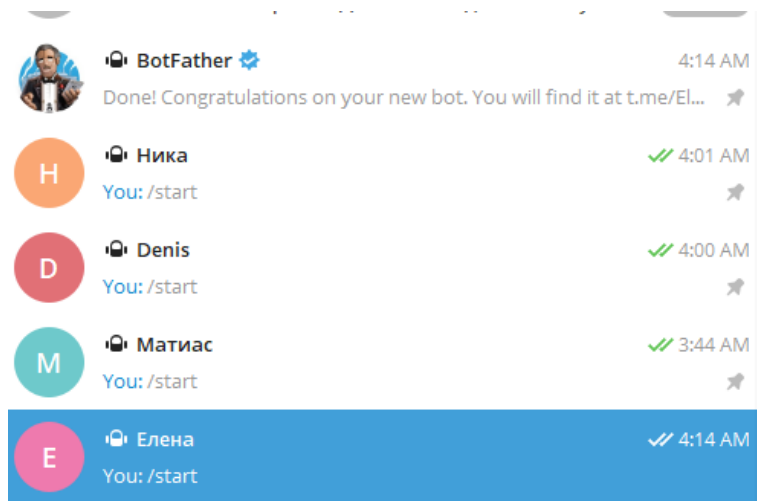


Рис.3.25 Представлення ботів у месенджері Telegram

Якщо представити взаємозв'язок між усіма компонентами, то можна отримати наступну діаграму.

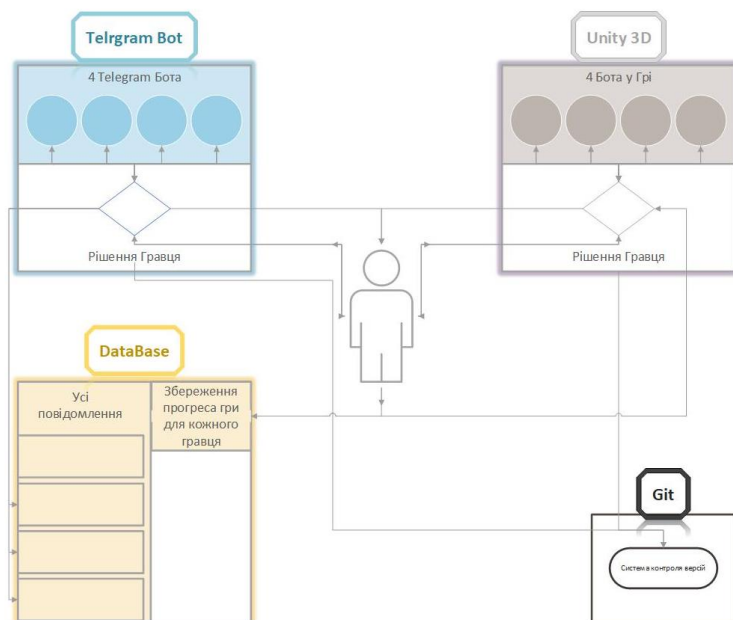


Рис.3.26 діаграма взаємодії основних компонентів

Програмна інженерія

Під час розробки важливим компонентом виступало планування задачі, цю проблему вирішує діаграма життєвого циклу проекту, яку було побудовано у каскадному стилі.

ЖИТТЄВИЙ ЦИКЛ

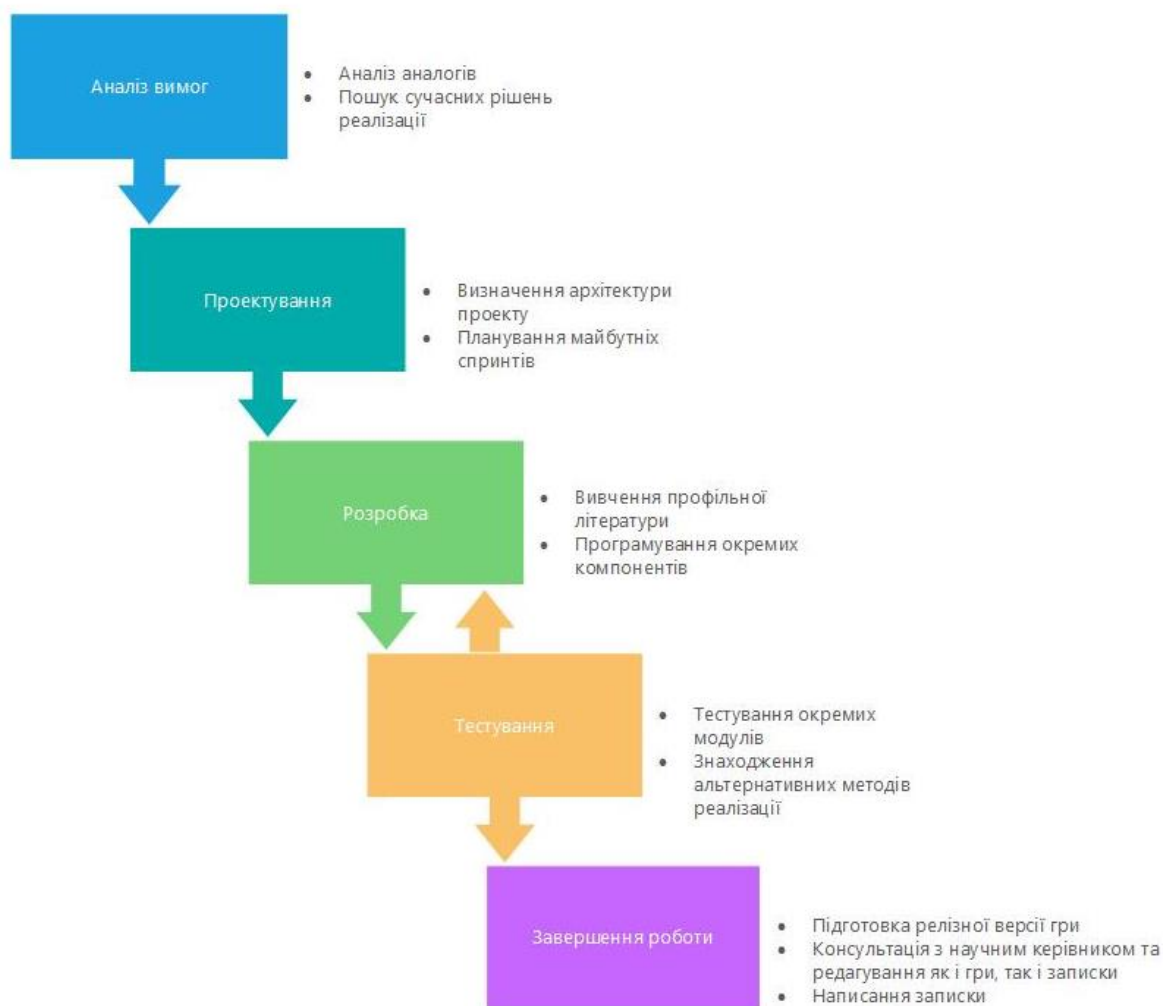


Рис.3.27 життєвий цикл проекту

ВИСНОВКИ

Під час написання бакалаврської роботи було спроектовано та реалізовано проект гри.

Проектування включало в себе сучасні методи програмної інженерії, аналіз ринку, планування життєвого циклу, чітка модерація усього шляху роботи.

Сама гра була розроблена згідно з планами, описаними у постановці завдання.

Знання, отримані під час роботи над бакалаврським проектом, є актуальними та корисними, та допоможуть у побудуванні майбутньої кар'єри програмного інженера.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://ru.ign.com/stardew-valley/81924/review/retsenziia-na-stardew-valley>
2. <https://app-android.ru/blog/na-chyom-razrabatyivat-prilozhenie-unity-3d-ili-nativno>
Unity Technologies – The World's Leading Game Engine". Nanalyze. October 18, 2017.
3. Suckley, Matt (December 10, 2015). "Unity 5.3 launches with improved support for iOS 9 and WebGL". *Pocketgamer.biz*.
4. Axon, Samuel (September 27, 2016). "Unity at 10: For better—or worse—game development has never been easier". *Ars Technica*.
5. Once Celebrated in Russia, the Programmer Pavel Durov Chooses Exile.—
6. *PostgreSQL contributors* (2011). "PostgreSQL server programming". *PostgreSQL 9.1 official documentation*. postgresql.org.
7. <http://archives.postgresql.org/pgsql-hackers/2008-09/msg00071.php>
8. "Khan Academy | Free Online Courses, Lessons & Practice". *Khan Academy*.
9. "CREATE PROCEDURE". *MariaDB KnowledgeBase*.
10. SQL-92, 4.22 SQL-statements, 4.22.1 Classes of SQL-statements "There are at least five ways of classifying SQL-statements:", 4.22.2, SQL statements classified by function "The following are the main classes of SQL-statements:"; SQL:2003 4.11 SQL-statements, and later revisions.
11. "What's new in C# 8.0". *Microsoft Docs*.
12. "What's new in C# 7.3". *Microsoft Docs*.
13. "Visual Studio 2019 Preview Release Notes". docs.microsoft.com.
14. ".NET Core 5". dotnetfoundation.org. .NET Foundation.
15. ["NuGet Gallery - Home"](https://nuget.org). *nuget.org*.
16. ["Mono 1.0 Release Notes"](https://docs.microsoft.com).
17. ["Announcing Xamarin - Miguel de Icaza"](https://tirania.org). *Tirania.org*.
18. ["Framework"](https://docs.microsoft.com). *DocForge*

ДОДАТКИ

Додаток А

```
public static class Configuration
{
    public readonly static string BotToken = "AAG1A87-
O3oPkeWDxAv4Qi73mLUnc";
}
```

Додаток Б

```
class DBUtils
{
    public static MySqlConnection GetDBConnection()
    {
        string host = "localhost";
        int port = 3306;
        string database = "telegram_bots";
        string username = "root";
        string password = "root#@!";

        return DBMySQLUtils.GetDBConnection(host, port, database, username,
password);
    }
}
```

Додаток В

```
class DBMySQLUtils
{
    public static MySqlConnection
        GetDBConnection(string host, int port, string database, string username, string
password)
    {
        String connString = "Server=" + host + ";Database=" + database
            + ";port=" + port + ";User Id=" + username + ";password=" + password;

        MySqlConnection conn = new MySqlConnection(connString);

        return conn;
    }
}
```

Додаток Г

```
using Telegram.Bot;  
using Telegram.Bot.Args;  
using Telegram.Bot.Types;  
using Telegram.Bot.Types.Enums;  
using Telegram.Bot.Types.InlineQueryResults;  
using Telegram.Bot.Types.InputFiles;  
using Telegram.Bot.Types.ReplyMarkups;
```

Додаток Д

```
public static async Task Main()  
{  
    Bot = new TelegramBotClient(Configuration.BotToken);  
  
    var me = await Bot.GetMeAsync();  
  
    Bot.OnMessage += BotOnMessageReceived;  
  
    Bot.StartReceiving(Array.Empty<UpdateType>());  
  
    Console.ReadLine();  
  
    Bot.StopReceiving();  
}
```

Додаток Е

```
public class Player_Movement : MonoBehaviour
{
    private bool IsSpacePressad;
    private float X_Input;
    private float Z_Input;
    private Rigidbody ridgetbody;
    public Transform groundCheckTransform;
    void Start()
    {
        ridgetbody = GetComponent<Rigidbody>();
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            IsSpacePressad = true;
        }
        X_Input = Input.GetAxis("Horizontal");
        Z_Input = Input.GetAxis("Vertical");
    }

    private void FixedUpdate()
    {
        if (Physics.OverlapSphere(groundCheckTransform.position, 0.4f).Length == 1)
```

```

    {
        return;
    }

    if (IsSpacePressad)
    {
        ridgetbody.AddForce(Vector3.up *10 , ForceMode.VelocityChange);
        IsSpacePressad = false;
    }

    ridgetbody.velocity = new Vector3(X_Input*15, ridgetbody.velocity.y,
ridgetbody.velocity.z);
    ridgetbody.velocity = new Vector3(ridgetbody.velocity.x, ridgetbody.velocity.y,
Z_Input * 15);

    }

}

```

Додаток Є

```

public static async Task Main()
{
    Bot = new TelegramBotClient(Configuration.BotToken);
    var me = await Bot.GetMeAsync();
    Console.Title = me.Username;
    Bot.OnMessage += BotOnMessageReceived;
}

```

```
Bot.StartReceiving(Array.Empty<UpdateType>());
Console.WriteLine($"Start listening for @ {me.Username}");

Console.ReadLine();
Bot.StopReceiving();
}

private static async void BotOnMessageReceived(object sender,
MessageEventArgs messageEventArgs)
{
    var message = messageEventArgs.Message;
    if (message == null || message.Type != MessageType.Text)
        return;

    switch (message.Text.Split(' ').First())
    {
        case "/start":
            await Start(message);
            break;
        case "/keyboard":
            await SendReplyKeyboard(message);
            break;

        default:
            await Dialog(message);
            break;
    }
}
```

```

static async Task Dialog(Message message)
{
    string title = "", answer1 = "", answer2 = "", answer3 = "";
    string mess = message.Text;
    int lastMessage_ID = 1, answer_id = 0;
    await Bot.SendChatActionAsync(message.Chat.Id, ChatAction.Typing);
    string connStr =
"server=localhost;user=root;database=telegram_bots;port=3306;password=root";
    MySqlConnection conn = new MySqlConnection(connStr);
    try
    {
        Console.WriteLine("Connecting to MySQL...");
        conn.Open();

        string sql = "SELECT * from player where Player_ID=" + message.Chat.Id
+ "";

        MySqlCommand cmd = new MySqlCommand(sql, conn);
        MySqlDataReader rdr = cmd.ExecuteReader();

        while (rdr.Read())
        {
            lastMessage_ID = Int32.Parse(rdr[2].ToString());
        }
        rdr.Close();
    }
}

```

```

sql = "SELECT * from Matiasmessages where ID=" + lastMessage_ID +
"";

cmd = new MySqlCommand(sql, conn);
MySqlDataReader rdr2 = cmd.ExecuteReader();

try
{
    while (rdr2.Read())
    {
        if (rdr2[2].ToString().Equals(mess)) { answer_id =
Int32.Parse(rdr2[3].ToString()); break; }
        else if (rdr2[4].ToString().Equals(mess)) { answer_id =
Int32.Parse(rdr2[5].ToString()); break; }
        else if (rdr2[6].ToString().Equals(mess)) { answer_id =
Int32.Parse(rdr2[7].ToString()); break; }
    }
}
catch
{
    Console.WriteLine("Pizda");
}
rdr2.Close();

sql = "update player SET Matias_lastMessage_ID=" + answer_id + "
WHERE Player_ID =" + message.Chat.Id + "";
cmd = new MySqlCommand(sql, conn);

```

```
cmd.ExecuteNonQuery();

sql = "SELECT * from matiasmessages where ID=" + answer_id + "";
cmd = new MySqlCommand(sql, conn);
MySqlDataReader rdr3 = cmd.ExecuteReader();
try
{
    while (rdr3.Read())
    {

        title = rdr3[1].ToString();
        answer1 = rdr3[2].ToString();
        answer2 = rdr3[4].ToString();
        answer3 = rdr3[6].ToString();
    }

}
catch
{
    Console.WriteLine("Pizda");
}
rdr3.Close();

}
catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
}
```

```

    }
    await Task.Delay(2000);

    var replyKeyboardMarkup = new ReplyKeyboardMarkup(
        new KeyboardButton[][]
        {
            new KeyboardButton[] { answer1, answer2, answer3 },
        },
        resizeKeyboard: true
    );
    await Bot.SendTextMessageAsync(
        chatId: message.Chat.Id,
        text: title,
        replyMarkup: replyKeyboardMarkup
    );
}

static async Task Start(Message message)
{
    string title = "", answer1 = "", answer2 = "", answer3 = "";
    int Is_end = 0;

    await Bot.SendChatActionAsync(message.Chat.Id, ChatAction.Typing);
    string connStr =
"server=localhost;user=root;database=telegram_bots;port=3306;password=root";
    MySqlConnection conn = new MySqlConnection(connStr);
    try

```

```
{
    Console.WriteLine("Connecting to MySQL...");
    conn.Open();

    string sql = "SELECT * from matiasmessages where ID = 1;INSERT INTO
player (Matias_lastMessage_ID, Matias_reputation,Player_ID) VALUES (1 , 0, " +
message.Chat.Id + ")";
    MySqlCommand cmd = new MySqlCommand(sql, conn);
    MySqlDataReader rdr = cmd.ExecuteReader();

    while (rdr.Read())
    {
        title = rdr[1].ToString();
        ansver1 = rdr[2].ToString();
        ansver2 = rdr[4].ToString();
        Is_end = Int32.Parse(rdr[8].ToString());
    }
    rdr.Close();
}
catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
}
await Task.Delay(2000);

var replyKeyboardMarkup = new ReplyKeyboardMarkup(
    new KeyboardButton[][]
```

```
{  
  
    new KeyboardButton[] { answer1, answer2 },  
    },  
    resizeKeyboard: true  
);  
await Bot.SendTextMessageAsync(  
    chatId: message.Chat.Id,  
    text: title,  
    replyMarkup: replyKeyboardMarkup  
);  
}  
  
static async Task SendReplyKeyboard(Message message)  
{  
    var replyKeyboardMarkup = new ReplyKeyboardMarkup(  
        new KeyboardButton[][]  
        {  
            new KeyboardButton[] { "1.1", "1.2" },  
            new KeyboardButton[] { "2.1", "2.2" },  
        },  
        resizeKeyboard: true  
    );  
  
    await Bot.SendTextMessageAsync(  
        chatId: message.Chat.Id,
```

```
text: "Choose",  
replyMarkup: replyKeyboardMarkup
```

```
);  
}
```

```
}
```

Додаток Ж

```
class Configuration  
{  
    public readonly static string BotToken =  
    "1879574120:AAHUKIeuwrX5H2IWANTrh0K8a2EtlfMVjrI";  
}
```