

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра Інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА FRONTEND-ЧАСТИНИ БАГАТОКОРИСТУВАЦЬКОЇ
КОМП'ЮТЕРОНІ ГРИ GTA V З ВИКОРИСТАННЯМ РУШІЯ AitV ТА
ФРЕЙМВОРКУ REACT»**

Виконав: студент 4 курсу, групи ПД–43
спеціальності

121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

Торжевський Я.Р.
(прізвище та ініціали)

Керівник Щербина І.С.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Напрямок підготовки - 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

О.В. Негоденко

“ ___ ” _____ 2023 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ
ТОРЖЕВСЬКОМУ ЯНУ РУСЛАНОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка frontend-частини багатокористувацької комп'ютерної гри GTA V з використанням рушія AltV та фреймворку React»

Керівник роботи Щербина І.С.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від “ 24 ” лютого 2023 року №26

2. Строк подання студентом роботи “ 1 ” червня 2023 року

3. Вхідні дані до роботи:

3.1. Науково-технічна література, пов'язана з розробкою багатокористувацьких рольових ігор.

3.2. Технічна документація платформи AltV.

3.3. Технічна документація платформи Figma.

3.4. Технічна документація фреймворку React.js.

4. Зміст розрахунково-пояснювальної записки (перелік питань які потрібно розробити).

4.1. Аналіз індустрії багатокористувацьких рольових ігор

4.2. Аналіз рушіїв для розробки комп'ютерних рольових ігор

4.3. Аналіз мов програмування, та фреймворків для розробки frontend-частини

4.4. Розробка frontend-системи багатокористувацької гри

4.4.1. Розробити систему реєстрації та аутентифікації

4.4.2. Розробити систему створення, та управління персонажами

4.4.3. Розробити систему чату

- 4.4.4. Розробити систему діалогів
- 4.4.5. Розробити систему телефону
- 4.4.6. Розробити систему меню взаємодії
- 4.4.7. Розробити систему екрану втрати свідомості
- 4.4.8. Розробити систему магазинів
- 4.4.9. Розробити систему управління нерухомістю
- 4.4.10. Розробити систему заправної станції
- 4.4.11. Розробити систему автошколи
- 4.4.12. Розробити міні-гру «Сортування пігулок»
- 4.4.13. Розробити систему головного інтерфейсу
- 4.4.14. Розробити систему меню документів
- 4.4.15. Розробити систему конструювання одягу
- 4.4.16. Розробити систему головного меню гравця.
- 4.4.17. Розробити систему інвентарю
- 4.5. Тестування розробленої системи

5. Перелік демонстраційного матеріалу

- 5.1. Мета, об'єкт та предмет дослідження
- 5.2. Аналоги
- 5.3. Технічне завдання
- 5.4. Стек технологій
- 5.5. Висновки

6. Дата видачі завдання “25” лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.02.23-27.02.23	Виконано
2	Аналіз та дослідження існуючих аналогів	28.02.23-10.03.23	Виконано
3	Моделювання, проектування системи	13.03.23-24.03.23	Виконано
4	Розробка основного функціоналу системи	27.03.23-28.04.23	Виконано
5	Тестування системи	01.05.23-05.05.23	Виконано
6	Розробка інфраструктури та розгортання системи	08.05.23-12.05.23	Виконано
7	Вступ, реферат, висновки	15.05.23-19.05.23	Виконано
8	Розробка основних демонстраційних матеріалів	22.05.23-26.05.23	Виконано
9	Попередній захист роботи		Виконано
10	Здача роботи	01.06.23	Виконано

Студент _____

(підпис)

(прізвище та ініціали)

Керівник роботи _____

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 83 с., 3 табл., 71 рис., 2 дод., 10 джерел.

JAVASCRIPT, FIGMA, ALTV, GTAV, MONGODB.

Об'єкт дослідження - геймплей у багатокористувацькій комп'ютерній рольовій грі.

Предмет дослідження - frontend-частина багатокористувацької комп'ютерної гри на базі GTA V з використанням рушія AltV.

Мета роботи - покращення взаємодії користувача з сервером багатокористувацької рольової гри засобами frontend компонентів на платформі AltV з використанням фреймворку React.js.

Методи дослідження - методи та технології розробки frontend-систем багатокористувацьких рольових ігор.

В ході виконання дипломної роботи була розроблена frontend-система багатокористувацької рольової гри. Був застосований повний цикл розробки програмного забезпечення який включає:

- планування архітектури та підбір технології;
- розробка системи;
- тестування системи;

Обрано оптимальні засоби розробки та підходи до проектування. Система має гнучку модульну архітектуру з використанням провідних шаблонів проектування. Систему було налагоджено, протестовано та випробувано, що показало, коректність роботи програмного забезпечення. Система є ефективною та вирішує всі поставлені задачі.

У процесі розробки було реалізовано всі необхідні системи проекту:

- система реєстрації та аутентифікації
- система створення, та управління персонажами
- система чату
- система діалогів
- система телефону

- система меню взаємодії
- система екрану втрати свідомості
- система магазинів
- система управління нерухомістю
- система заправної станції
- система автошколи
- міні-гра «Сортування пігулок»
- система головного інтерфейсу
- система меню документів
- система конструювання одягу
- система головного меню гравця.
- система інвентарю

ЗМІСТ

ВСТУП.....	11
1. ПОСТАНОВКА ЗАДАЧІ ТА ВИЗНАЧЕННЯ ТЕХНОЛОГІЧНОГО СТЕКУ ДЛЯ РОЗРОБКИ FRONTEND-ЧАСТИНИ БАГАТОКОРИСТУВАЦЬКОЇ РОЛЬОВОЇ ГРИ НА ОСНОВІ GTA V НА БАЗІ ПЛАТФОРМИ ALTV З В И К О Р И С Т А Н Н Я М REACT.JS.....	14
1.1 Дослідження індустрії комп'ютерних ігор у сучасному світі.....	14
1.2 Дослідження та вибір платформи для розробки багатокористувацьких серверів рольових комп'ютерних ігор.....	16
1.3 Вибір технологій для розробки frontend-частини багатокористувацької комп'ютерної рольової гри.....	18
1.4 Вибір технологій для розробки дизайну.....	20
1.5 Постановка задачі.....	22
2. ПРОЕКТУВАННЯ FRONTEND-ЧАСТИНИ БАГАТОКОРИСТУВАЦЬКОЇ РОЛЬОВОЇ ГРИ НА ОСНОВІ GTA V З ВИКОРИСТАННЯМ ПЛАТФОРМИ ALTV ТА REACT.JS.....	24
2.1 Розробка функціональних вимог для багатокористувацьких комп'ютерних рольових ігор.....	24
2.2 Розробка нефункціональних вимог для багатокористувацьких комп'ютерних рольових ігор.....	38
2.3 Можливості рушія AltV.....	39
2.4 Розробка алгоритму взаємодії користувача багатокористувацької рольової гри.....	41
3. РОЗРОБКА ІНТЕРФЕЙСІВ ТА АЛГОРИТМІВ РОБОТИ.....	43
3.1 Створення UI\UX дизайну з використанням платформи Figma.....	43
3.2. Розробка інтерфейсу реєстрації та аутентифікації.....	45
3.3. Розробка геймплейних систем та систем меню.....	47
4. РОЗРОБКА СИСТЕМ ВЗАЄМОДІЇ ТА АЛГОРИТМІВ РОБОТИ.....	69
4.1 Розробка інтерфейсів взаємодії.....	69

	10
ВИСНОВКИ.....	80
ПЕРЕЛІК ПОСИЛАНЬ.....	81
ДОДАТОК А.....	84
ДОДАТОК Б.....	91

ВСТУП

Сучасний світ неможливо уявити без впливу ігрової індустрії. Вона не лише перетворила наше уявлення про розваги, а й стала важливим сегментом глобальної економіки.

Комп'ютерні ігри, що з кожним днем стають все більш реалістичними та захоплюючими, пропонують гравцям множинність жанрів, сюжетів та дій. Одним з найпопулярніших жанрів відеоігор є ігри у відкритому світі, які надають гравцям можливість вільно досліджувати величезний ігровий простір та взаємодіяти з різними об'єктами та персонажами.

Ігри у відкритому світі, такі як Grand Theft Auto (GTA), The Elder Scrolls, The Witcher та Red Dead Redemption, користуються попитом через свою гнучкість, реалістичність та безмежні можливості для взаємодії з ігровим середовищем. Гравці можуть вільно вибирати, яким шляхом йти, які завдання виконувати та як розвивати свого персонажа. Безумовно однією з найбільш відомих ігор цього жанру є Grand Theft Auto V (GTA V), розроблена компанією Rockstar North і випущена Rockstar Games в 2013 році. GTA V прославилася завдяки своєму гігантському, деталізованому і реалістичному відкритому світу, який відображає різноманітність сучасного американського життя. Гра надає гравцям безліч можливостей для дій, включаючи місії, перегони, різні види взаємодії з оточуючим світом. Однак найбільшу популярність гра набула після виходу багатокористувацького режиму, в якому гравці мали змогу взаємодіяти, спілкуватися та разом виконувати різноманітні завдання.

Проте, із розвитком технологій, галузь комп'ютерних ігор взяла курс на більшу персоналізацію ігрового досвіду. Гравці більше не хочуть бути просто спостерігачами, вони хочуть створювати, модифікувати та впливати на світ ігор, в яких вони беруть участь. Це призвело до появи нових можливостей і технологій для розробки ігор, з окремою увагою на розробку багатокористувацьких комп'ютерних рольових ігор [1]. Багатокористувацькі комп'ютерні рольові ігри є одним із найбільш перспективних напрямків в ігровій індустрії. Вони забезпечують гравцям можливість іммерсивної взаємодії з віртуальним світом,

виконуючи різноманітні ролі та завдання. Відкритий світ GTA V став ідеальним полем для реалізації цієї ідеї, дозволивши розробникам створити багатокористувацькі комп'ютерні рольові ігри на основі вже існуючого різноманітного та реалістичного світу.

Під час розробки рольових режимів на базі GTA V важливо зосередитися на створенні глибокої і динамічної геймплейної механіки, здатної задовольнити різні інтереси гравців. Сюжет, персонажі, завдання та інтерактивні елементи повинні бути ретельно продуманими та добре виконаними, щоб забезпечити неперевершений ігровий досвід.

Метою дипломної роботи є покращення взаємодії користувача з сервером багатокористувацької рольової гри на платформі AltV з використанням фреймворку React.js, для задоволення потреб гравців, адміністраторів сервера та розробників.

Об'єктом дослідження є геймплей, frontend-компоненти та структура інтерфейсу користувача у багатокористувацькій комп'ютерній рольовій грі.

Предметом дослідження є технології розробки frontend-у багатокористувацької комп'ютерної гри на базі GTA V з використанням рушія AltV.

Актуальність роботи, а саме розробки якісного frontend-у для рольового серверу на базі GTA V з використанням рушія AltV полягає у наступних аспектах:

- зростання популярності рольових серверів: RP сервери приваблюють все більше гравців, які прагнуть зануритися в альтернативний світ гри та відчувати себе в ролі різних персонажів. Якісний frontend допомагає створити комфортне середовище для гравців, сприяючи їхньому задоволенню та залученню до сервера;
- високі вимоги до користувацького досвіду: сучасні гравці ставлять високі вимоги до зручності та ергономіки ігрового інтерфейсу. Розробка якісного frontend-у дозволяє задовольнити потреби користувачів та покращити їхній ігровий досвід;
- багатофункціональність: RP сервери містять велику кількість різноманітних елементів геймплею, таких як економіка, місії, транспорт тощо. Якісний frontend дозволяє структурувати та візуалізувати ці елементи, спрощуючи взаємодію гравців з ігровим середовищем та підвищуючи загальний рівень занурення у гру;
- спрощення роботи розробників: якісний frontend дозволяє розробникам легше

працювати над оновленнями та додаванням нового контенту, адже він забезпечує зручний і модульний підхід до розширення функціональності;

- забезпечення безпеки: на RP серверах GTA V часто обмінюються важливими даними, такими як персональна інформація гравців, інформація про аккаунти та ін. Розробка якісного frontend-у включає захист від можливих загроз, таких як несанкціонований доступ або зловживання даними;

- адаптація до змінних вимог: комп'ютерні рольові ігри постійно розвиваються, і сервери GTA V не є винятком. Якісний frontend повинен мати адаптивну архітектуру, яка дозволяє швидко та ефективно вносити зміни в ігрові механіки, додавати новий контент та відповідати на зміни в ігровій індустрії;

Враховуючи вищезазначені аспекти, ця дипломна робота є актуальною та важливою у контексті сучасної ігрової індустрії. Вона покликана внести вклад у розвиток ігор з відкритим світом, використовуючи одну з найпопулярніших ігор цього жанру - GTA V - як основу для створення нового і захоплюючого ігрового досвіду. Завдяки ній, ми матимемо змогу краще зрозуміти, як розробляти ігри, які можуть задовольнити потреби сучасних гравців.

1. ПОСТАНОВКА ЗАДАЧІ ТА ВИЗНАЧЕННЯ ТЕХНОЛОГІЧНОГО СТЕКУ ДЛЯ РОЗРОБКИ FRONTEND-ЧАСТИНИ БАГАТОКОРИСТУВАЦЬКОЇ РОЛЬОВОЇ ГРИ НА ОСНОВІ GTA V НА БАЗІ ПЛАТФОРМИ ALTV З ВИКОРИСТАННЯМ REACT.JS

1.1 Дослідження індустрії комп'ютерних ігор у сучасному світі

Рольові режими в іграх походять від настільних RPG (roleplay game), таких як Dungeons & Dragons, які були винайдені у 1970-х роках. Вони були засновані на ідеї гравців, що втілюються в ролях персонажів у фентезійному світі, де вони можуть взаємодіяти з іншими персонажами та середовищем. Комп'ютерні рольові ігри почали з'являтися в кінці 1970-х і на початку 1980-х років, переносючи ці концепції в цифровий формат. Однією з ключових характеристик комп'ютерних рольових ігор є можливість вибору і наслідків. Гравці мають можливість вибирати дії своїх персонажів, а ці вибори впливають на хід гри. Ця концепція була запозичена багатьма іншими жанрами ігор, включаючи пригодницькі ігри, стратегічні ігри та навіть бойові ігри. З розвитком ігрової індустрії елементи рольових режимів все більше і більше почали застосовуватись у різноманітних іграх.

Серед ігор які одними з перших отримали рольові режими на серверах можна виокремити гру GTA: San Andreas. Це гра розроблена та випущена студією Rockstar Games у 2004 році. Серед основних особливостей гри був великий відкритий світ, та безліч механік які надихнули гравців на початок розробки онлайн модифікації. Як ми можемо побачити серія ігор GTA нерозривно пов'язана з жанром комп'ютерних рольових ігор, та рольових серверів в цілому. З розвитком технологій та розвитком серії, логічним продовженням напрацювань студії стала GTA V, це гра розроблена з використанням рольових механік для створення більш глибокого користувацького досвіду. Система розвитку персонажа в GTA V також відображає елементи RPG, дозволяючи гравцям покращувати навички своїх персонажів в процесі гри. Більше того, GTA V включає багатокористувацький режим, який дозволяє гравцям взаємодіяти з іншими гравцями в великому

відкритому світі. Але гравці не дивлячись на шалену популярність вбудованого багатокористувацького режиму та наявність рольових елементів у грі хотіли більше персоналізації, та механік які у звичайній грі були не доступні. Саме це було причиною для розвитку нових двигунів та створення рольових серверів на базі GTAV.

Одним з перших великих серверів був NoPixel[10]. Його вихід ще більше підвищив рівень популярності гри, про що свідчить статистика StreamsCharts [3] стрімінгової платформи Twitch представлена на рисунку 1.1.

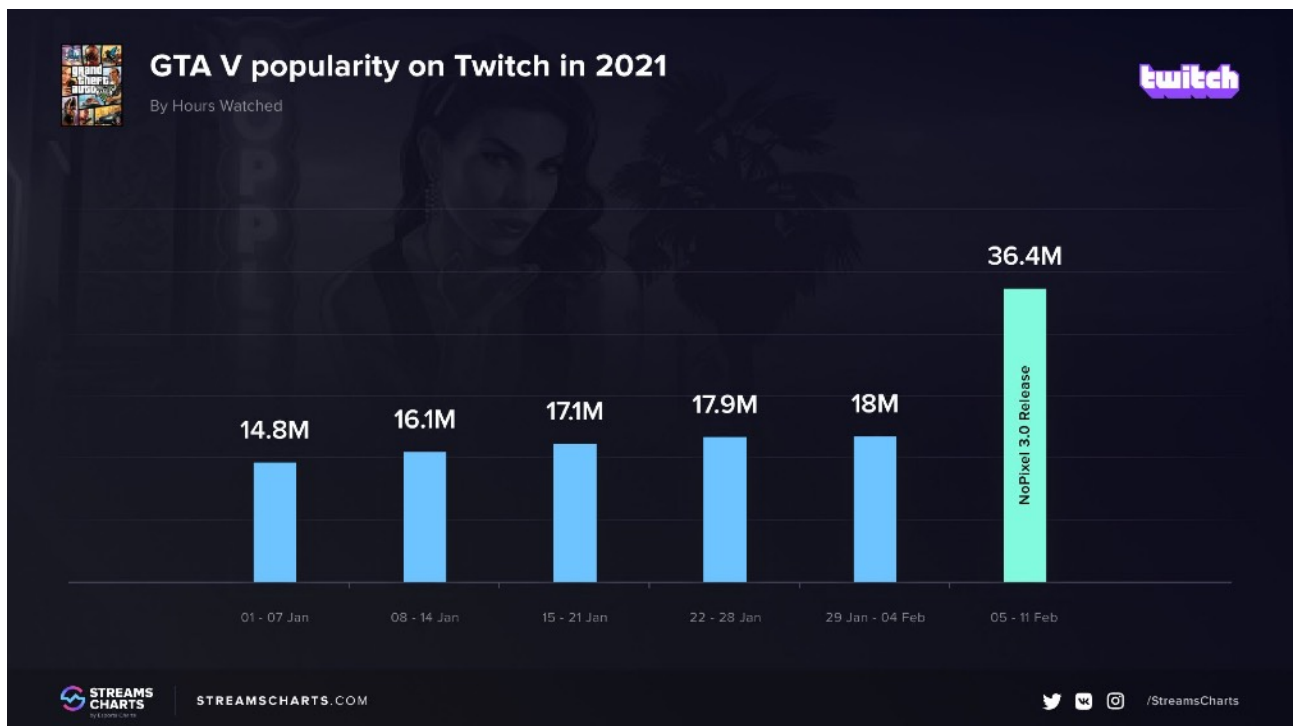


Рис 1.1 Кількість глядачів в категорії GTA V помісячно

Таким чином, історія комп'ютерних рольових ігор - це історія інновацій та еволюції. Від їхніх коренів у настільних RPG до сучасних комп'ютерних ігор, вони продовжують впливати на ігрову індустрію, пропонуючи нові та цікаві способи взаємодії та занурення в ігрові світи. І з появою нових технологій та підходів до геймдизайну, майбутнє рольових режимів виглядає яскравим та обіцяє бути ще більш захоплюючим.

1.2 Дослідження та вибір платформи для розробки багатокористувацьких серверів рольових комп'ютерних ігор

У світі рольових серверів на базі GTA V існує кілька платформ для розробки та розгортання серверів, найпопулярнішими з яких є AltV [4], FiveM [5] та RageMP [6]. Ці платформи пропонують розробникам різні можливості та переваги. У цьому розділі буде розглянуто кожен з цих платформ і обґрунтовано, чому для даної роботи було обрано саме AltV.

FiveM - одна з найвідоміших платформ для створення серверів GTA V. Вона пропонує широкий спектр можливостей, таких як підтримка різних мов програмування, велика кількість готових модів та ресурсів, а також активна спільнота розробників. Однак FiveM має певні обмеження з точки зору продуктивності та оптимізації, особливо коли мова йде про великі сервери з великою кількістю гравців.

RageMP - це альтернативна платформа для розробки серверів GTA V, яка зосереджена на високій продуктивності та стабільності. RageMP пропонує широкі можливості розробки, такі як підтримка мов програмування C# та JavaScript, а також відносно невелику кількість відгуків від спільноти. Однак RageMP має менш активну спільноту, що може ускладнити пошук допомоги та ресурсів для розробки.

AltV - відносно нова платформа для розробки серверів GTA V, яка набуває популярності завдяки своїй високій продуктивності, стабільності та гнучкості. AltV підтримує різні мови програмування, такі як JavaScript, C# та Lua (за допомогою модулів розширення).

Після ґрунтовного аналізу кожної платформи було створено Таблиця 1.1, на якій зображені усі критерії та особливості основних платформ для розробки багатокористувацьких комп'ютерних рольових ігор на основі GTA V за якими було обрано платформу для виконання даної роботи.

Критерії	AltV	Rage MP	FiveM
Макс. кількість гравців	До 4096	До 1000	До 128
Підтримка мов програмування	JS, C#, Lua, Python	JS, C#, C++	Lua, JS
Стабільність	Висока	Середня	Середня
Персоналізація	Висока	Висока	Обмежена
Підтримка модів	Дуже гнучка	Гнучка	Обмежена
Перфоманс	Високий	Середній	Середній
Серверна оптимізація	Дуже висока	Середня	Низька
Підтримка стрімінгу асетів	Так	Обмежено	Ні
Кроссплатформенність	Так	Ні	Ні

Таблиця 1.1 Порівняння платформ для розробки багатокористувацьких рольових серверів

Мною було обрано платформу AltV.

Нижче наведено деякі з переваг AltV, які обґрунтували його використання у дипломній роботі:

- висока продуктивність: AltV відома своєю високою продуктивністю та оптимізацією, що дозволяє їй підтримувати великі сервери з великою кількістю гравців без значного зниження часу відгуку та стабільності;
- гнучкість та підтримка різних мов програмування: AltV підтримує різні мови програмування, такі як JavaScript, C# та Lua (за допомогою модулів розширення). Це дозволяє розробникам вільно обирати мову програмування, яка найкраще підходить для їхнього проекту та є зручною у використанні;
- активна спільнота: AltV має активну спільноту розробників, яка пропонує допомогу, підтримку та ресурси як для початківців, так і для досвідчених програмістів. Це може бути корисно для розробки RP-серверів, де виникають специфічні проблеми, що вимагають спеціалізованих знань;

- написання скриптів на сервері та клієнті: AltV дозволяє використовувати скрипти як на сервері, так і на клієнті, що дає розробникам широкі можливості для модифікації гри та створення взаємодії між гравцями та ігровим середовищем;
- відкритий вихідний код: AltV - це проект з відкритим вихідним кодом, що означає, що розробники можуть вільно долучатися до платформи, вдосконалювати її або навіть створювати власні модифікації;
- якісний мережевий код: AltV має неперевершену систему синхронізації внутрішньоігрових подій, що дозволяє уникнути десинхронізації між гравцями;

1.3 Вибір технологій для розробки frontend-частини багатокористувацької комп'ютерної рольової гри

Вибір технологій для розробки ґрунтувався на вимогах рушія. AltV рекомендує використовувати React.js, або Vue.js. Однак важливо зазначити, що Vue.js може некоректно працювати на певних конфігураціях комп'ютерів.

Розглянемо обидва фреймворка:

React.js [7] - це популярна та потужна JavaScript бібліотека, спеціально розроблена

для створення користувацьких інтерфейсів. React.js підтримує модульний підхід до побудови користувацьких інтерфейсів, що дозволяє легко створювати багаторазові компоненти, які можна комбінувати для створення складних та інтерактивних елементів інтерфейсу. Цей підхід добре узгоджується з динамічною природою інтерфейсів RP-серверів, де різні елементи інтерфейсу повинні безперешкодно працювати разом.

Важливою складовою є те що React.js використовує віртуальну DOM (Document Object Model) для оптимізації продуктивності рендерингу. Ця функція гарантує, що оновлюються лише необхідні частини інтерфейсу, мінімізуючи вплив на продуктивність та забезпечуючи плавний та чуйний ігровий процес на RP-серверах. React.js також має велику екосистему бібліотек та інструментів, які можуть допомогти пришвидшити процес розробки та спростити такі завдання, як

управління станом, валідація форм, тощо. Інтеграція з іншими frontend-технологіями дуже легка, такими як наприклад CSS та HTML, а також з бекенд-системами за допомогою API.

Неменш важливо те що React.js широко використовується розробниками по всьому світу, що означає наявність великої спільноти та великої кількості ресурсів. Ця підтримка може бути неоціненною при вирішенні проблем.

Vue.js – це відкритий JavaScript-фреймворк, що використовується для створення динамічних користувацьких інтерфейсів. Vue.js дозволяє створювати шаблони за допомогою HTML-подібного синтаксису, який автоматично оновлюється при зміні даних. Це означає, що ви можете створювати динамічні інтерфейси з меншим зусиллям і кодом. Завдяки компонентній системі, ви можете будувати застосунки, використовуючи повторно використовувані компоненти, що спрощує створення великих масштабних проектів.

Однією з ключових переваг Vue.js є його швидкість та продуктивність. Він використовує легкий віртуальний DOM і оптимізовані відповіді на події, що робить його одним з найшвидших фреймворків. Крім того, Vue.js відрізняється простотою вивчення завдяки своєму простому і зрозумілому API.

Гнучкість Vue.js полягає у тому, що він може бути використаний для розробки як простих веб-сторінок, так і складних односторінкових застосунків (SPA). Він також легко інтегрується з іншими проектами та бібліотеками, такими як Vuex для управління станами і Vue Router для роутингу.

Vue.js є популярним вибором серед розробників завдяки його простоті, гнучкості та продуктивності, що робить його добрим вибором для розробки веб-застосунків середнього розміру.

Результати дослідження обох фреймворків відображені у Таблиця 1.2 критеріїв та особливостей, завдяки якій було обрано фреймворк для розробки frontend-системи комп'ютерної рольової гри.

Характеристика	Vue.js	React.js
Модель програмування	Орієнтована на шаблони	Базована на компонентах
Швидкість вивчення	Легше для новачків, бо має більш простий синтаксис	Може бути складнішим для новачків через JSX та концепції, як Flux або Redux. Однак, дуже гнучкий та потужний для досвідчених розробників
Розмір	Менший розмір бібліотеки	Великий розмір бібліотеки, але це не є проблемою з правильною конфігурацією
Швидкодія	Швидкий, але може бути повільнішим на великих масштабах	Швидкий, ідеально підходить для великих додатків завдяки віртуальному DOM
Популярність	Має зростаючу спільноту	Дуже популярний, велика спільнота, багато ресурсів для вивчення
Використання	Підходить для невеликих та середніх проектів	Ідеально підходить для великих та складних проектів
Інтеграція з AltV	Вимагає додаткової конфігурації	Ідеально підходить для AltV завдяки підтримці JSX та віртуального DOM, що дозволяє ефективно управляти ігровим інтерфейсом

Таблиця 1.2 Порівняння фреймворків для розробки frontend-систем комп'ютерних рольових ігор

Враховуючи ці переваги, React.js був обраний в якості основної технології розробки frontend-системи для дипломної роботи. React.js пропонує надійне та ефективне рішення для створення інтерактивних та захоплюючих інтерфейсів для RP-серверів на платформі AltV, забезпечуючи приємний ігровий досвід для гравців.

1.4 Вибір технологій для розробки дизайну

Вибір правильного інструменту для розробки дизайну є важливим етапом у процесі створення багатокористувацької комп'ютерної гри рольової гри. Від нього залежить ефективність та якість кінцевого продукту [2]. На ринку сьогодні

представлені численні інструменти для дизайну, включаючи Figma, Adobe XD, Sketch та інші.

Figma[8] - це веб-базований інструмент для дизайну інтерфейсів, який пропонує реальний спільний доступ до роботи, широкі можливості для створення прототипів, а також інтеграцію з іншими інструментами, такими як Slack та Zeplin.

Adobe XD - це професійний інструмент для дизайну інтерфейсів від Adobe, який пропонує широкий набір функцій для дизайну, анімації та прототипування, а також глибоку інтеграцію з іншими продуктами Adobe.

Sketch - це популярний інструмент для дизайну інтерфейсів для Mac, який пропонує широкий набір функцій для дизайну та прототипування, але обмежений тільки платформою Mac.

В результаті дослідження особливостей кожної з наведених платформ, була створена порівняльна Таблиця 1.3, яка відображає основні відмінності кожного інструменту.

Критерії	Figma	Adobe XD	Sketch
Платформа	Веб, Windows, Mac	Windows, Mac	Mac
Співпраця в реальному часі	Так	Ні	Ні
Прототипування	Так	Так	Так
Інтеграції	Slack, Zeplin, і інші	Adobe Suite	Обмежено
Ціна	Безкоштовний план, платні плани	Безкоштовний план, платні плани	Платний

Таблиця 1.3 Порівняння програм та платформ для розробки дизайну інтерфейсу комп'ютерної рольової гри

Враховуючи ці фактори, Figma видається найкращим вибором. Це найбільш універсальний інструмент з представлених, що працює на різних платформах і пропонує можливості співпраці в реальному часі, що може бути надзвичайно корисним для ефективної роботи. Крім того, Figma має широкий набір інтеграцій

з іншими популярними інструментами, що робить його ще більш гнучким у використанні.

1.5 Постановка задачі

Сутність гри полягає у тому, що гравець отримує можливість відчувати себе у ролі працівника певної компанії, державної служби або створити свою особисту компанію та розвивати її, шляхом підбору персоналу, будівництву. Як показали дослідження для розробки frontend-системи рольового серверу найбільш доцільно використовувати платформу AltV та фреймворк React.js. Для досягнення мети необхідно виконати такі ключові завдання:

- Аналіз сучасної ігрової індустрії: Визначення основних трендів у розробці відеоігор.
- Аналіз вимог до гри: Визначити специфікацію та функціональні вимоги, що включають розробку концепції гри, дизайну, графічних вимог, інтерфейсу користувача та інших ключових елементів.
- Вибір інструментів для розробки: Вибрати оптимальні інструменти для розробки, які відповідають вимогам проекту. Це включає вибір рушія (AltV), мови програмування (JavaScript), інструменту для дизайну (Figma) та інших необхідних інструментів.
- Визначення функціональних та нефункціональних вимог: Розробити детальний список вимог до гри, включаючи функціональні та нефункціональні аспекти.
- Розробка frontend-системи: Реалізувати гру згідно з визначеними вимогами, включаючи розробку геймплея, дизайну, інтерфейсу користувача, графіки та інших елементів.
- Тестування роботи frontend-компонентів: Провести ретельне тестування роботи інтерфейсу користувача для виявлення та виправлення помилок.
- Інтеграція з backend-системою гри: Інтеграція frontend-системи з розробленою backend-системою.

Виконання цих завдань дозволить створити багатокористувацьку

комп'ютерну рольову гру, яка задовольнить потреби користувачів і буде ефективно працювати на двигуні AltV.

2. ПРОЕКТУВАННЯ FRONTEND-ЧАСТИНИ БАГАТОКОРИСТУВАЦЬКОЇ РОЛЬОВОЇ ГРИ НА ОСНОВІ GTA V З ВИКОРИСТАННЯМ ПЛАТФОРМИ ALTV TA REACT.JS

2.1 Розробка функціональних вимог для багатокористувацьких комп'ютерних рольових ігор

Функціональні вимоги визначають конкретні функції або можливості, які система або продукт повинні мати. Це специфічні вимоги, які визначають, що повинна робити система, включаючи обробку даних, функції користувача, процеси, послуги та поведінку системи в різних ситуаціях. Функціональні вимоги описують дії, які користувач може виконувати в системі, як система повинна відповідати на конкретні вхідні дані, як вона повинна обробляти деякі процеси і т.д.

Створення функціональних вимог є важливим етапом процесу розробки програмного забезпечення. Вони використовуються для визначення функціональності, яку система повинна надавати, і стають основою для проектування, розробки, тестування і впровадження системи. Вони допомагають уникнути непорозумінь та неясностей у вимогах до системи, підвищують ефективність процесу розробки і забезпечують, що кінцевий продукт відповідає очікуванням користувачів.

При розробці frontend-частини багатокористувацької комп'ютерної рольової гри на базі GTA V, основна увага приділяється розробці користувацького інтерфейсу.

Для створення функціональних вимог скористаємось UML-діаграмою прецедентів. Для початку ознайомимось що з себе представляє UML-діаграма прецедентів.

UML, або Мова моделювання об'єктів, є стандартною мовою, призначеною для специфікації, візуалізації, побудови та документації артефактів системи програмного забезпечення. Вона надає комплексний набір символів та діаграм для опису структурних та поведінкових аспектів системи. UML була розроблена

ініціативою OMG (Об'єднання для стандартизації моделей об'єктів), і є універсальним мовним інструментом для інженерії програмного забезпечення.

Діаграма прецедентів є одним з типів діаграм UML, яка є важливим інструментом для візуалізації та аналізу взаємодії між різними акторами і системою в контексті виконання конкретних функцій або цілей. Вона використовується, перш за все, для визначення та розуміння вимог до системи. Часто діаграми прецедентів використовуються на початкових етапах розробки системи, коли важливо з'ясувати, які функції система повинна виконувати і як вона буде взаємодіяти з кінцевими користувачами або іншими системами.

Діаграма прецедентів включає в себе такі основні елементи:

Прецеденти - це типові сценарії взаємодії між акторами та системою. Вони представляють конкретні функції або дії, які система виконує.

Актори - представляють різні ролі, які взаємодіють з системою. Вони можуть бути людьми, іншими системами або процесами.

Відносини - це зв'язки між акторами та прецедентами. Вони мають форму ліній, які з'єднують акторів з прецедентами, що відображає участь акторів у прецедентах.

Системний контур - це межа, яка показує межі системи. Всі прецеденти знаходяться всередині цього контуру.

Залежності - у них можуть бути прецеденти, що включають або розширюють інші прецеденти.

Розробка діаграми прецедентів починається з визначення акторів і прецедентів. Актори визначаються на основі ролей, які взаємодіють з системою, а прецеденти визначаються на основі функціональності, яку система повинна надавати акторам. Після визначення акторів і прецедентів, вони з'єднуються за допомогою відносин. Це відображає, які актори взаємодіють з якими прецедентами. Наступним кроком є визначення системного контуру, який відображає межі системи. Всі прецеденти розміщуються всередині цього контуру. На останньому етапі визначаються залежності між прецедентами. Це відображає, як прецеденти взаємодіють один з одним, а також включають або розширюють інші прецеденти.

Таким чином, діаграма прецедентів дає ясне уявлення про те, як система взаємодіє з акторами, і які функції вона виконує. Вона є важливим інструментом для розробки та аналізу системи.

Розробимо UML-діаграму прецедентів для frontend-системи багатокористувацької рольової гри на базі GTA V з використанням платформи AltV та фреймворку React.js.

Узагальнена схема функціональних вимог представлена у вигляді UML-діаграми прецедентів на рисунку 2.1.

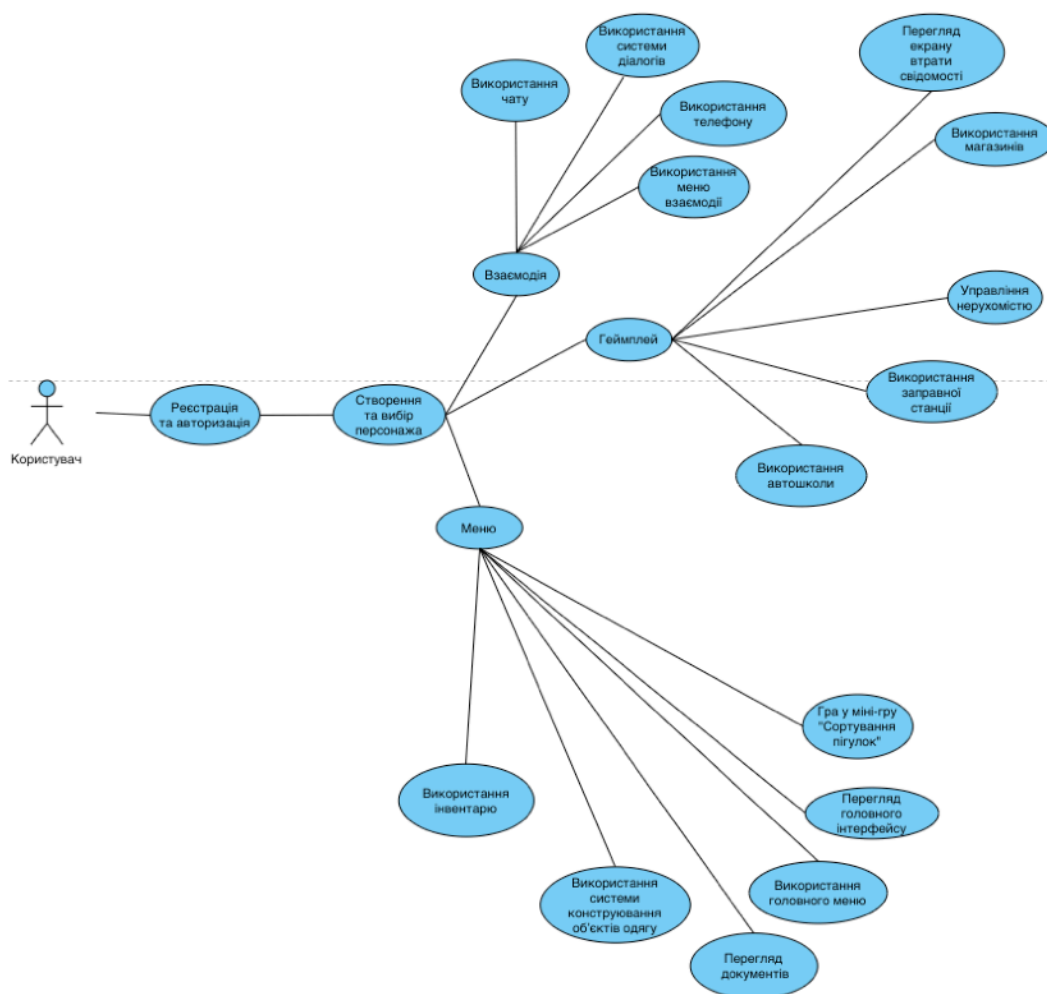


Рис 2.1 Узагальнена схема функціональних вимог

На даній схемі представлені узагальнені можливості користувача гри. Спочатку він використовує меню авторизації та реєстрації, після їх успішного проходження він потрапляє в меню вибору та створення персонажа. Після вибору

персонажу, користувач потрапляє у простір гри. У грі він може використати можливості 3 основних груп: системи взаємодії, системи меню, та геймплейні можливості.

У системах взаємодії користувач може використовувати системи для спілкування, та взаємодії з іншими учасниками ігрового процесу. Серед систем які доступні гравцю є:

Система чату - дозволяє отримувати та надсилати текстові повідомлення до інших гравців.

Система діалогів - надає користувачу, можливість отримувати ігрову інформацію шляхом спілкування з NPC (non-playable-character).

Система телефону - дозволяє використовувати телефон для комунікації, зберігання контактів, надсилання повідомлень, та використання опціональних додатків.

Система взаємодії - надає можливість для взаємодії користувача з іншими гравцями, транспортом, нерухомістю та предметами.

Серед систем геймплею є такі:

Система екрану втрати свідомості - це відображення стану персонажу користувача, система надає можливість викликати медичну допомогу, отримати пропозицію про реанімацію та з'явитись у госпіталі після закінчення таймеру.

Система магазинів - надає користувачу можливість купівлі предметів першої необхідності, таких як: їжа, напої, медикаменти та інструменти, в універсальному магазині. Магазин амуніції надає можливість для придбання набоїв та зброї. Магазин одягу дозволяє гравцю кастомізувати свого персонажа, або покращити його характеристики, такі як: показник броні, та кількість доступних комірок у інвентарі. Магазин транспорту дозволяє обрати та придбати необхідний транспорт.

Система управління нерухомістю - дана система дозволяє користувачу покращувати його житло шляхом придбання покращень, оплати комунальних платежів, а також виставляти нерухомість на продаж, або здачу в оренду.

Система заправної станції - використовується гравцем для заправлення транспорту паливом. Користувач може обрати як тип палива, так і кількість для покупки.

Меню документів - надає можливість для перегляду документів персонажа, таких як паспорт, медична карта, ліцензії на зброю та прав на транспорт.

Меню конструювання одягу - це унікальна система для конструювання наборів одягу, та перетворення їх у ігровий об'єкт. Меню дозволяє створити новий об'єкт, а також переглянути створені об'єкти за категоріями та оновити значення їх характеристик.

Меню інвентарю - меню яке дозволяє гравцю здійснювати управління предметами які в нього є.

Серед ігрових систем, які відносяться до категорії відображення даних через меню є:

Меню міні-гри «Сортування пігулок» - заохочувальна гра для співробітників медицини, дозволяє отримати премію за кожне вірне сортування.

Меню головного інтерфейсу - дана система є основною для коректної гри користувача, вона включає в себе відображення показників, поточних задач, системних повідомлень, інвентарю, спідометру, відображення часу, дати та місцезнаходження персонажа.

Розглянемо кожен з прецедентів більш детально.

Авторизація та реєстрація: Користувач має мати змогу зареєструватися, відновити забутий пароль, або увійти у особистий профіль. Згідно до даних вимог сформовано схему прецедентів системи, що зображена на рисунку 2.2.

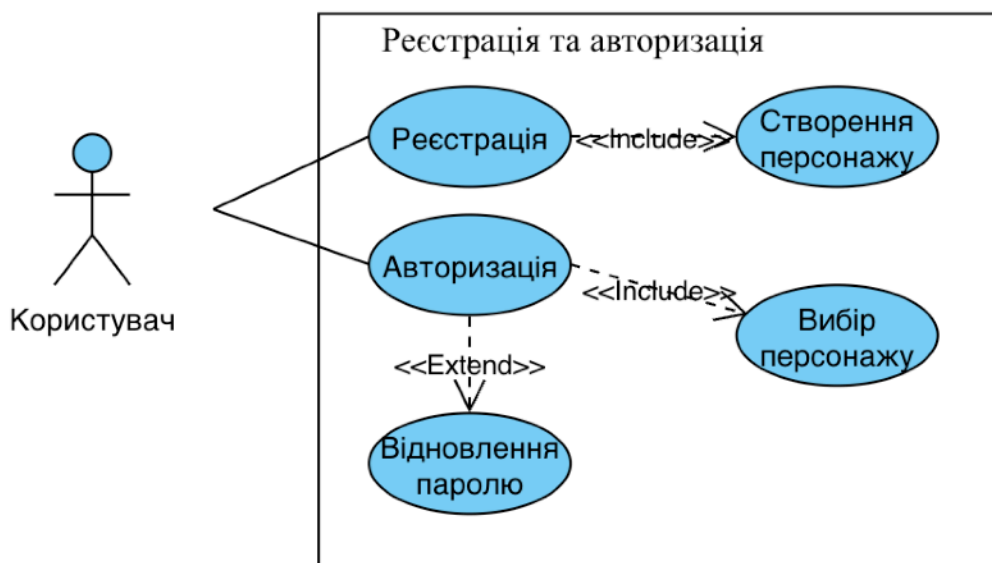


Рис 2.2 Діаграма прецедентів прецеденту «Реєстрація та авторизація»

Створення та вибір персонажу: Гравець може створити персонажа, налаштувати його зовнішність, початковий одяг, ім'я та вік. Схему прецедентів зображено на рисунку 2.3.



Рис 2.3 Діаграма прецедентів прецеденту «Створення та вибір персонажу»

Використання меню документів: Користувач може показувати, або переглядати власні документи, документи, які обов'язково містять номер документи, ім'я, вік, дату видачі, серед опціональних: фото громадянина, перелік доступних для водіння категорій, діагнози та стан здоров'я. Схему прецедентів зображено на рисунку 2.4

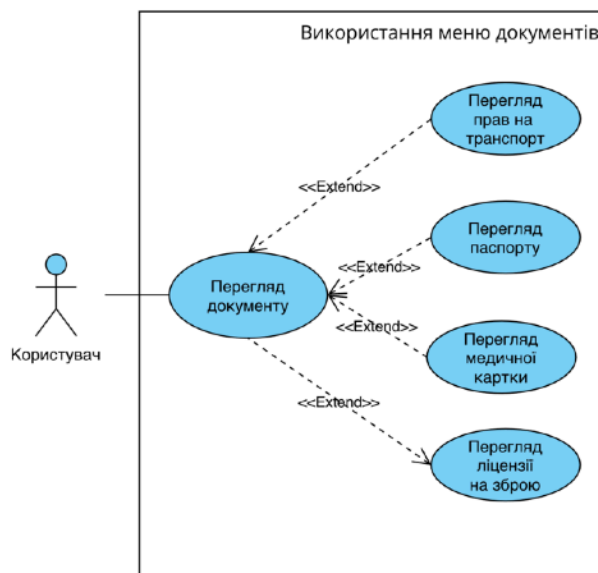


Рис 2.4 Діаграма прецедентів прецеденту «Перегляд документів»

Використання магазинів: Користувач має мати змогу придбати одяг, їжу, інструменти, тощо відведених для цього магазинах. Схему прецедентів зображено на рисунку 2.5.

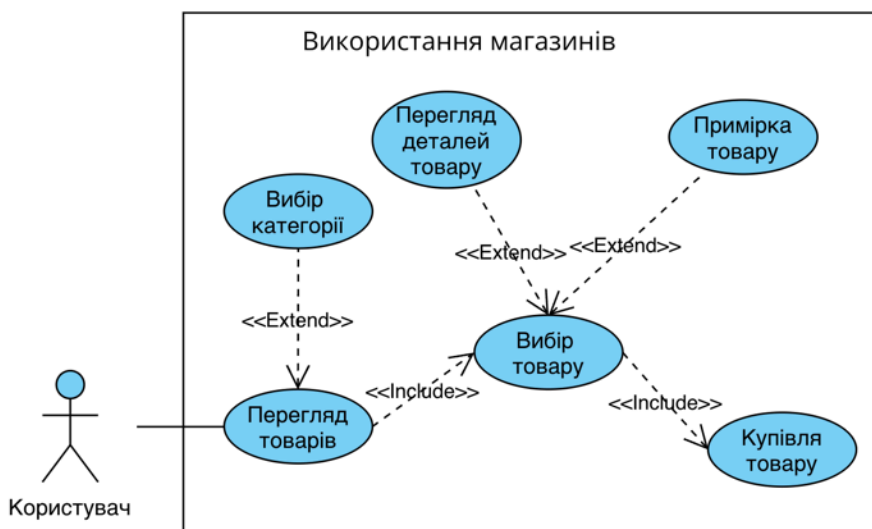


Рис 2.5 Діаграма прецедентів прецеденту «Використання магазину»

Управління нерухомістю: Користувач може сплачувати комунальні платежі, створювати запити на продаж чи здачу в оренду нерухомості, купувати покращення для домівки, відкривати чи закривати нерухомість. Схему прецедентів зображено на рисунку 2.6.

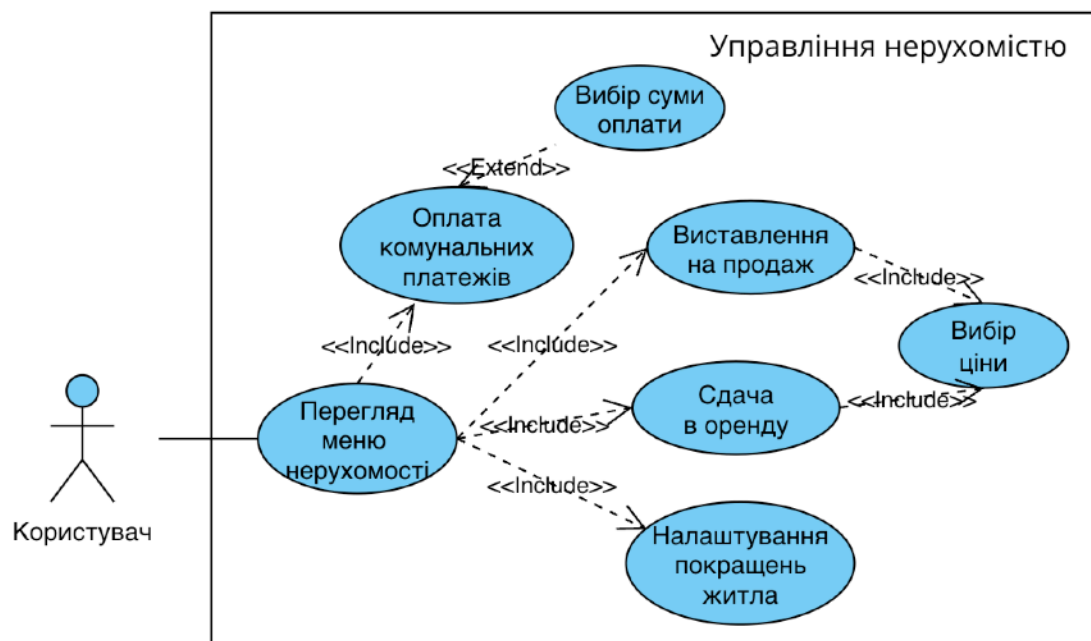


Рис 2.6 Діаграма прецедентів прецеденту «Управління нерухомістю»

Використання системи діалогів: Користувач може спілкуватись з не ігровими персонажами. У результаті такого спілкування користувач може отримати інформацію, або виконати певну дію. Схему прецедентів зображено на рисунку 2.18.

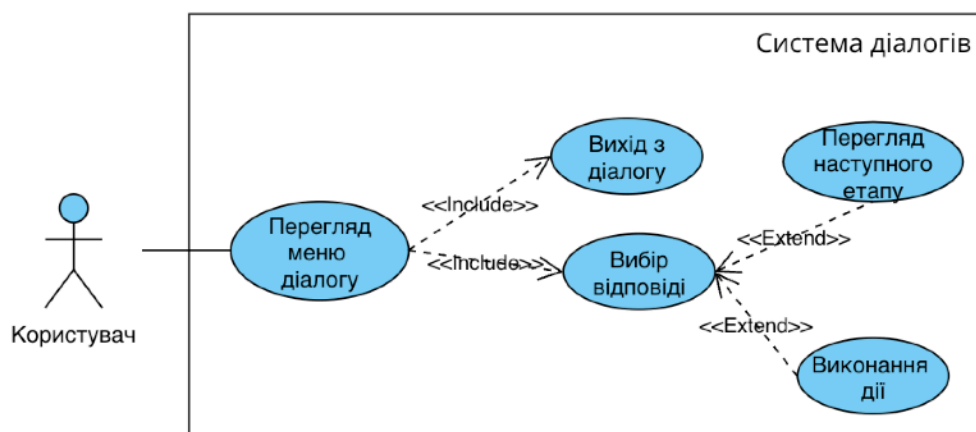


Рис 2.18 Діаграма прецедентів прецеденту «Використання системи діалогів»

Використання телефону: В користувача має бути доступний телефон, завдяки якому він може надсилати/отримувати повідомлення, телефонувати та приймати дзвінки, додавати або редагувати контакти у книзі, використовувати додатки.

Схему прецедентів зображено на рисунку 2.7.

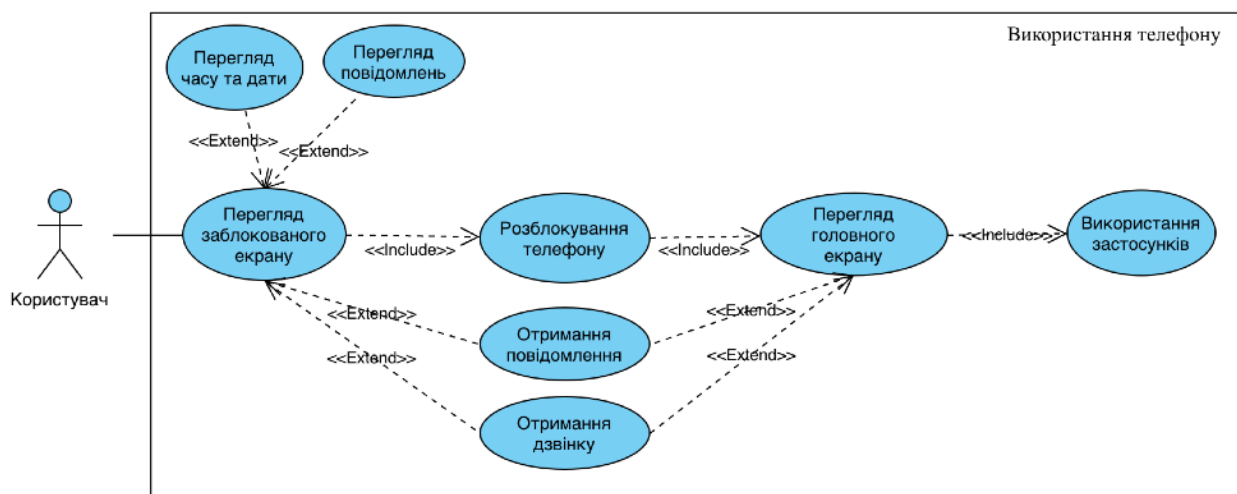


Рис 2.7 Діаграма прецедентів прецеденту «Використання телефону»

Меню інвентарю: Користувач може переміщувати, ділити, об'єднувати, передавати, викидати, використовувати предмети в інвентарі, може відкривати їх опис. Схему прецедентів зображено на рисунку 2.8.

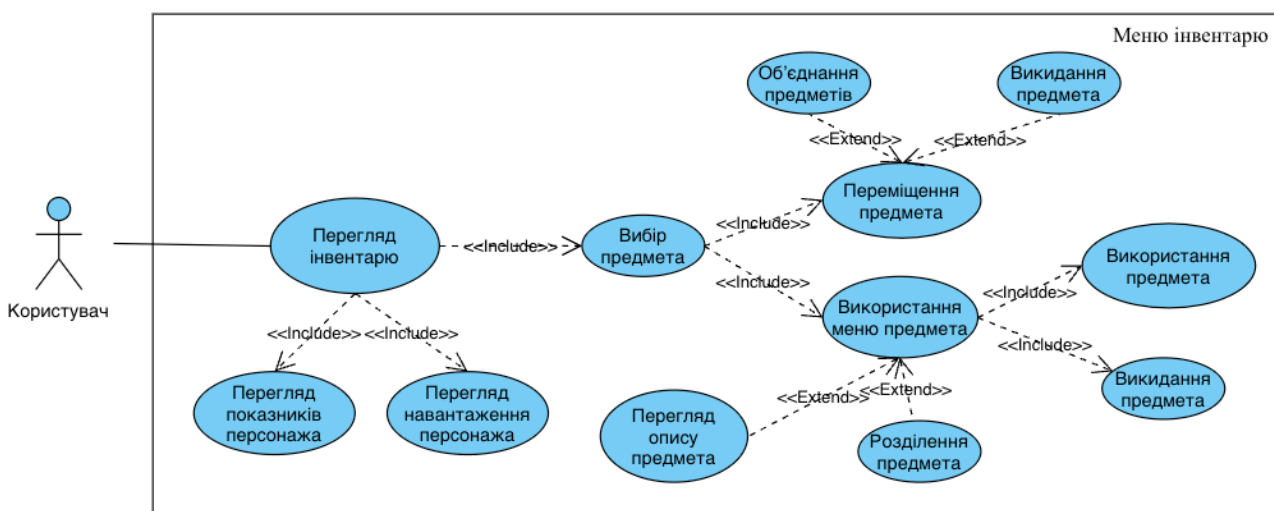


Рис 2.8 Діаграма прецедентів прецеденту «Використання інвентарю»

Використання меню взаємодії: Має складатись з опціональних компонентів, які доступні в залежності від об'єкту взаємодії. Це арешт гравця, запит на перегляд документів, перегляд інформації про нерухомість, лікування, взаємодія з транспортом. Схему прецедентів зображено на рисунку 2.9.

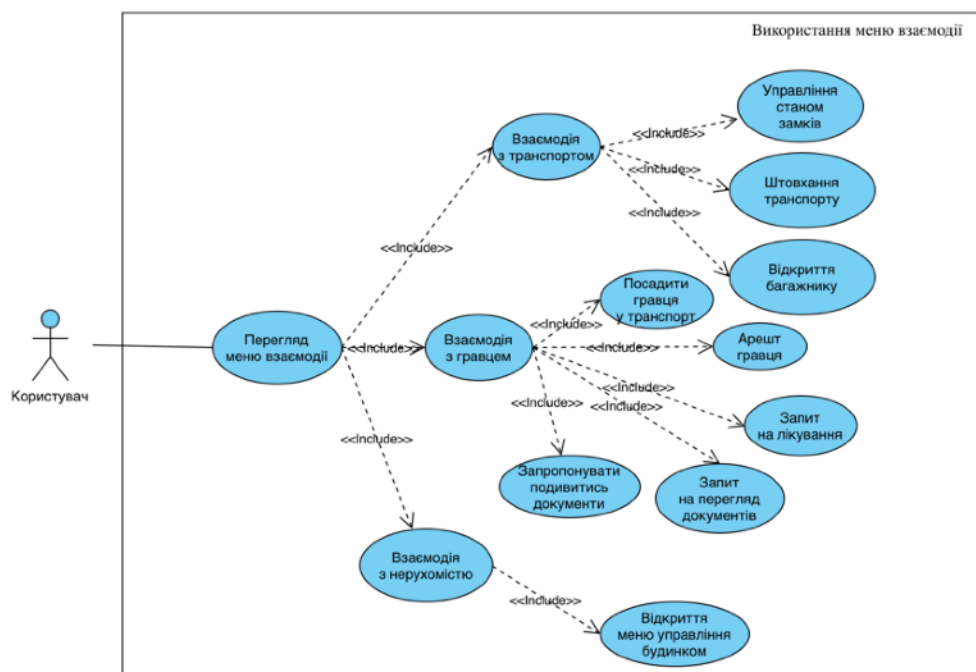


Рис 2.9 Діаграма прецедентів прецеденту «Використання меню взаємодії»

Використання заправної станції: Користувач може обрати об'єм для заправки, вибрати тип палива, та оплатити суму. Схему прецедентів зображено на рисунку 2.10.

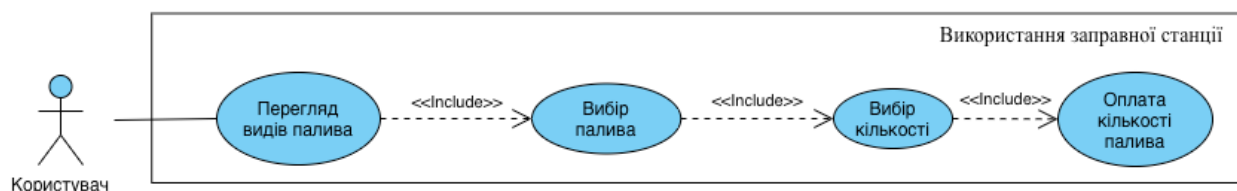


Рис 2.10 Діаграма прецедентів прецеденту «Використання заправної станції»

Використання головного інтерфейсу: Гравець може побачити рівень ситості, здоров'я, броні, кількість патронів, кількість грошей, час, місцезнаходження та багато іншого. Схему прецедентів зображено на рисунку 2.11

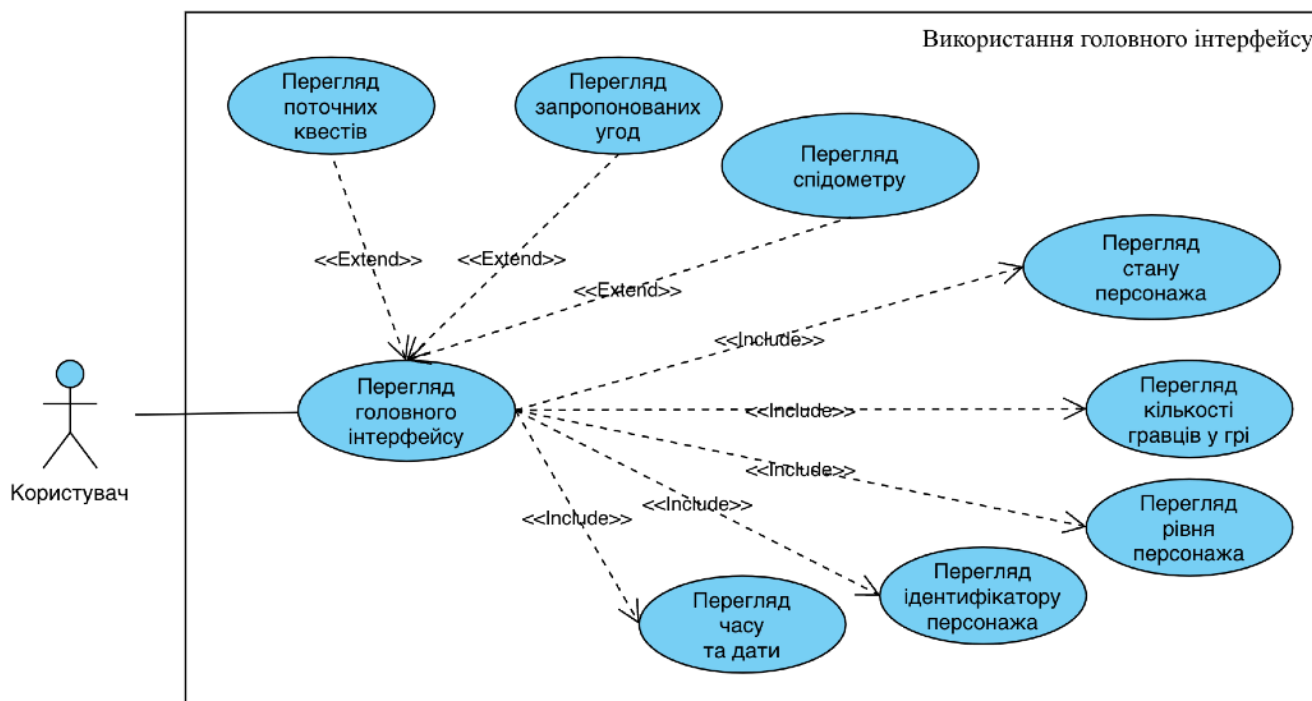


Рис 2.11 Діаграма прецедентів прецеденту «Перегляд головного інтерфейсу»

Використання екрану втрати свідомості: Екран має надавати можливість викликати доктора, прийняти запит на лікування, чи з'явитись у лікарні. Схему прецедентів зображено на рисунку 2.12.

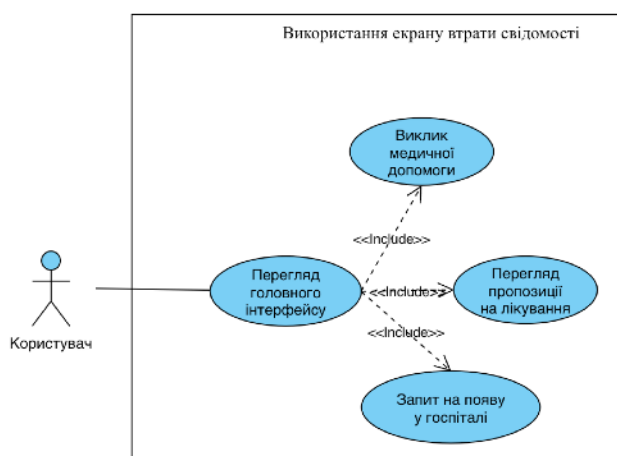


Рис 2.12 Діаграма прецедентів прецеденту «Перегляд екрану втрати свідомості»

Використання чату: Чат має мати згорнутий, та розгорнутий вигляд, мати підказки по командам, мати кольорове розділення для відіграних реплік чи дій, мати індикацію про кількість пропущених повідомлень. Схему прецедентів зображено на рисунку 2.13.

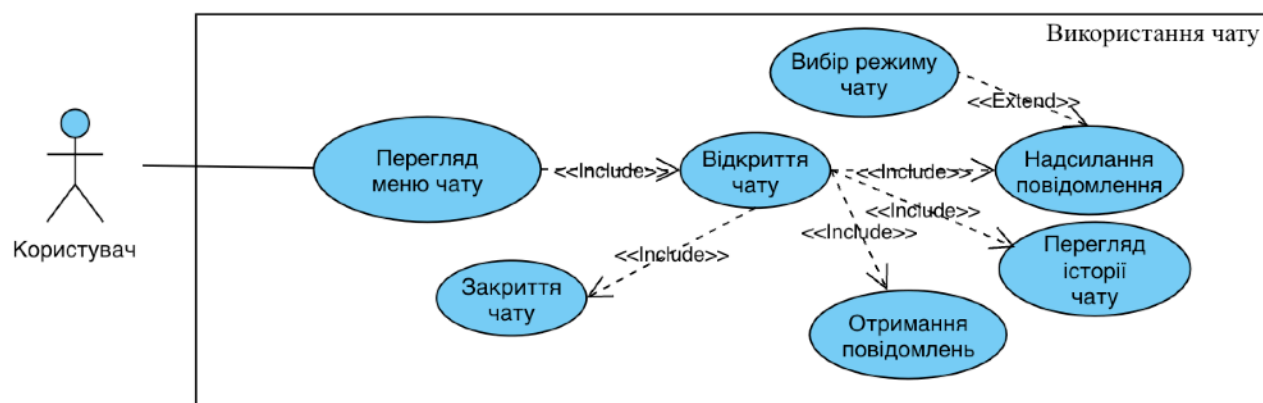


Рис 2.13 Діаграма прецедентів прецеденту «Використання чату»

Використання автошколи: Користувач може обрати категорію прав, які він хоче отримати, ознайомитись з правилами ПДД, придбати проходження теоретичного тесту, відповідати на задані питання, подивитись на результат проходження тесту, придбати проходження практичного тесту та отримати права. Схему прецедентів зображено на рисунку 2.15.

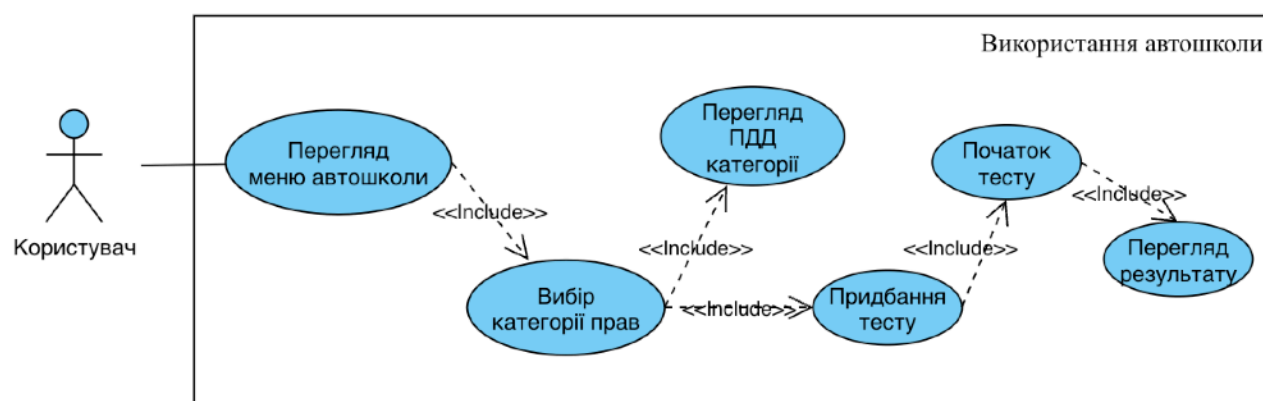


Рис 2.15 Діаграма прецедентів прецеденту «Використання автошколи»

Використання головного меню: В даному меню гравець має змогу подивитися повну статистику персонажу, кількість його грошей, витрати, штрафи, правопорушення, стан здоров'я, рівень, кількість унікальної валюти, також гравець може відправити скаргу, подивитись налаштування клавіш, придбати унікальні предмети у ігровому магазині, та побачити закінчені чи поточні квести. Схему прецедентів зображено на рисунку 2.14.

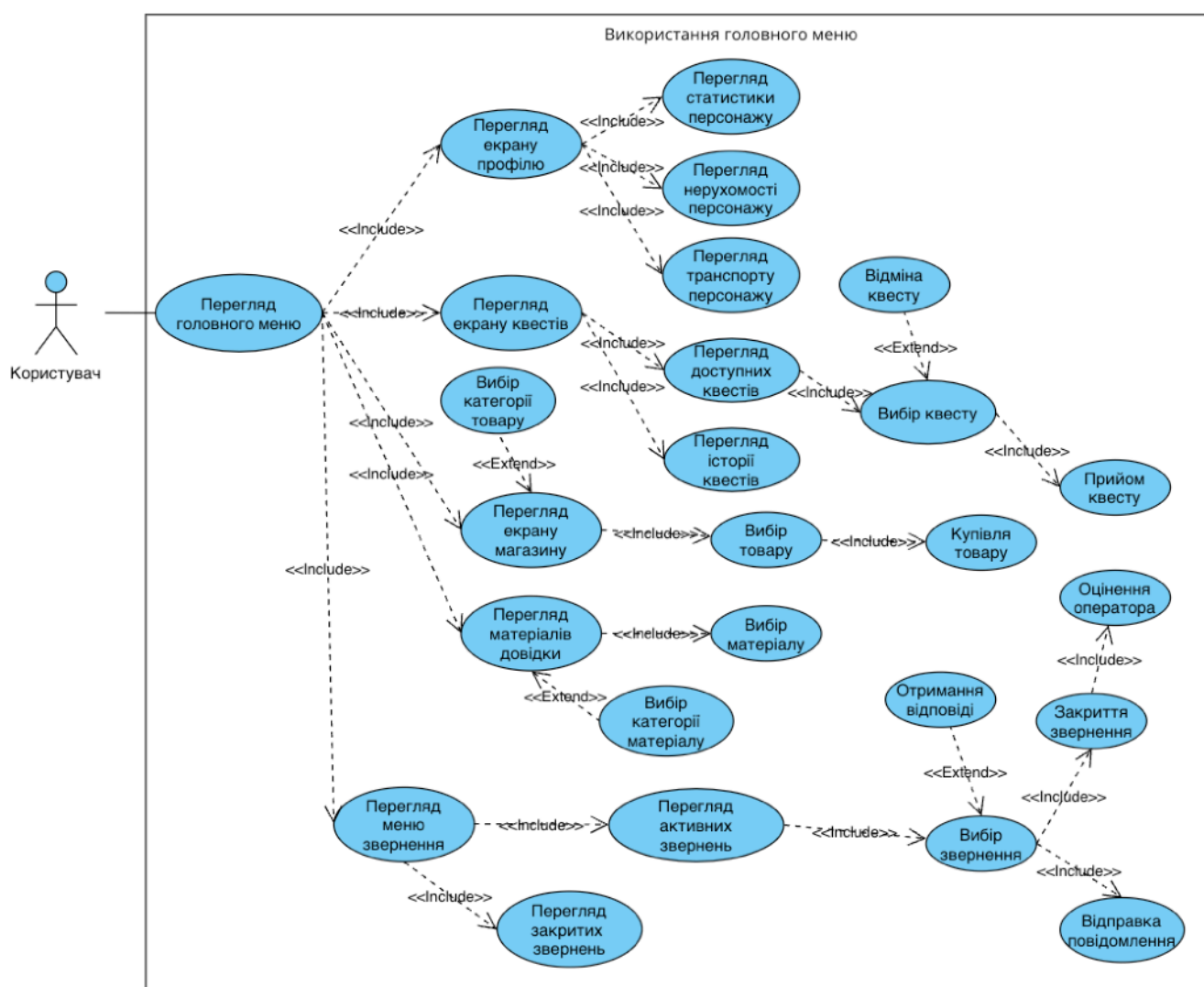


Рис 2.14 Діаграма прецедентів прецеденту «Використання головного меню»

Використання системи конструювання об'єктів одягу: Адміністратор може створювати об'єкти одягу, редагувати їх, переглядати створені об'єкти та їх характеристики. Схему прецедентів зображено на рисунку 2.16.

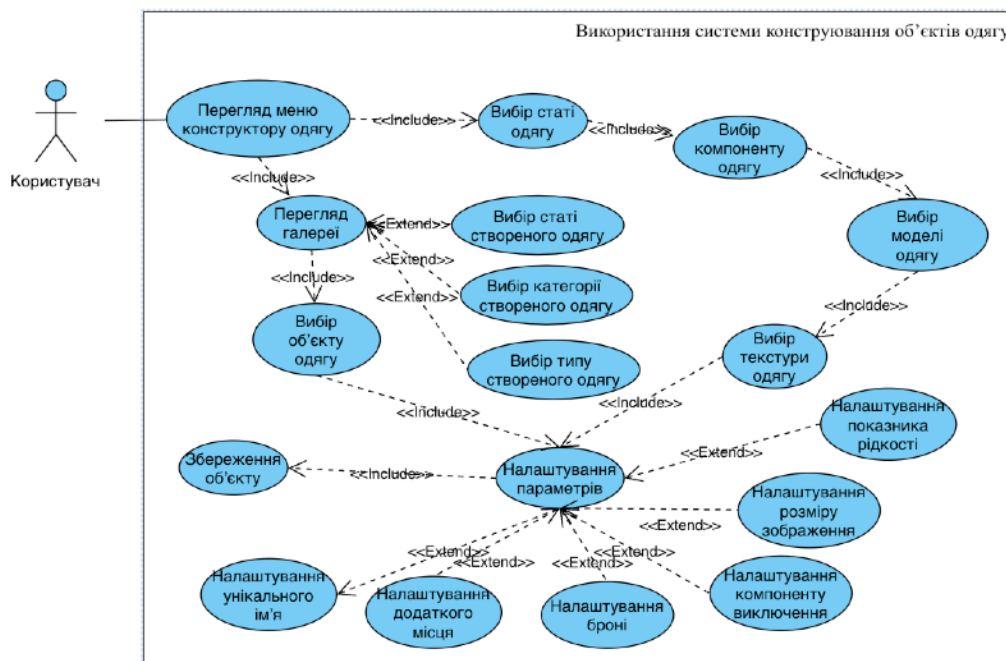


Рис 2.16 Діаграма прецедентів прецеденту «Використання системи конструювання об'єктів одягу»

Використання міні-гри сортування пігулок: Користувач медик може використати меню міні-гри «Сортування пігулок» для отримання преміальних грошей. Суть гри полягає у вірному сортуванні пігулок за обмежений час. Схему прецедентів зображено на рисунку 2.17.

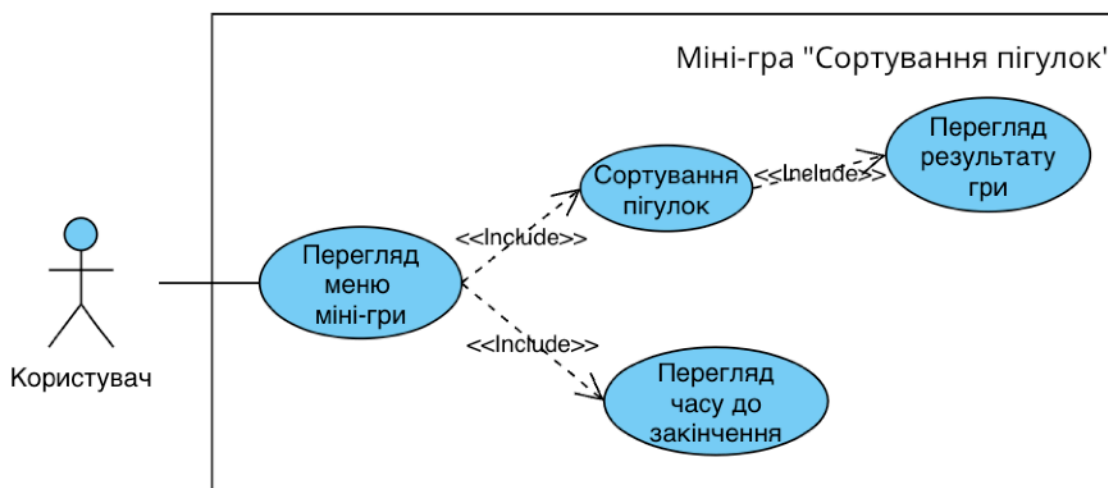


Рис 2.17 Діаграма прецедентів прецеденту «Міні-гра «Сортування пігулок»»

2.2 Розробка нефункціональних вимог для багатокористувацьких комп'ютерних рольових ігор

Нефункціональні вимоги - це обов'язкові вимоги, що характеризують якість системи з різних точок зору, зокрема продуктивності, надійності, сумісності, безпеки, масштабованості, підтримки, доступності, юзабіліті, міжнароднізації та локалізації. Їх використання в процесі проектування та розробки забезпечує високу якість продукту, його ефективну роботу в різних умовах та високий рівень задоволення користувачів. Нижче представлені основні нефункціональні вимоги до frontend-системи нашої гри:

Продуктивність: Frontend-система має забезпечувати високу продуктивність з мінімальним часом відгуку, оптимізацією використання ресурсів та ефективним завантаженням. Використання React.js забезпечує високу продуктивність за рахунок віртуального DOM і "lazy loading". Метрика: час відгуку менше 200 мс.

Сумісність: Frontend-система має бути сумісним з AltV і GTA V, забезпечуючи гладке використання коду та візуальну гармонію з дизайном GTA V. Сумісність досягнута за допомогою тісної співпраці з командою розробників AltV і використання Figma для проектування UI. Метрика: 100% функціоналу працює без помилок.

Надійність: Frontend-система має бути стабільною і відповідати на дії користувача без помилок та затримок. Досягається через детальне тестування і безпечне кодування. Метрика: 99,9% часу без збоїв.

Безпека: Всі дані користувачів мають бути захищені за допомогою сучасних методів шифрування. Час виявлення та виправлення хиб в безпеці не повинен перевищувати 24 години.

Масштабованість: Інтерфейс має бути готовим до масштабування, зберігаючи свою продуктивність та стабільність при зростанні числа користувачів. Метрика: здатність підтримувати стабільну роботу при збільшенні кількості користувачів, значення навантаження серверу не повинна перевищувати 30%.

Підтримка: Frontend-система повинна бути легко підтримуваним, включаючи простоту внесення змін, додавання нового функціоналу, виправлення помилок і оновлень. Метрика: час потрібний для виправлення помилок менше однієї доби.

Юзабіліті: Інтерфейс має бути інтуїтивно зрозумілим для користувачів, дозволяючи їм легко взаємодіяти з грою. Метрика: 95% користувачів відзначають високий рівень зручності використання за допомогою анкетування.

2.3 Можливості рушія AltV

AltV — це високопродуктивний мультиплеєрний двигун для Grand Theft Auto V, що надає розробникам потужний інструмент для створення незалежних багатокористувацьких середовищ. Він розроблений з метою забезпечення максимальної гнучкості і контролю над створюваним контентом, а також забезпечення високої стабільності і продуктивності великих мультиплеєрних серверів. AltV відрізняється від інших подібних двигунів своєю низькорівневою архітектурою, яка дозволяє більш точний контроль над грою, включаючи маніпуляції на рівні пакетів даних. Це дозволяє створювати більш ефективні, оптимізовані рішення, які можуть краще працювати на великих серверах з багатьма гравцями.

Головні можливості що надаються двигуном AltV включають в себе:

Мультиплеєрна архітектура: Забезпечує засоби для створення великих, стабільних серверів з багатьма гравцями.

Потужний API: Дозволяє контролювати більшість аспектів GTA V, включаючи AI, фізику, взаємодію з персонажами, графіку, звук і багато іншого.

Скриптування на JavaScript або C#: Надає гнучкість і потужність для створення багатих, динамічних багатокористувацьких середовищ.

Підтримка модифікацій: AltV підтримує широкий спектр модифікацій, що дозволяє розробникам змінювати ігрове середовище та додавати новий контент.

Низькорівневий контроль: AltV дає розробникам змогу контролювати ігрові процеси на низькому рівні, що дозволяє створювати більш оптимізовані, ефективні рішення.

Безпека: AltV включає в себе ряд механізмів безпеки для захисту від читів, а також має вбудовані засоби для боротьби з DDOS-атаками.

Згідно з документацією платформи AltV[9] надає великий список функціональних можливостей для взаємодії з WebView, серед них є такі як:

alt.getByID - надає можливість ідентифікувати активні WebView у стеку для подальшої взаємодії.

alt.emit - функція для відправки даних з WebView через WebSocket на сторону клієнта.

alt.on - функція реєстрації обробника події, яка дозволяє додати обробник до стеку обробників

alt.off - функція для від'язання обробника події, яка дозволяє видалити обробник зі стеку.

alt.once - дозволяє виконати обробник події лише один раз, після чого він буде автоматично від'язаний.

alt.focus - дозволяє надати інтерфейсу активного статусу, для взаємодії користувача з ним.

alt.unfocus - дозволяє прибрати активний статус з інтерфейсу.

Серед властивостей WebView, які виявились надзвичайно зручними:

isVisible - властивість налаштування стану видимості інтерфейсу.

url - властивість для задання відносного посилання на сторінку білду інтерфейсу.

all - властивість що дозволяє отримати всі веб-елементи, що відображені на клієнтському веб-інтерфейсі.

count - властивість що дозволяє отримати кількість активних веб-елементів на клієнтському веб-інтерфейсі.

isOverlay - властивість, яка вказує, чи є веб-елемент накладеним на гру.

isReady - це властивість, яка вказує, чи готовий веб-елемент до відображення та взаємодії.

focused - властивість, яка вказує, чи є веб-елемент активним для взаємодії користувача.

size - це властивість, яка повертає поточний розмір веб-елемента, його ширину та висоту.

pos - це властивість, яка повертає позицію веб-елемента на екрані у координатній сітці x та y.

2.4 Розробка алгоритму взаємодії користувача багатокористувацької рольової гри

У контексті AltV, взаємодія користувача з сервером через інтерфейс набуває специфічних особливостей. Alt:V використовує WebView для створення користувацького інтерфейсу, що дозволяє використовувати сучасні веб-технології для створення графічного інтерфейсу користувача.

Серед ситуацій взаємодії:

Користувач взаємодіє з грою через інтерфейс, створений за допомогою WebView. Дані вводу відправляються на клієнт користувача, після чого відправляються на сервер, для отримання актуальної інформації для відображення. Всі дії користувача, від входу до системи до взаємодії в грі, відбуваються через цей інтерфейс. Схему взаємодії представлено на рисунку 2.22.

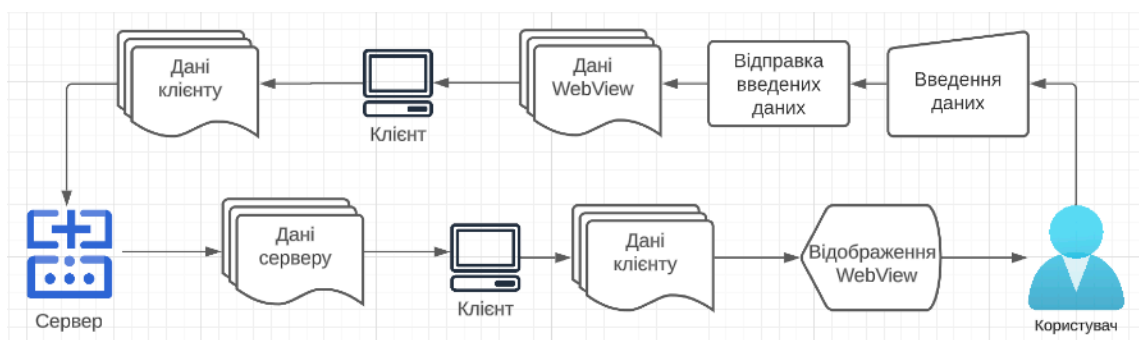


Рис 2.22 Взаємодія користувача з сервером шляхом вводу даних

WebView отримує дані від сервера без запиту і відображає відповідні дані для користувача. Використовується у випадках примусового оновлення інтерфейсу, при виклику серверної події. Схему взаємодії представлено на рисунку 2.23.

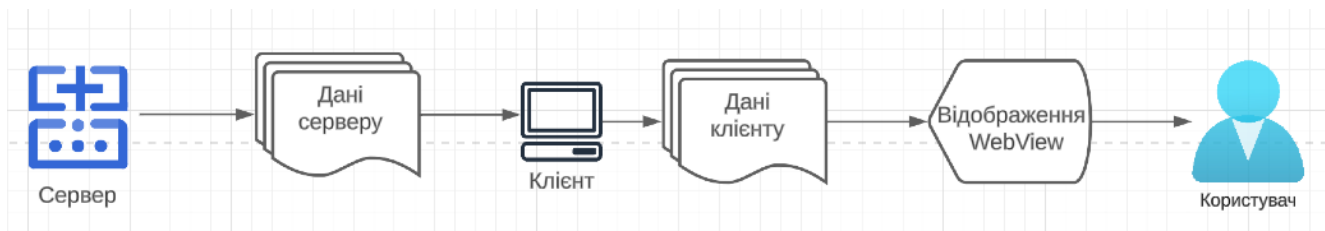


Рис 2.23 Примусове оновлення даних без участі користувача

Обмін інформацією між WebView та сервером без участі гравця використовується у випадках коли час виконання дії зберігається у WebView. Цей тип роботи є виключенням та використовується у специфічних ситуаціях. Схему взаємодії представлено на рисунку 2.24.

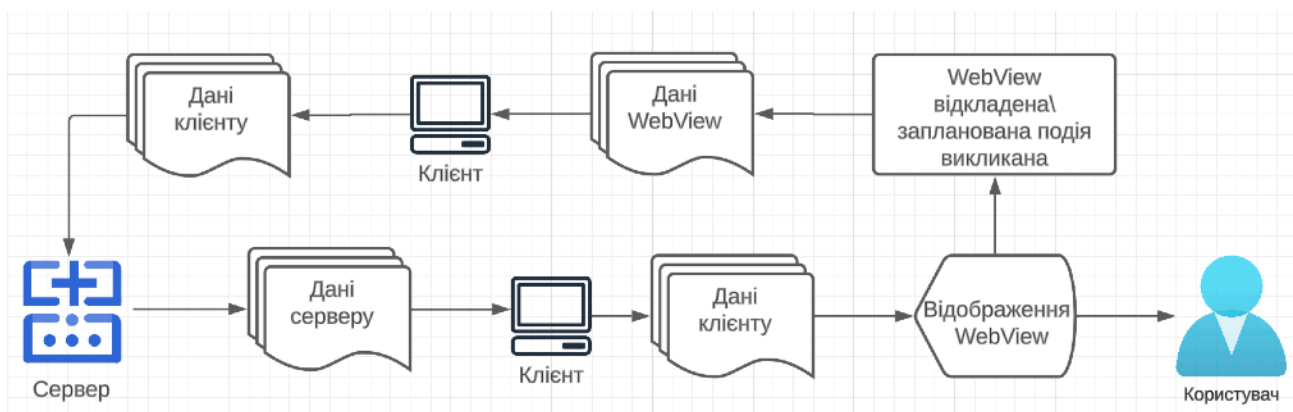


Рис 2.24 Запланований запит на оновлення даних без участі користувача

Всі дані ситуації відбувається за допомогою HTTP запитів та WebSocket з'єднань. Це забезпечує швидкий та ефективний обмін даними, що є важливим для забезпечення плавної та реактивної гри.

3. РОЗРОБКА ІНТЕРФЕЙСІВ ТА АЛГОРИТМІВ РОБОТИ

3.1 Створення UI/UX дизайну з використанням платформи Figma

Для розробки інтерфейсу користувача за допомогою платформи Figma, спочатку було створено макети майбутнього інтерфейсу. Макет інтерфейсу переставлено на рисунку

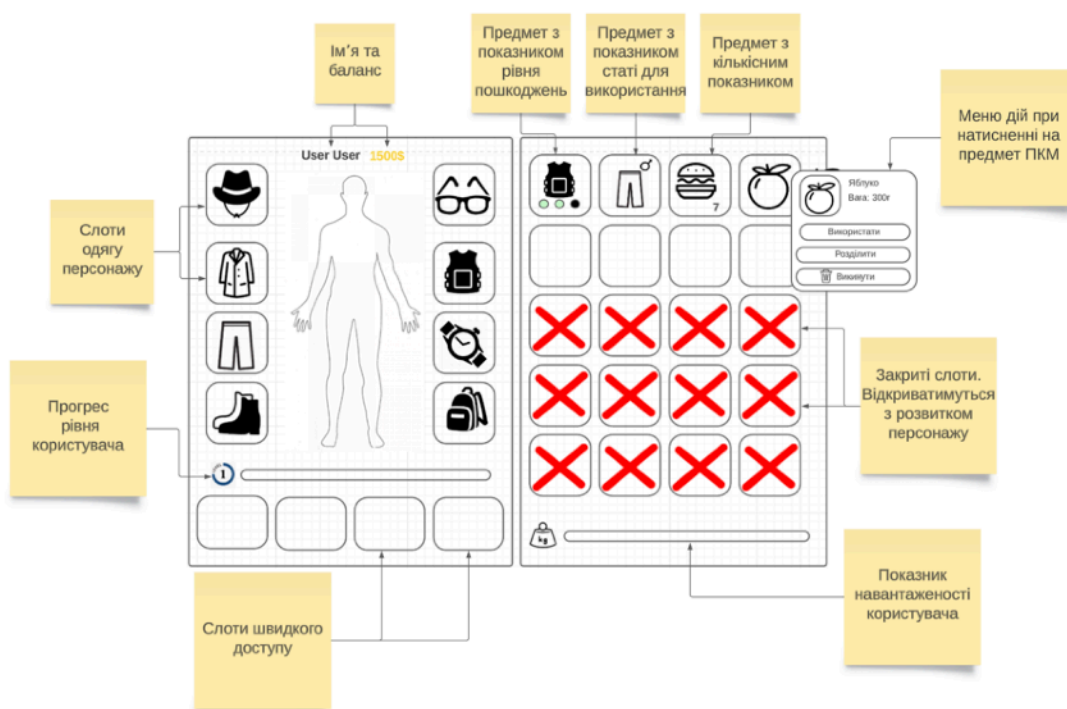


Рис 3.1 Макет інтерфейсу інвентарю

Після розробки макету на його основі було створено макет у Figma, варто зазначити, що в процесі розробки інтерфейсу деякі графічні елементи були спрощені, або змінені для більшої зручності використання кінцевим користувачем.

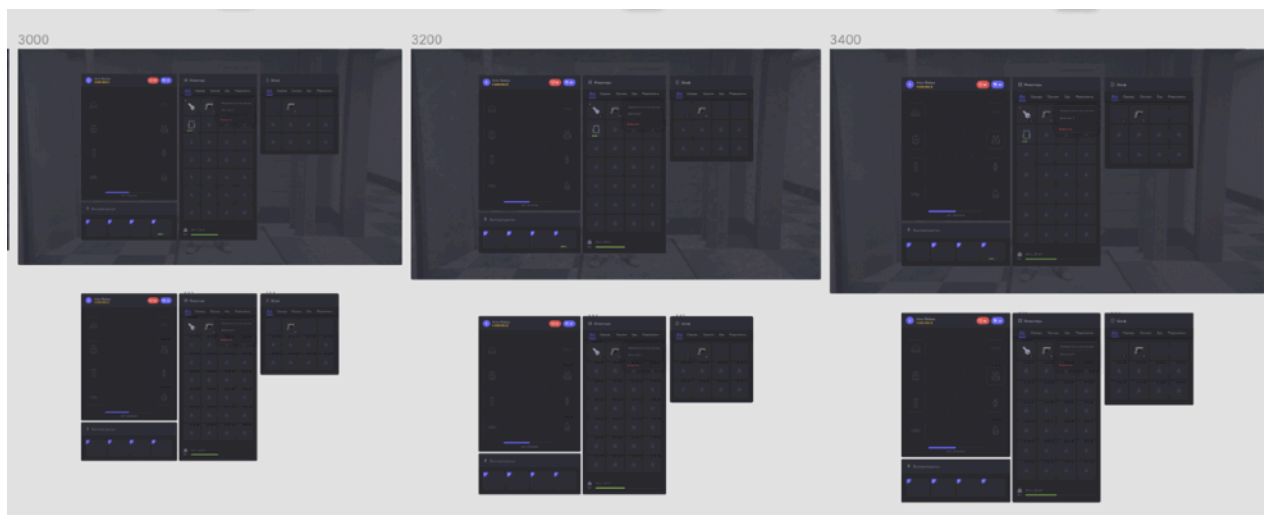


Рис 3.2 Макет створений на платформі Figma

Після того як всі макети інтерфейсів були створені у Figma почалось їх розробка за допомогою фреймворку React.js.

3.2. Розробка інтерфейсу реєстрації та аутентифікації

Система реєстрації та авторизації є вкрай важливою системою. Завдяки ній користувачі можуть створити обліковий запис, та зберегти в ньому дані про гру. Завдяки системі авторизації у користувача є змога перенести прогрес своєї гри на інший пристрій. Схема роботи меню представлена на рисунку 3.3.

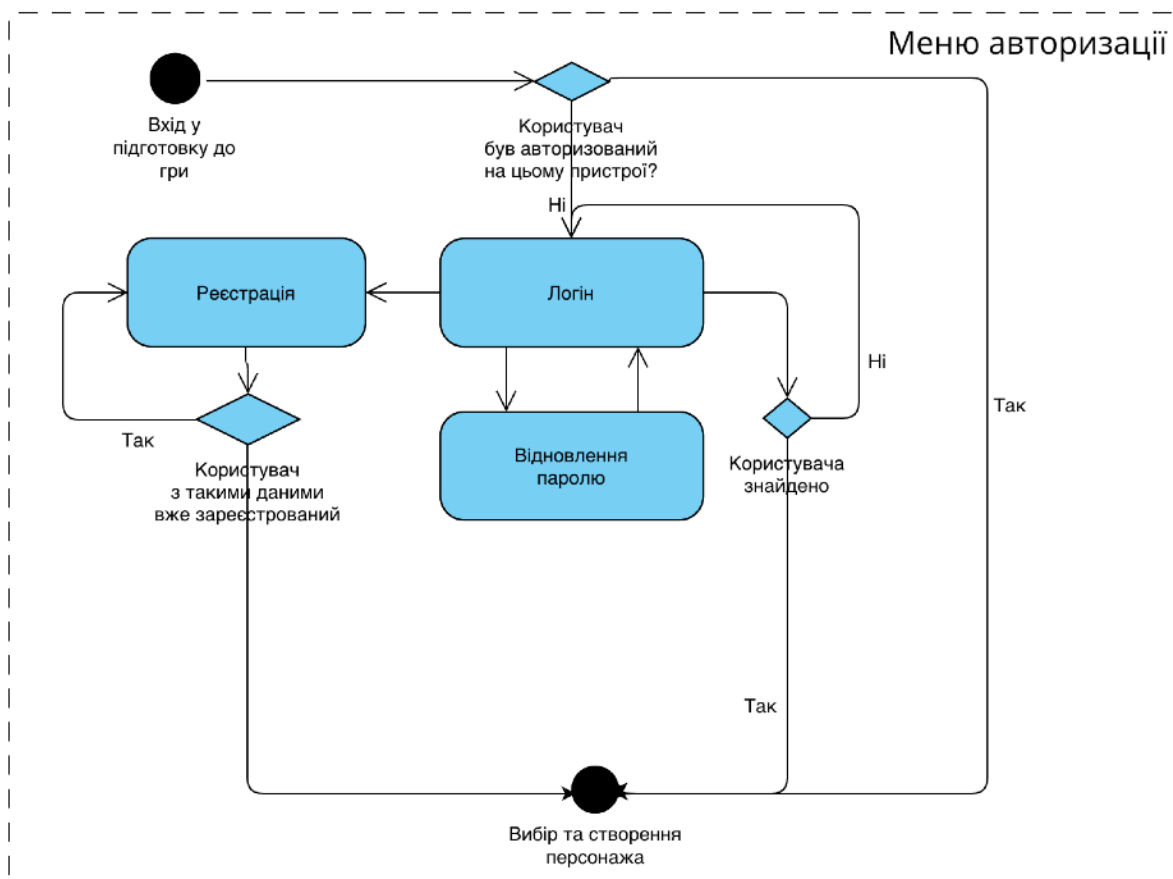


Рис 3.3 Схема роботи меню авторизації

Екранна форма розробленого меню представлена на рисунку 3.4.

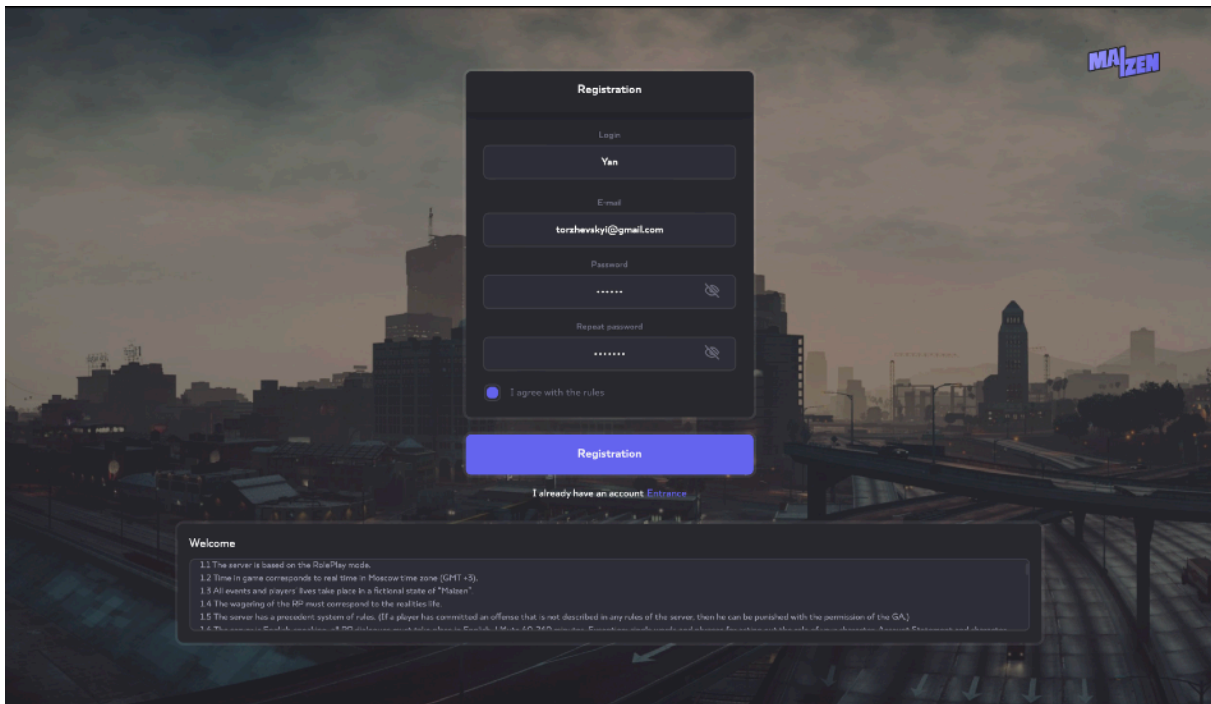


Рис 3.4 Екранна форма авторизації

3.3. Розробка геймплейних систем та систем меню

Для рольових комп'ютерних ігор дуже важлива роль персонажа, але користувачі не завжди хочуть знаходитись в рамках однієї ролі, тож необхідним компонентом для комфортної рольової гри є система створення та управління створеними персонажами. Алгоритм роботи меню представлена на рисунку 5.1.

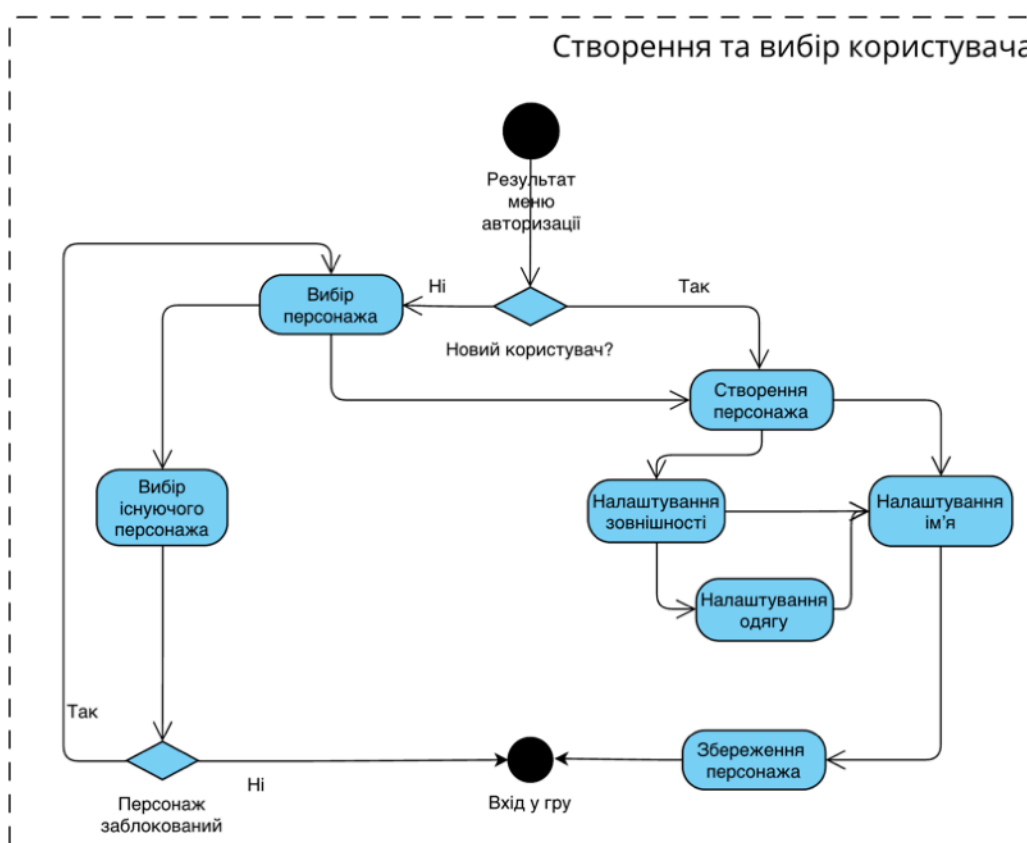


Рис 3.5 Алгоритм роботи меню створення та вибору персонажа

Екранна форма розробленого меню статистики представлена на рисунку 3.6, екранна форма розробленого меню налаштування зовнішності персонажа представлена на рисунку 3.7.

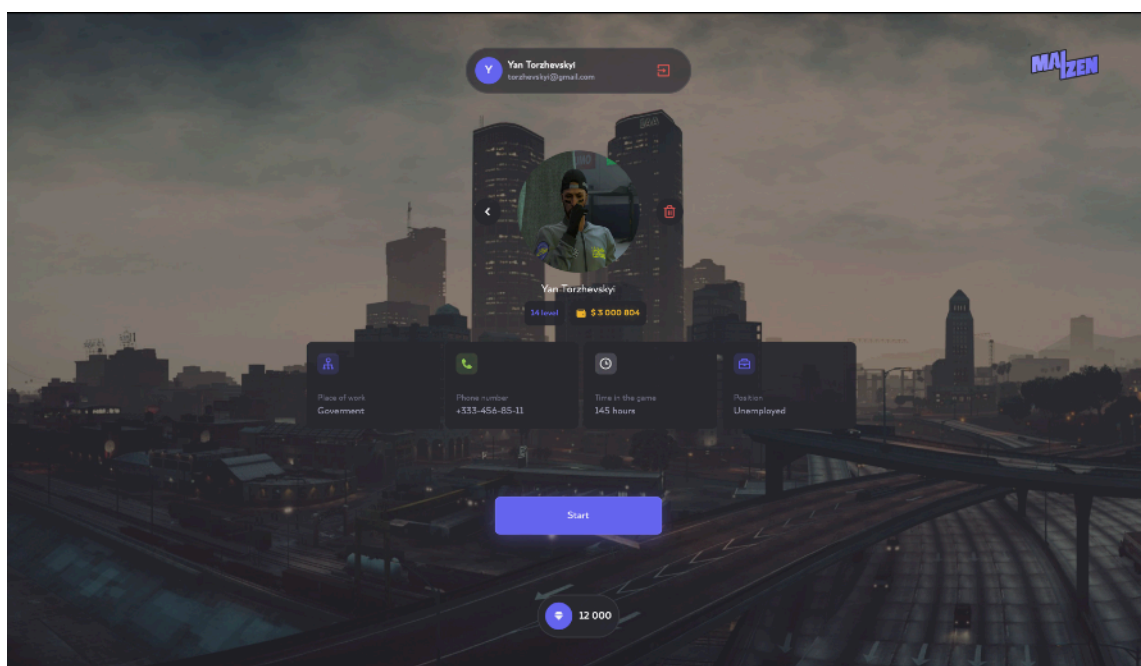


Рис 3.6 Екранна форма статистики персонажа



Рис 3.7 Екранна форма налаштування зовнішності персонажа

Розробка екрану втрати свідомості. Екран втрати свідомості доволі проста, але вкрай необхідна річ. Вона дозволяє збалансувати час до повернення гравця на місце протиборства компаній або банд.

Також даний екран надає необхідність у виклику медика, що додає гравцям з даною роллю роботи. Алгоритм роботи даного меню представлено на рисунку 3.21.

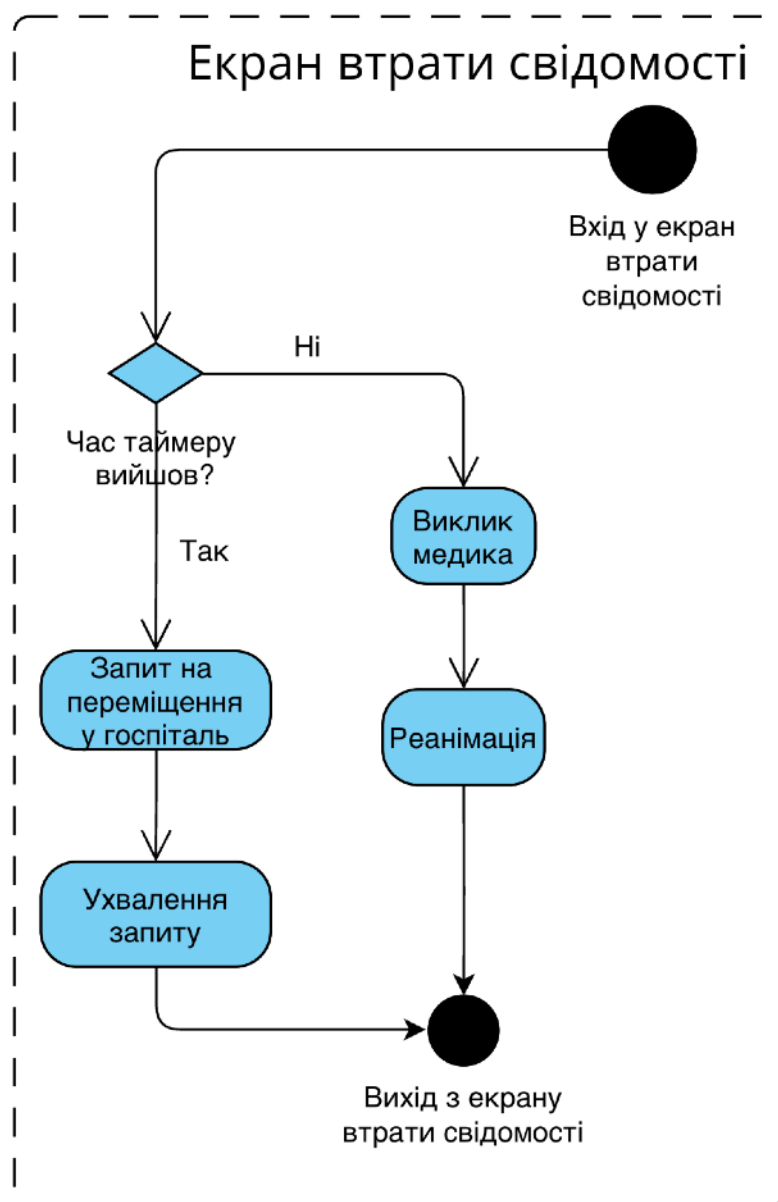


Рис 3.21 Алгоритм роботи екрану втрати свідомості

Екранна форма системи представлена на рисунку 3.22.

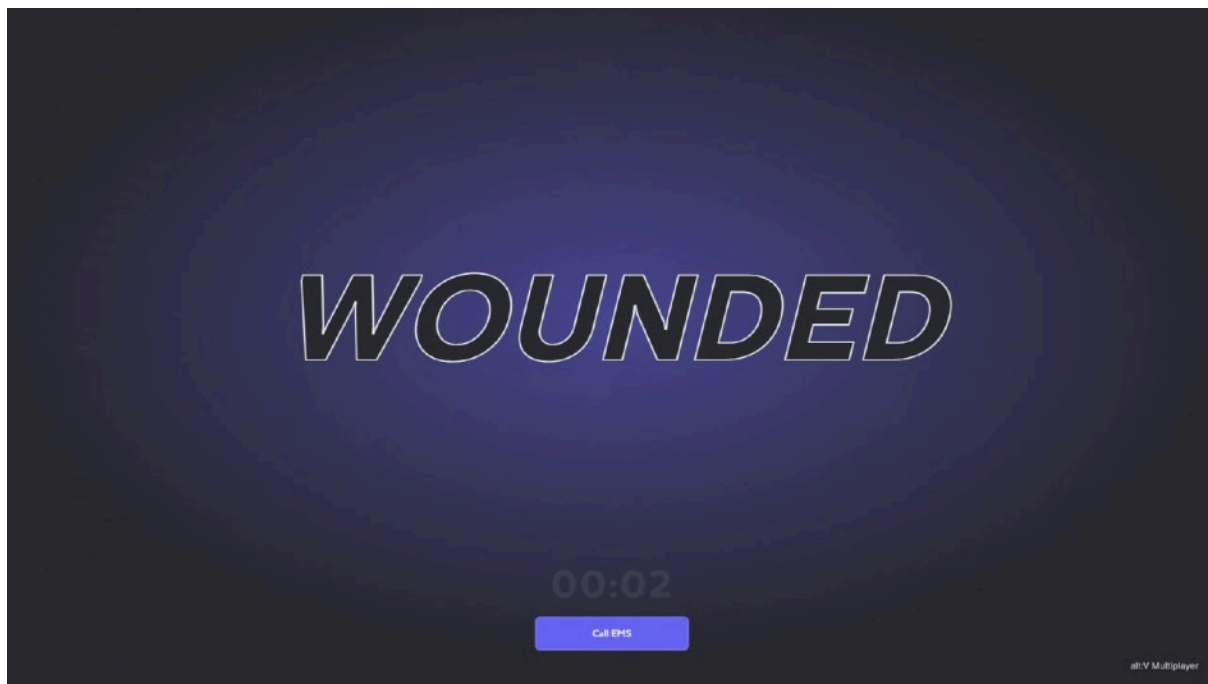


Рис 3.22 Екранна форма екрану втрати свідомості

Розробка магазинів. Магазины у рольовій грі є найважливішим компонентом. Саме вони надають можливість для кастомізації персонажа, та підтримки його характеристик. Алгоритм роботи системи магазинів наведено на рисунку 3.23

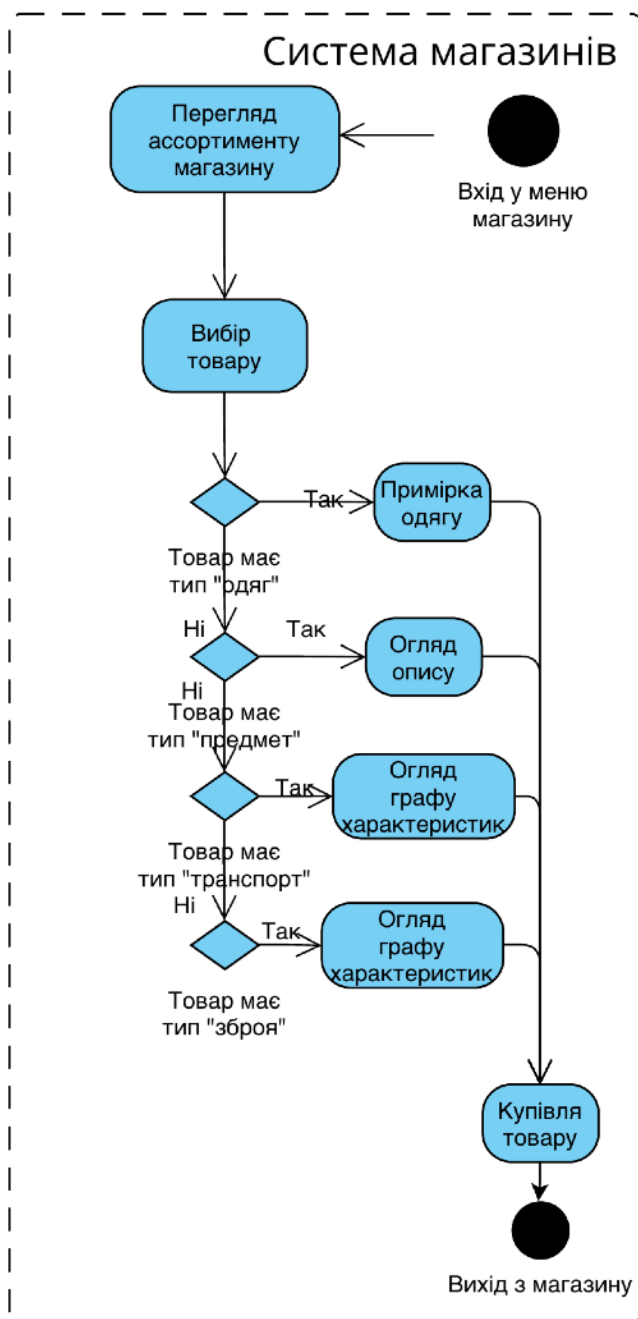


Рис 3.23 Алгоритм використання системи магазинів

Екранні форма системи магазину зброї представлена на рисунку 3.24, екранна форма системи універсального магазину зображена на рисунку 3.25, на рисунку

3.26 представлена екранна форма магазину одягу, екранна форма магазину транспорту наведена на рисунку 3.27.

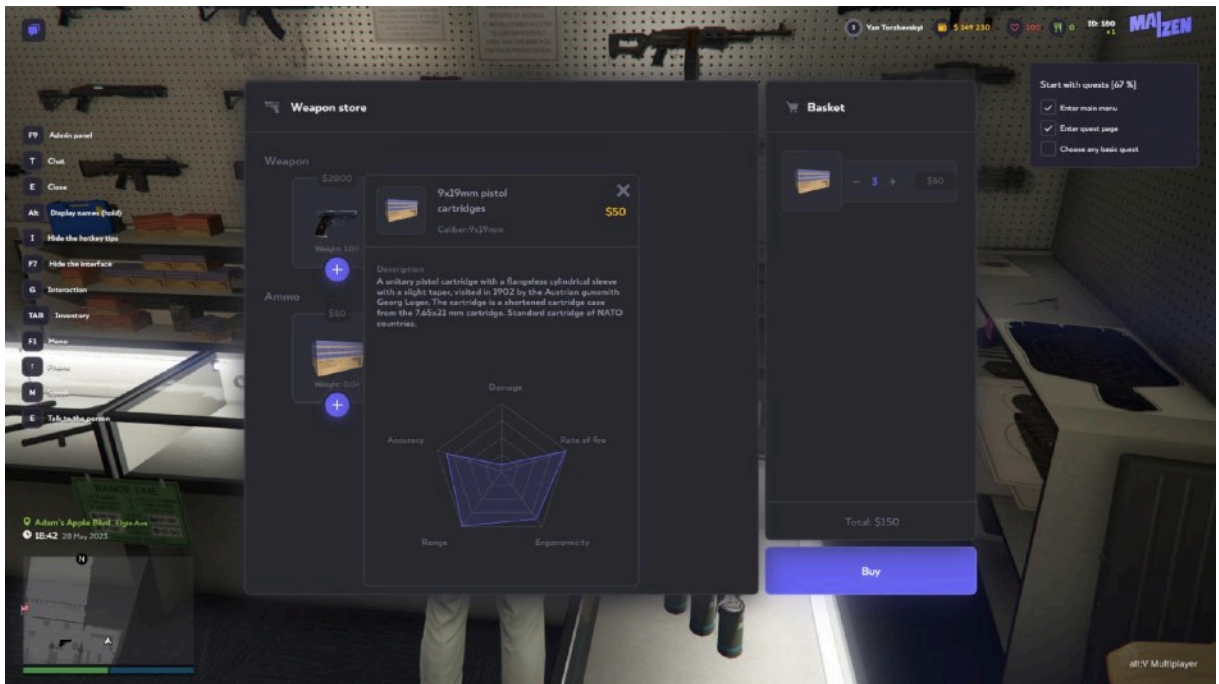


Рис 3.24 Екранна форма магазину амуніції



Рис 3.25 Екранна форма універсального магазину

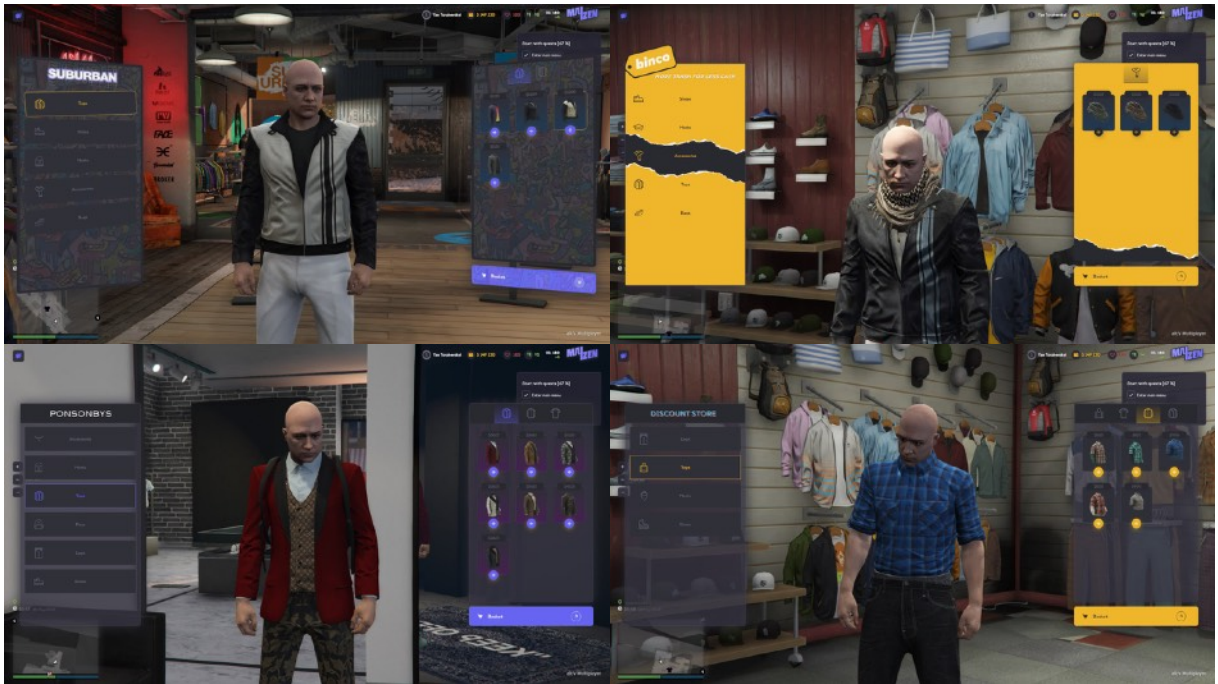


Рис 3.26 Екранні форми чотирьох стилів магазину одягу

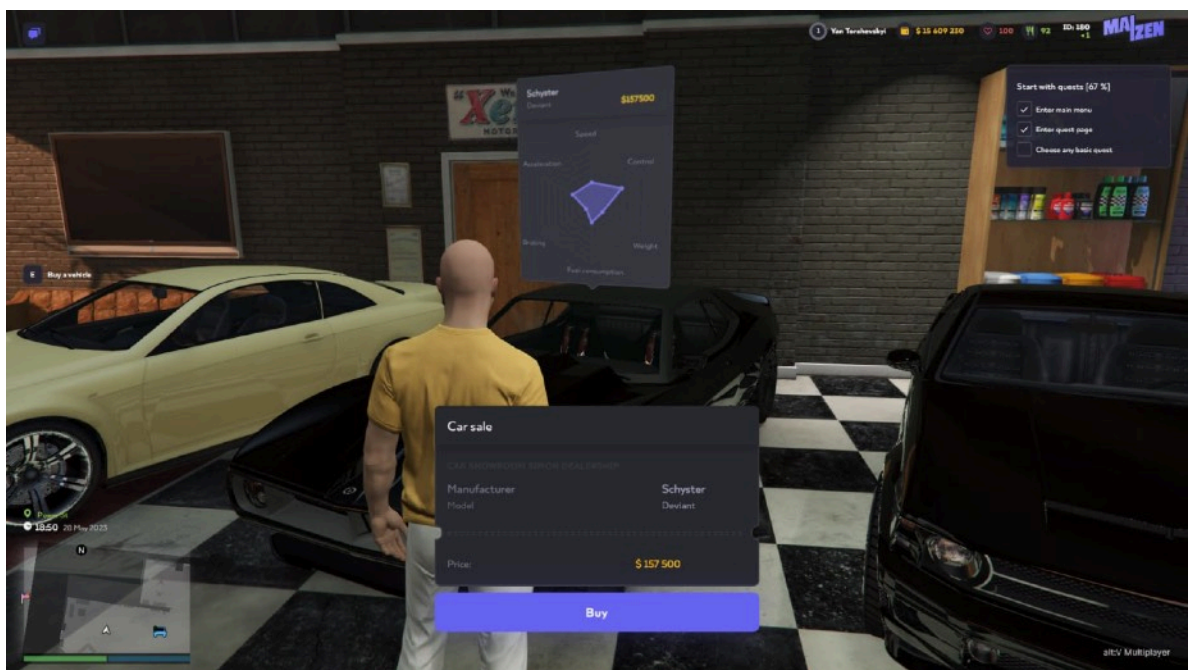


Рис 3.27 Екранна форма купівлі транспорту

Розробка системи управління нерухомістю. Кожен гравець може покращувати, продавати та оплачувати комунальні платежі своєї нерухомості. Алгоритм роботи даного меню представлено на рисунку 3.28.

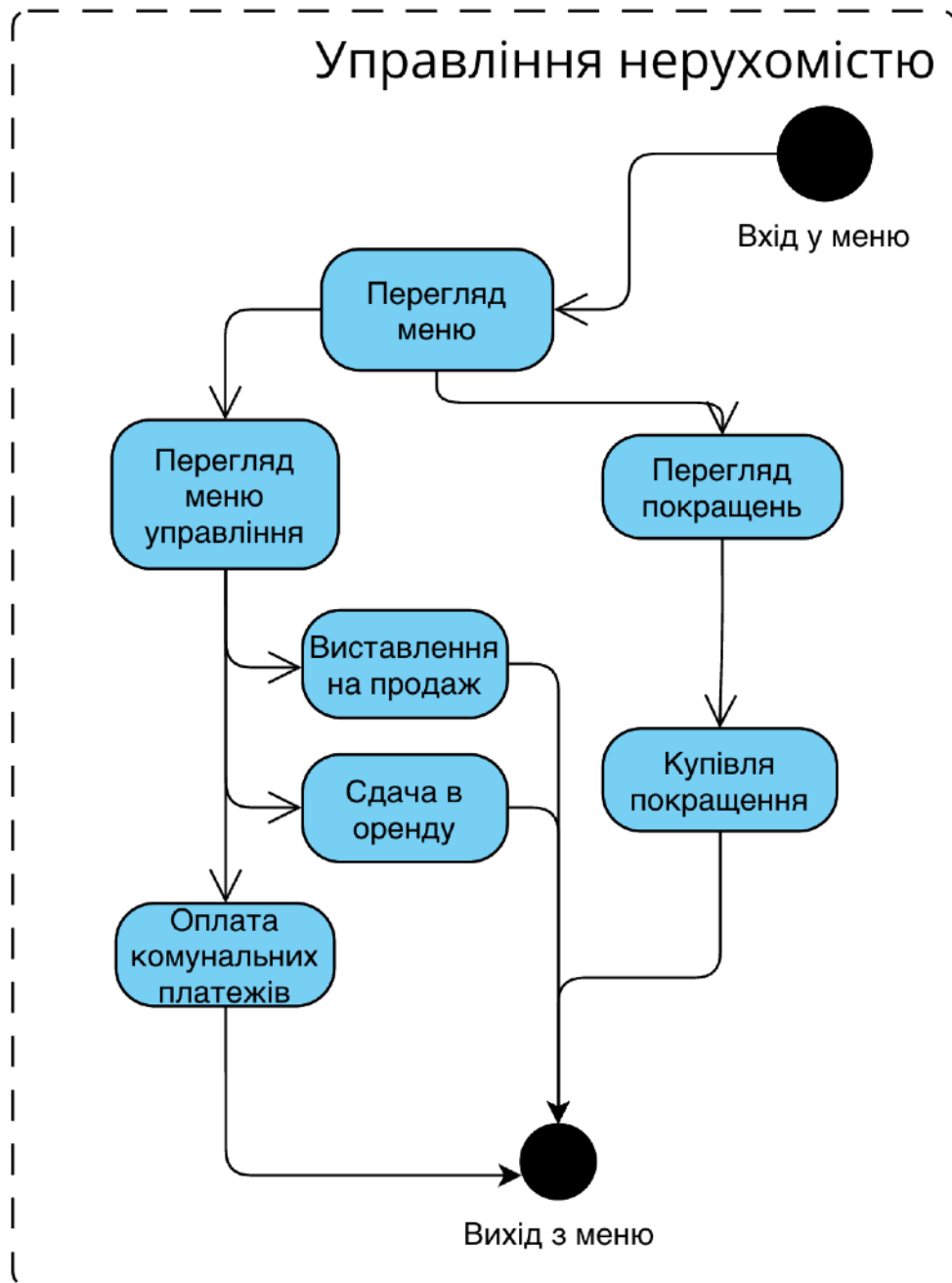


Рис 3.28 Алгоритм управління нерухомістю

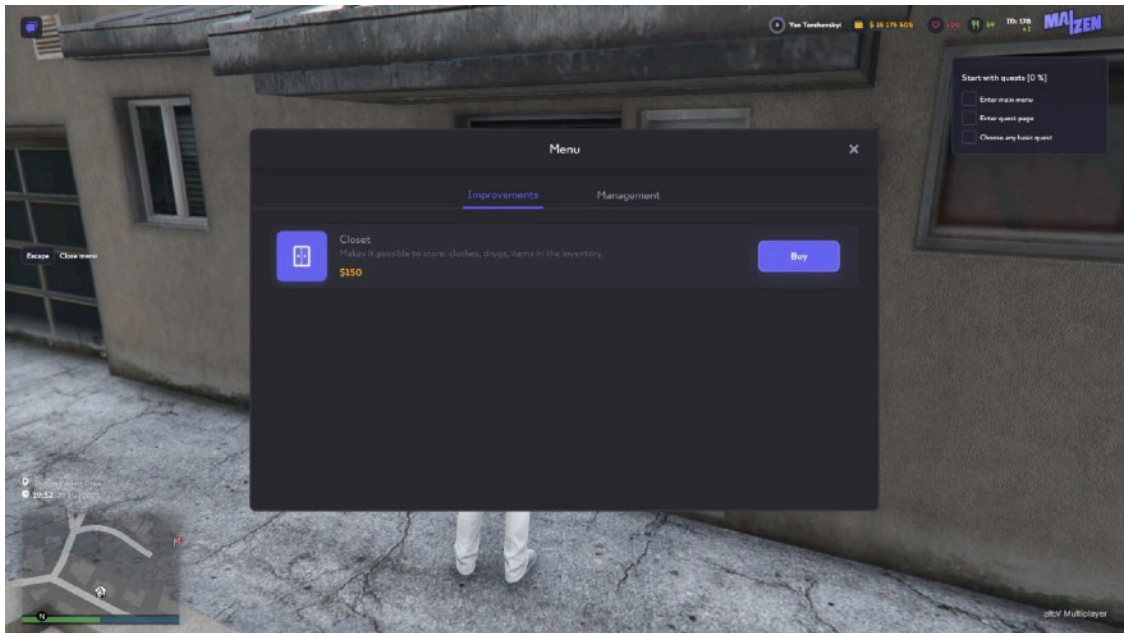


Рис 3.29 Екранна форма управління нерухомістю

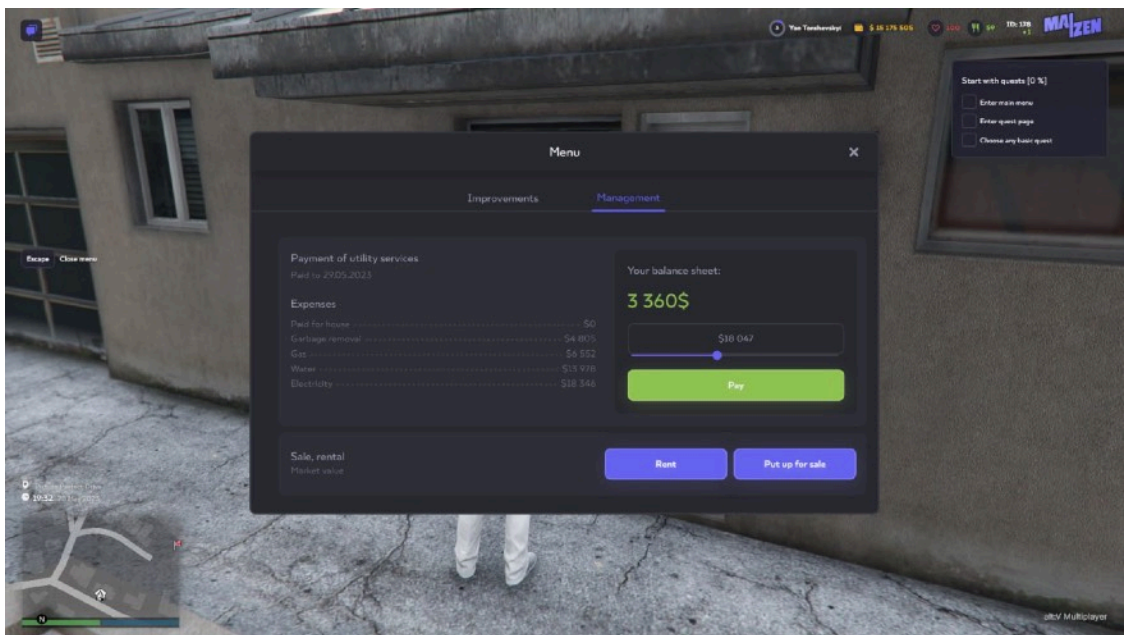


Рис 3.30 Екранна форма покращення нерухомості

Розробка заправної станції. Заправна станція це система яка дозволяє користувачу поповнювати запас пального в його транспорті, що також добре впливає на процес занурення у рольовий світ гри. Алгоритм роботи даної систем представлено на рисунку 3.31.

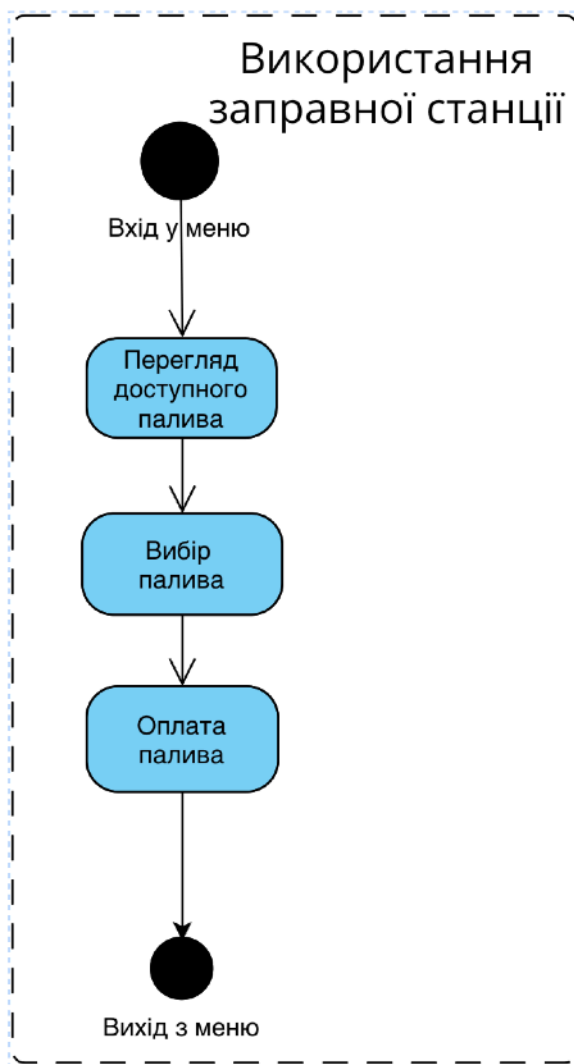


Рис 3.31 Алгоритм роботи заправної станції

Екранна форма заправної станції представлена на рисунку 3.32.



Рис 3.32 Екранна форма заправної станції

Розробка автошколи. Для покращення ситуації на дорогах у грі, було створено автошколу завдяки якій гравець може не тільки дізнатися основні правила віртуального світу, а й покращити свої навички водіння. Алгоритм роботи автошколи представлено на рисунку 3.33.

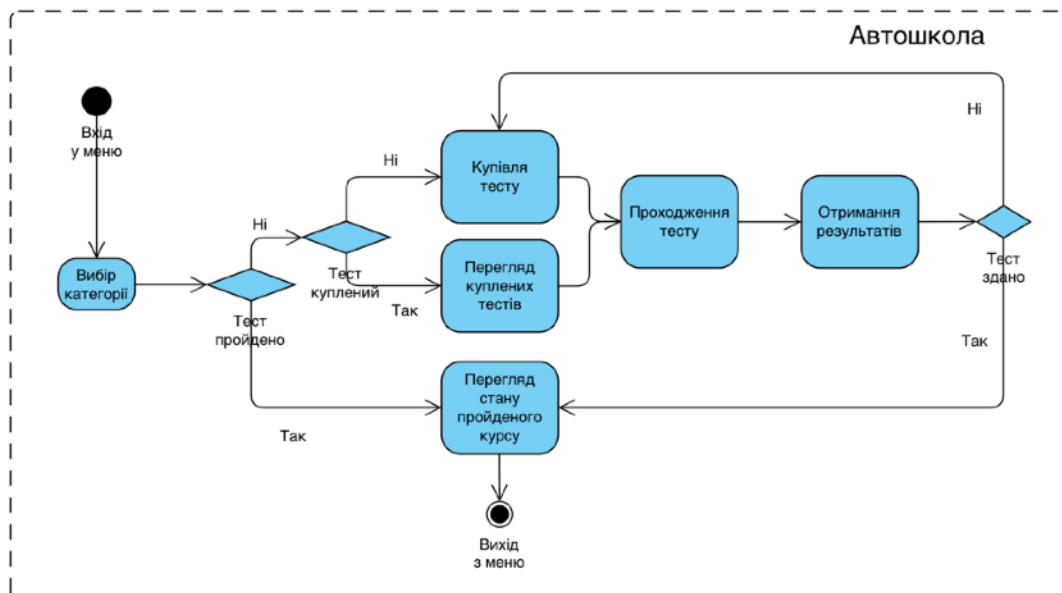


Рис 3.33 Алгоритм роботи автошколи

Екранна форма меню тестів системи представлена на рисунку 3.34, екранна форма процесу проходження тесту зображена на рисунку 3.35.

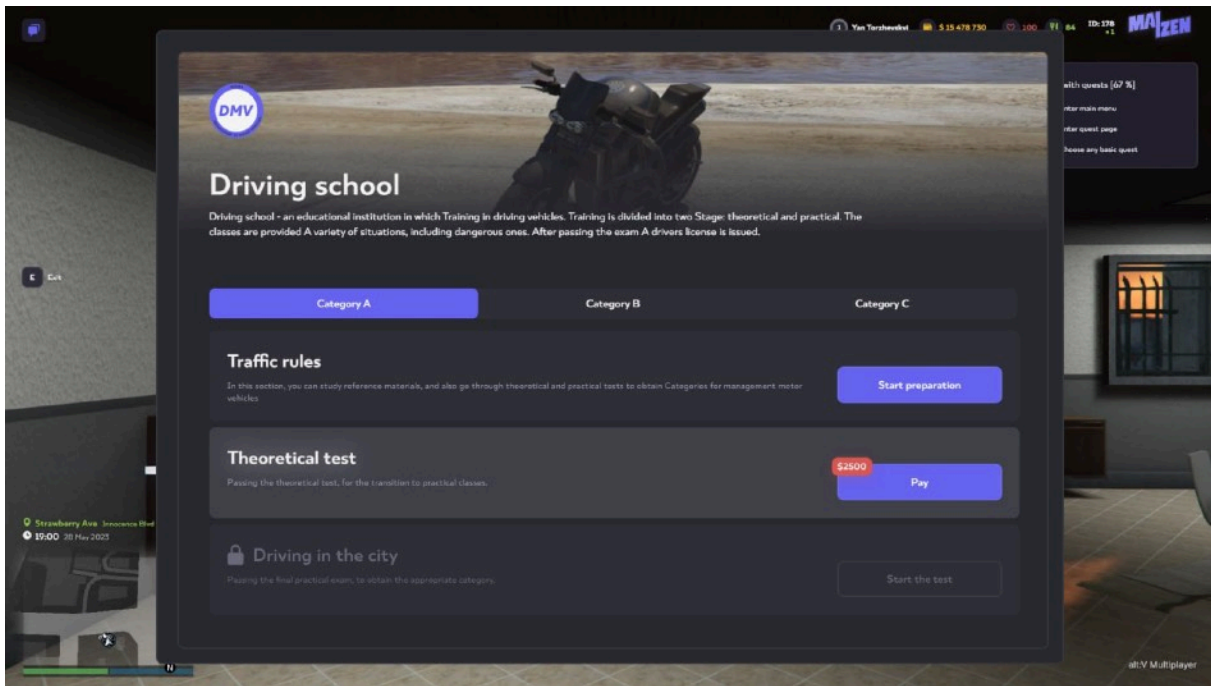


Рис 3.34 Екранна форма купівлі тесту

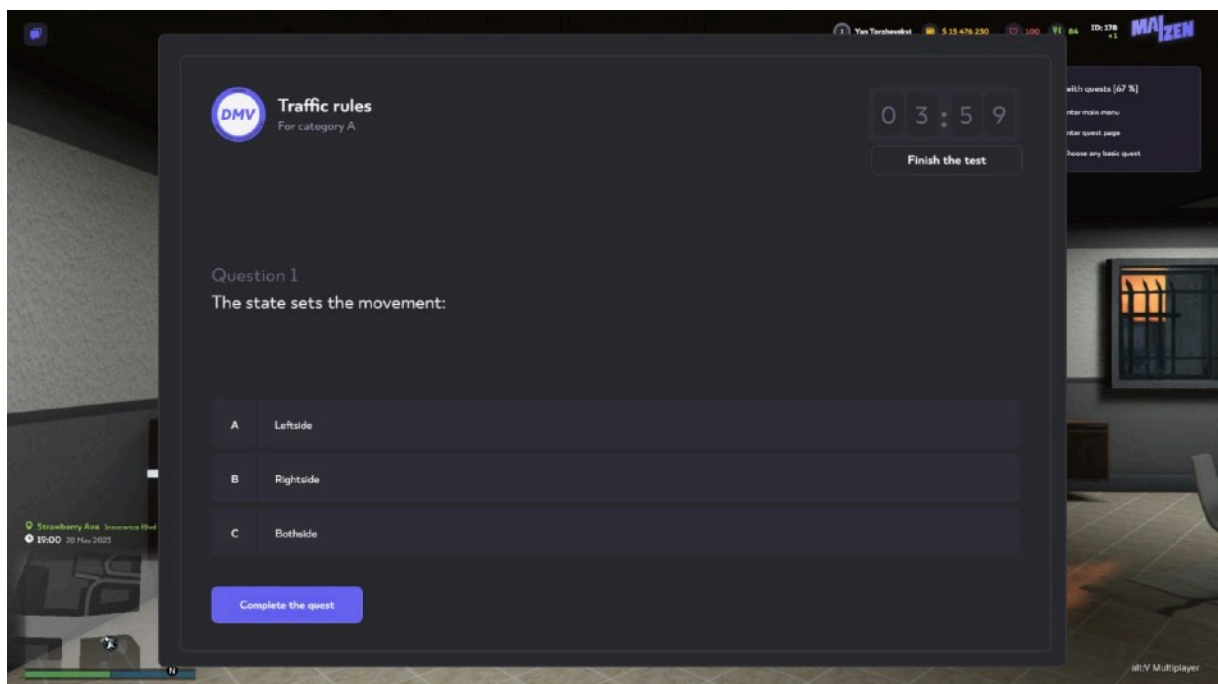


Рис 3.35 Екранна форма проходження тесту

Розробка міні-гри «Сортування пігулок». Дана гра була розроблена для працівників сфери медицини. За допомогою даної гри медики можуть отримати премію за кожен правильно відсортовану пігулку. Алгоритм роботи міні-гри представлено на рисунку 3.36.

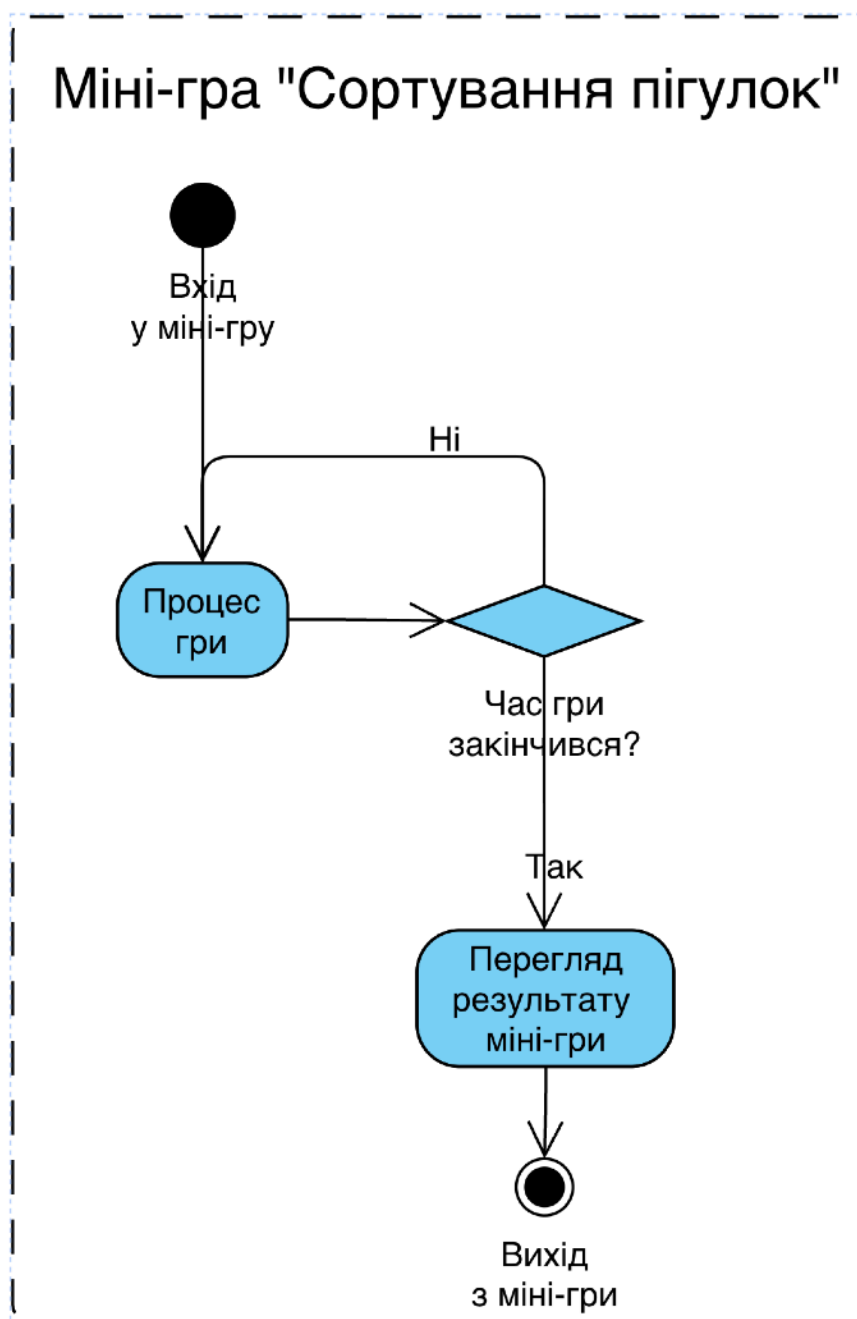


Рис 3.36 Алгоритм роботи міні-гри «Сортування пігулок»

Екранна форма процесу гри зображена на рисунку 3.37, екранна форма результатів гри представлена на рисунку 3.38.



Рис 3.37 Екрана форма міні-гри «Сортування пігулок»

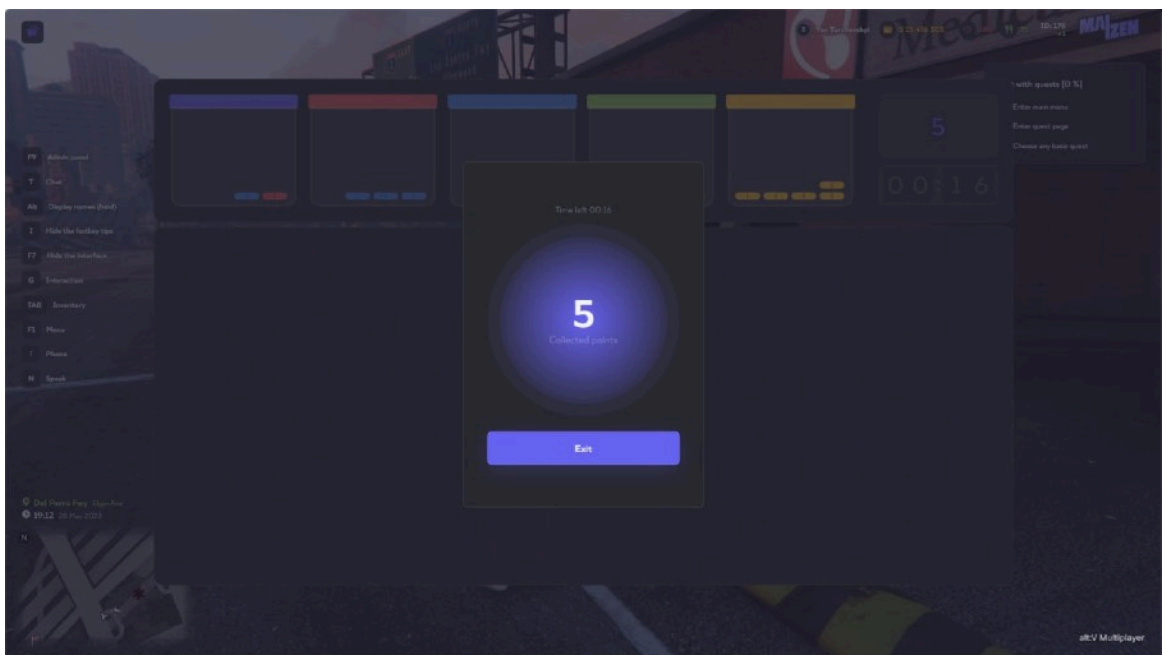


Рис 3.38 Екрана форма результату міні-гри «Сортування пігулок»

Розробка меню документів. Меню документів це меню, яке може відображати один з документів користувача. Виклик даного меню зазвичай відбувається в процесі взаємодії двох гравців. Алгоритм роботи даного меню представлений на рисунку 3.39.

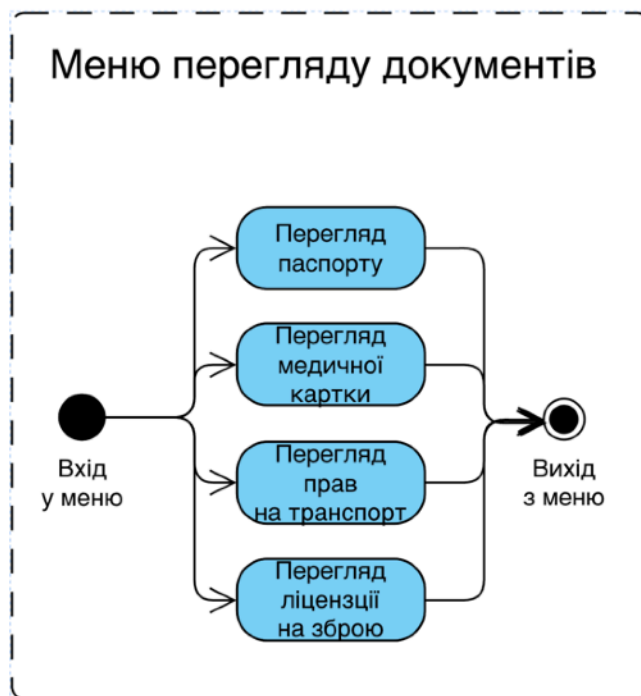


Рис 3.39 Алгоритм меню документів

Екранні форми всіх існуючих документів наведені на рисунку 3.40



Рис 3.40 Екранні форми всіх видів документів

Розробка системи конструювання одягу. Дана система є унікальною, вона розроблена для більш легкого створення наборів, комплектів та об'єктів одягу. Система включає в себе систему рендеру, налаштувань потрібних для гри параметрів та можливість використання галереї створених об'єктів. Алгоритм даної системи представлено на рисунку 3.41.

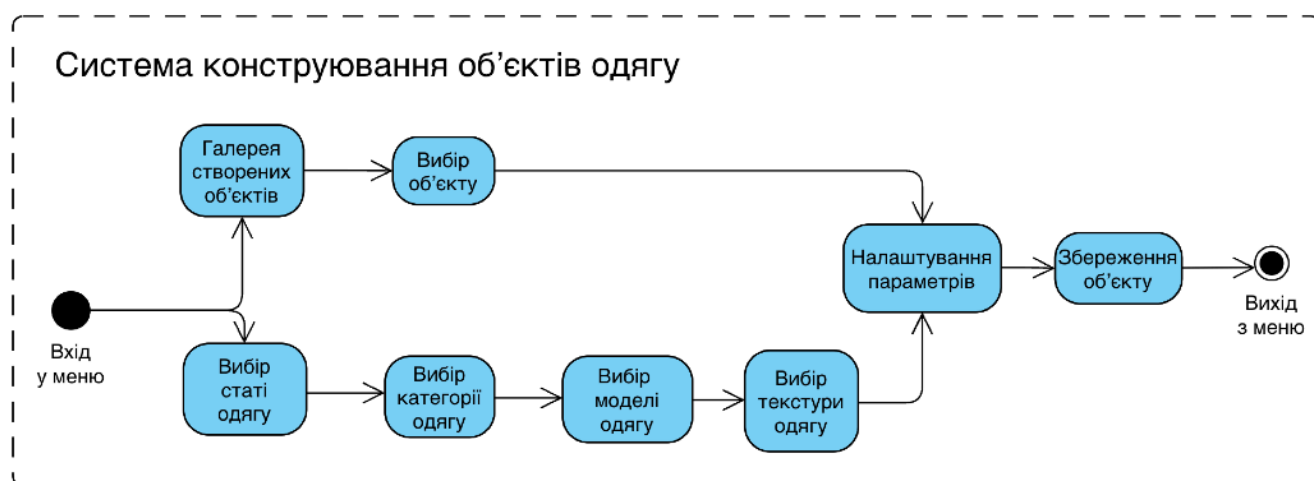


Рис 3.41 Алгоритм системи конструювання одягу

Екранна форма меню вибору категорії одягу представлено на рисунку 3.42, екранна форма процесу конструювання представлена на рисунку 3.43.

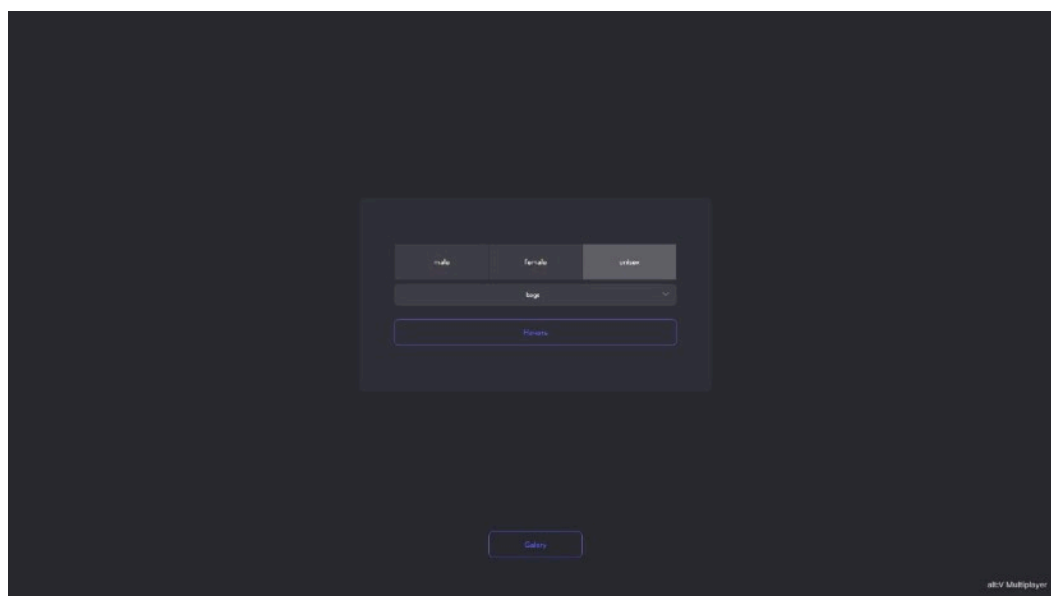


Рис 3.42 Екранна форма меню вибору категорій



Рис 3.43 Екранна форма налаштування вибраної текстури

Розробка головного меню гравця. Головне меню це головний пункт гравця для перегляду статистики, покупки різних предметів та транспорту, вибору квестів, та звернення по допомогу до адміністрації. Алгоритм роботи меню представлено на рисунку 3.44.

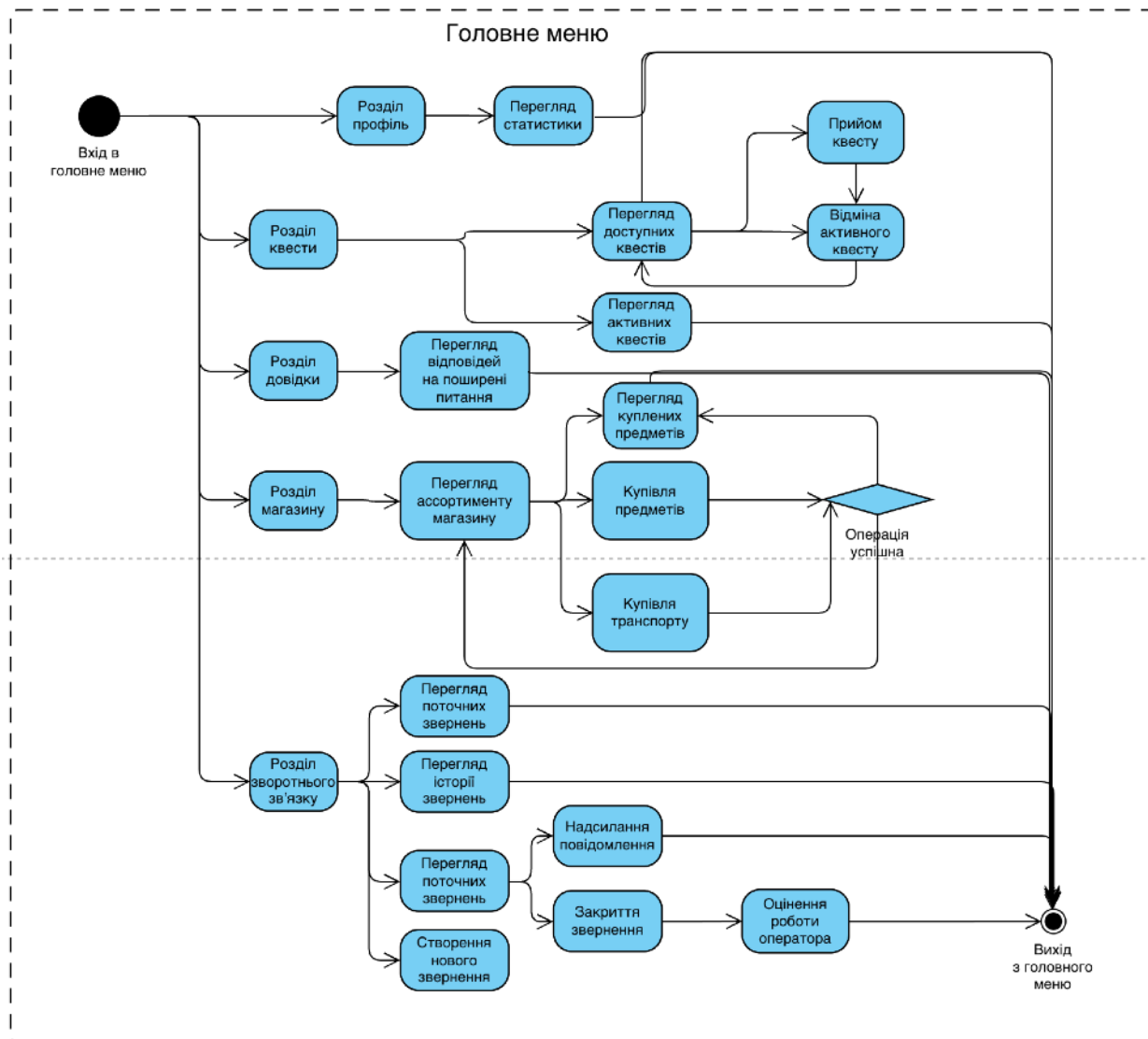


Рис 3.44 Алгоритм головного меню

Екранна форма головного розділу головного меню представлено на рисунку 3.45.

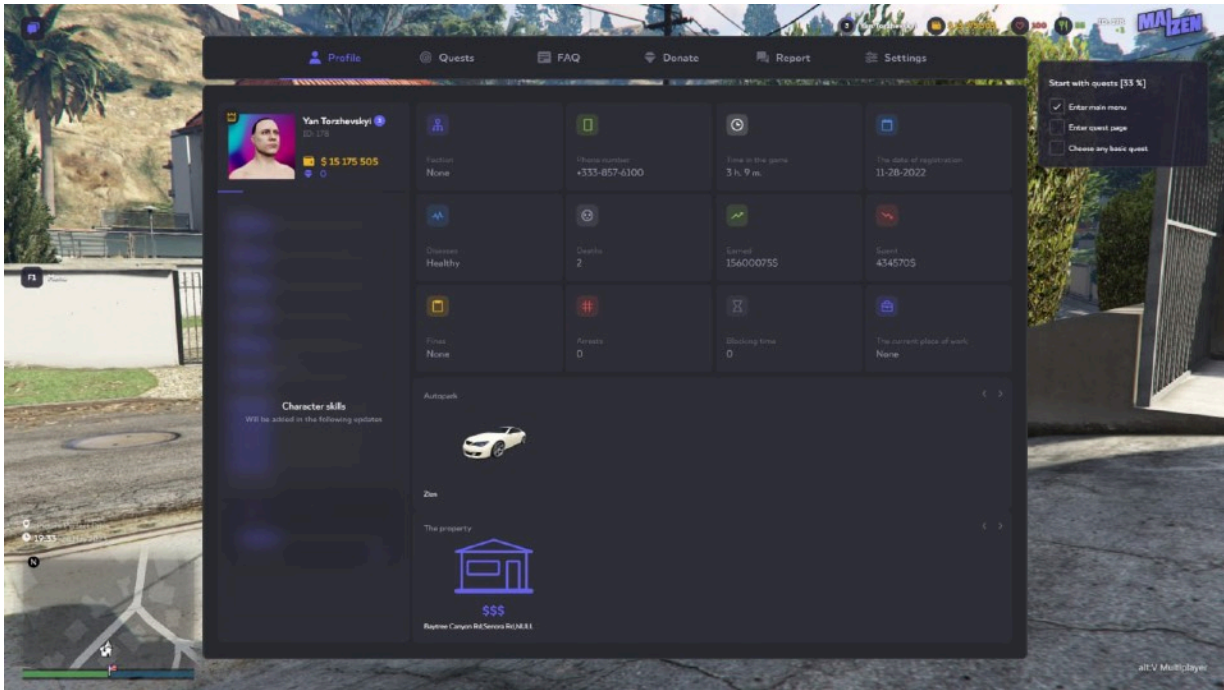


Рис 3.45 Экранна форма статистики гравця

Розробка інвентарю. Інвентар є необхідним компонентом для будь-якої рольової гри. З його допомогою користувач, може переносити, використовувати та ділитися предметами з іншими гравцями. Алгоритм роботи представлено на рисунку 3.46.

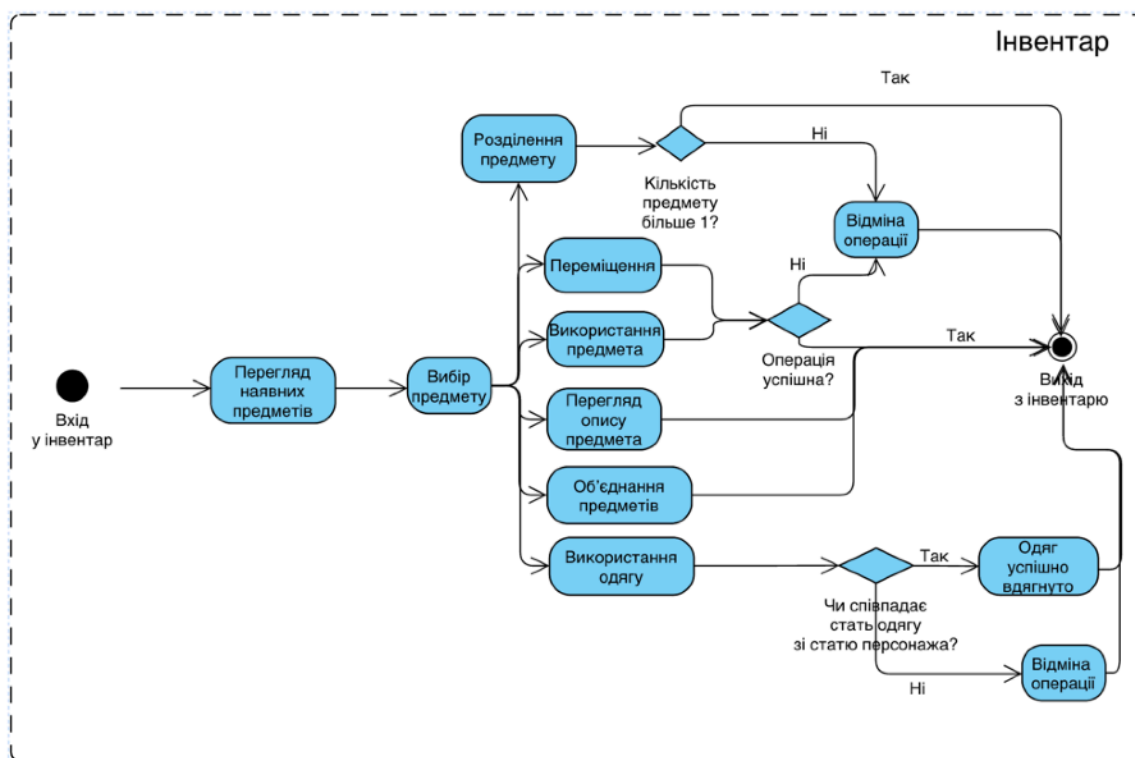


Рис 3.46 Алгоритм інвентарю

Екранна форма системи наведена на рисунку 3.47.



Рис 3.47 Екранна форма інвентарю

Розробка головного інтерфейсу. Головний інтерфейс є основним екраном користувача. Він був створений як SPA (single-page-application) для максимальної плавності переходів між різними меню головного інтерфейсу. За допомогою даного екран користувач може дізнатись показники його персонажу, баланс, отримати інформацію про час, дату та місцезнаходження, використовувати чат, переглядати поточні квести, активні для використання клавіши. Схема роботи меню представлена на рисунку 3.48.

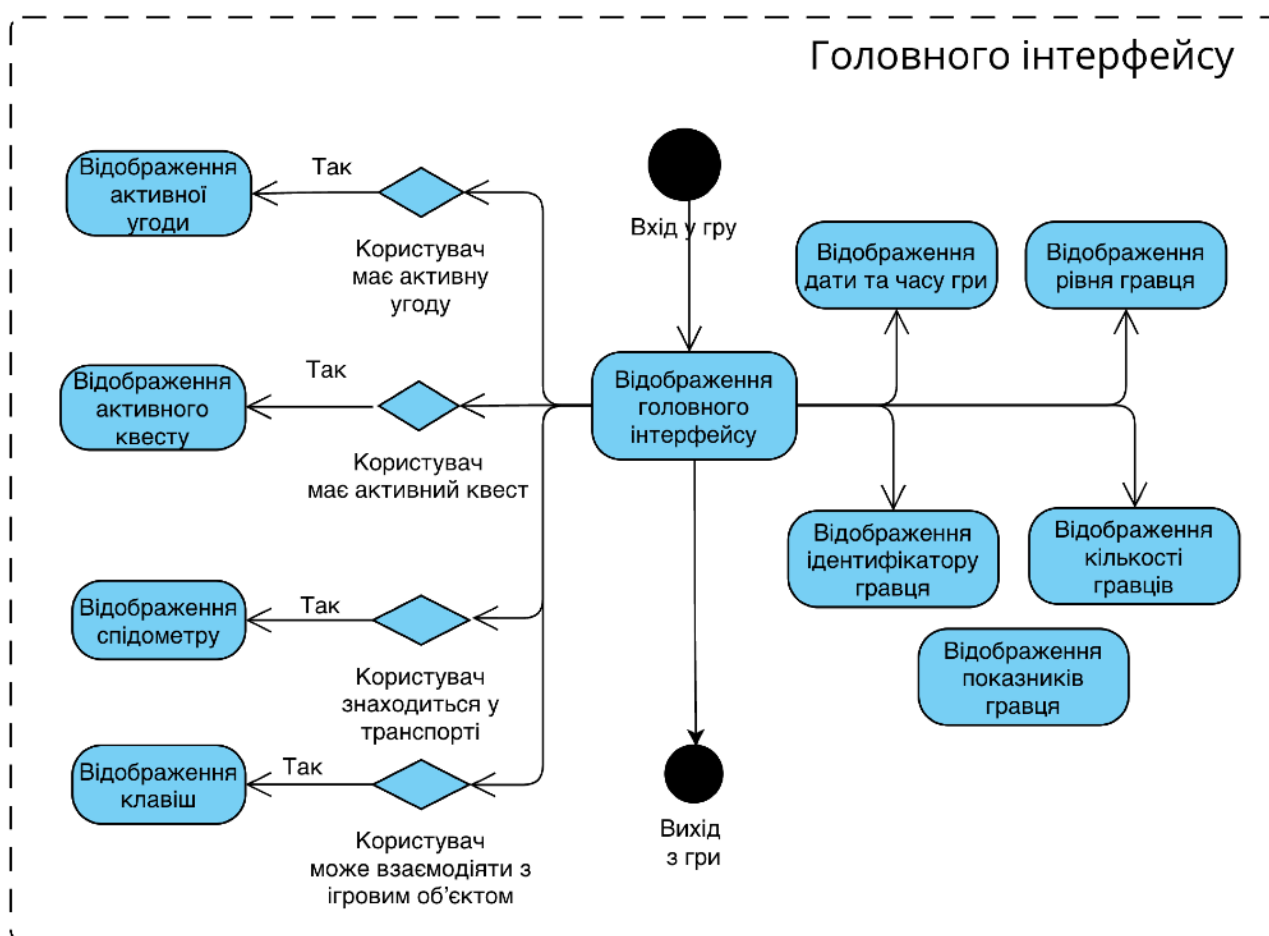


Рис 3.48 Схема роботи головного інтерфейсу



Рис 3.49 Екранна форма головного інтерфейсу. (Згорнутий чат, час та місце знаходження персонажу, його показники, та поточний квест)



Рис 3.50 Екранна форма головного інтерфейсу.
(Спідометр)

4. РОЗРОБКА СИСТЕМ ВЗАЄМОДІЇ ТА АЛГОРИТМІВ РОБОТИ

4.1 Розробка інтерфейсів взаємодії

Розробка чату. Чат дозволяє користувачам спілкуватися у текстовому форматі. Це може бути зручним для користувачів, які не можуть використовувати голосовий чат, або для обговорення не ігрових тем та технічної допомоги. Алгоритм роботи чату зображено на рисунку 4.1.

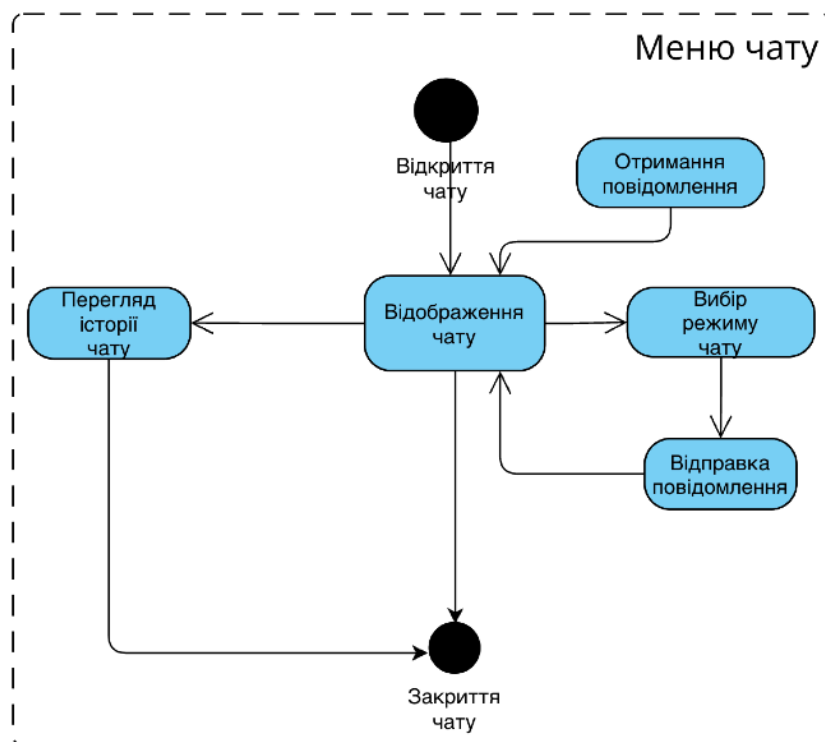


Рис 4.1 Алгоритм роботи чату

Екранна форма розробленого меню представлена на рисунку 4.2.



Рис 4.2 Екранна форма чату

Розробка телефону. Телефон - це багатофункціональна система, яка окрім класичних для телефону систем таких як дзвінки, месенджер та книга контактів, дозволяє гравцю використовувати додатки, частин з яких є опціональною для певних ролей.

Телефон дозволяє реалізувати меню максимально органічно, не порушуючи ігрову логіку та рольову атмосферу. Алгоритм роботи інтерфейсу телефону зображено на рисунку 4.3.

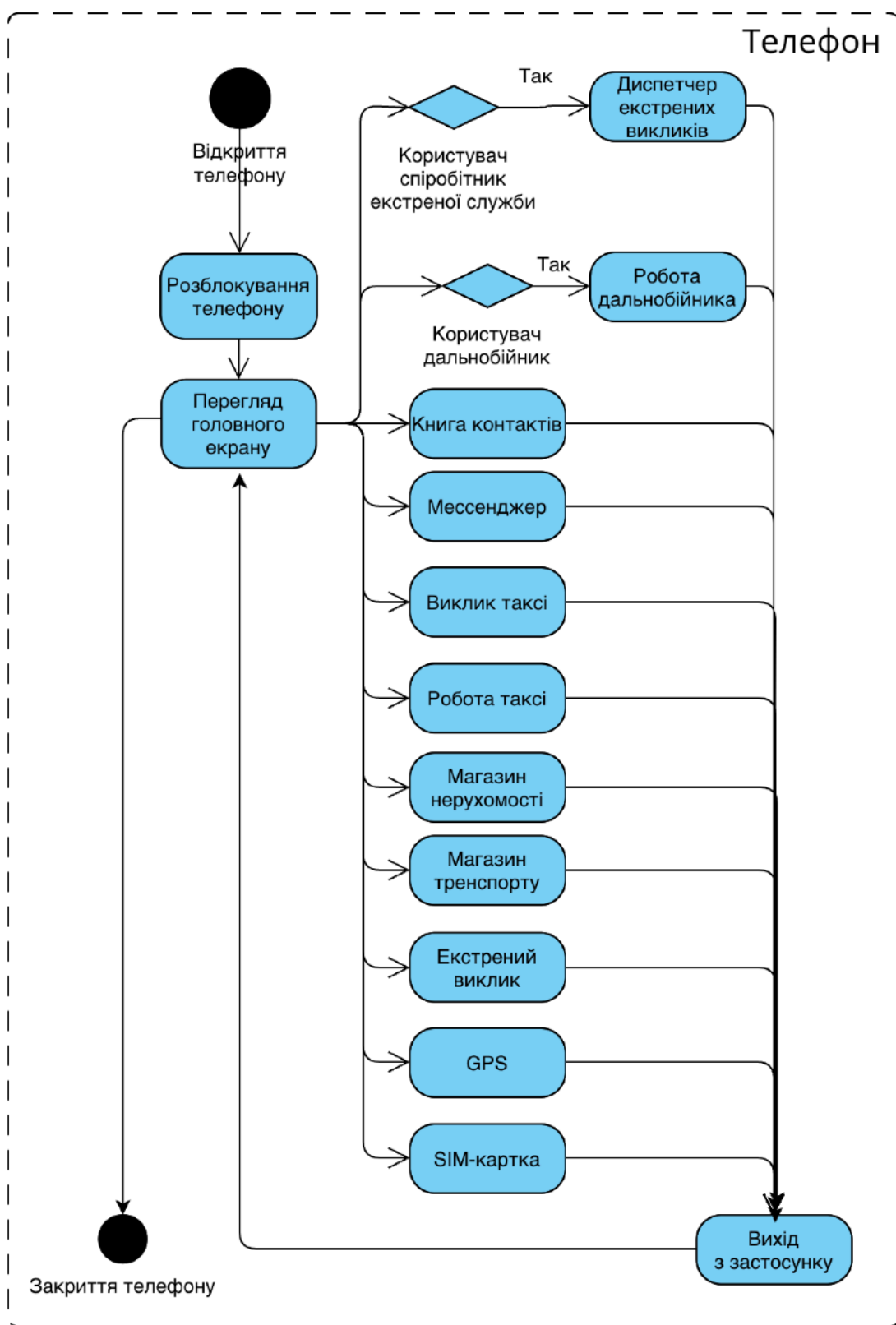


Рис 4.3 Алгоритм роботи системи телефону

Екранні форми розробленого заблокованого та розблокованого головного меню представлені на рисунку 4.4, екранні форми застосунку телефонної книги та набору номеру представлено на рисунку 4.5, екранні форми комунікаційної системи телефону представлено на рисунку 4.6, екранні форми застосунку «Мессенджер» представлено на рисунку 4.47, екранні форми застосунку «Виклик таксі» представлено на рисунку 4.8.



Рис 4.4 Екранні форми заблокованого та розблокованого екранів

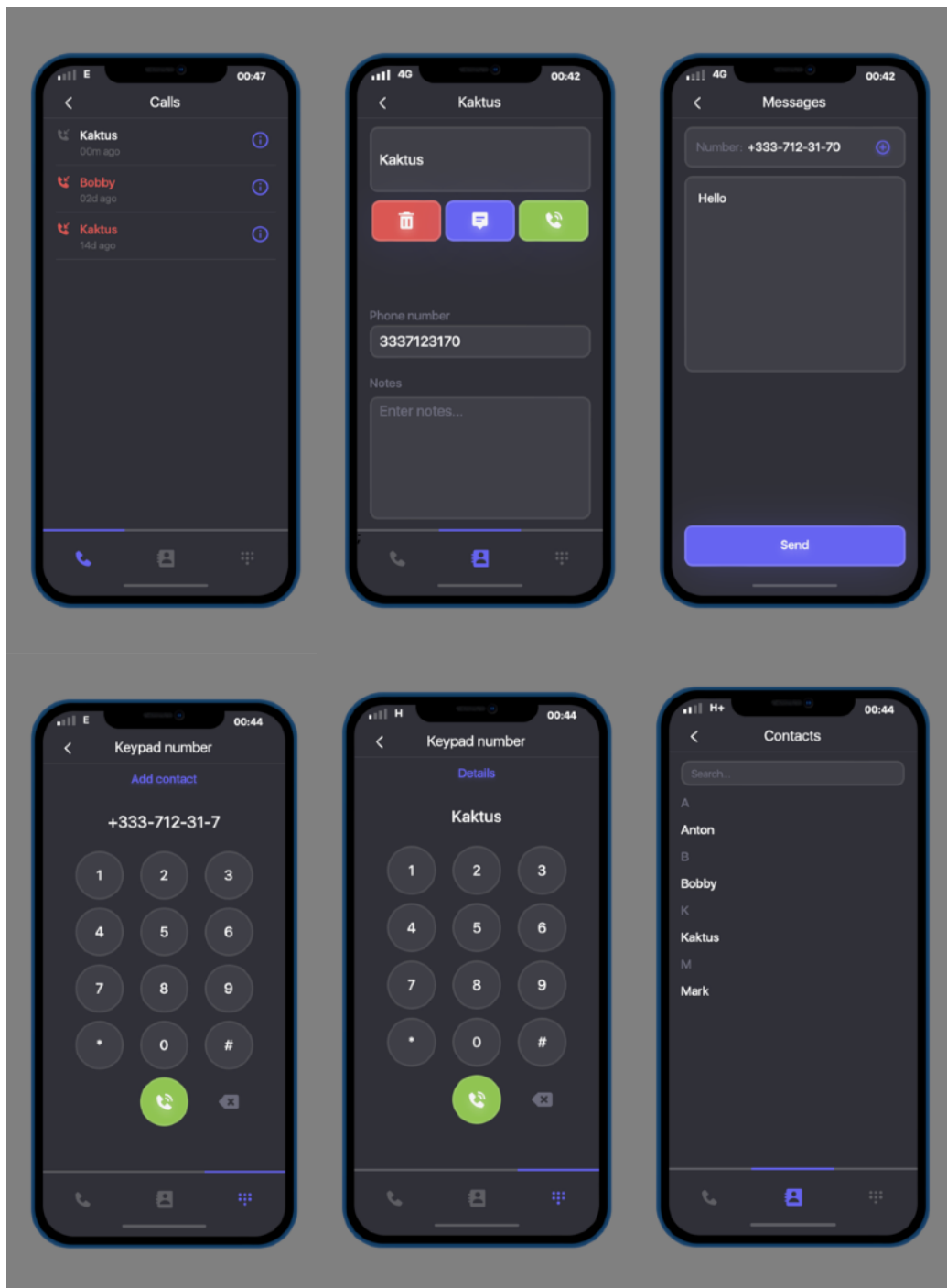


Рис 4.5 Екранні форми застосунку «Книга контактів»



Рис 4.6 Екранні форми комунікаційної системи телефону

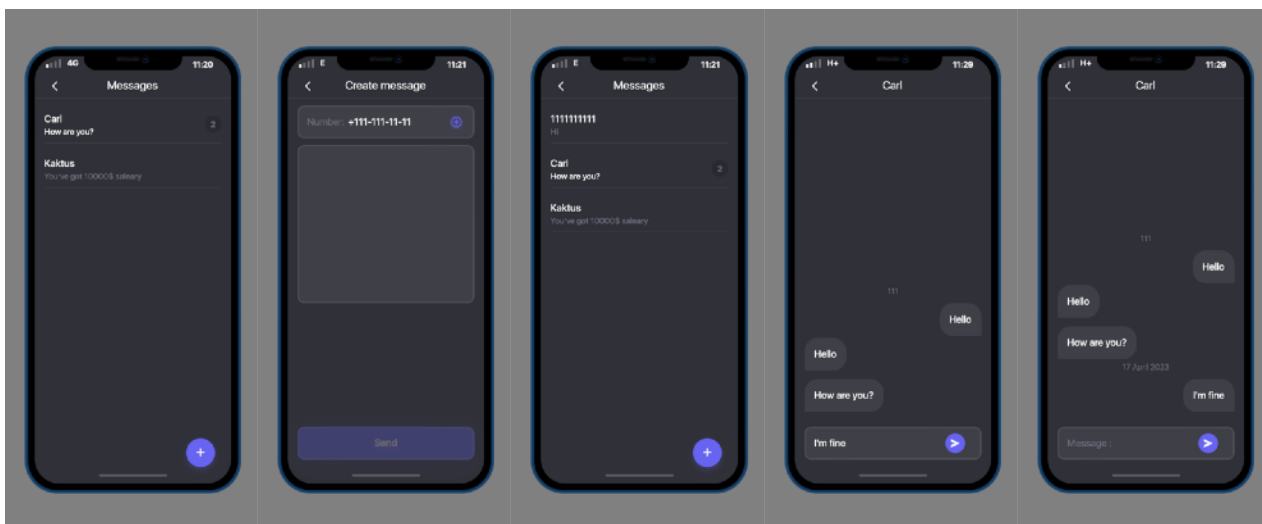


Рис 4.7 Екранні форми застосунку «Мессенджер»

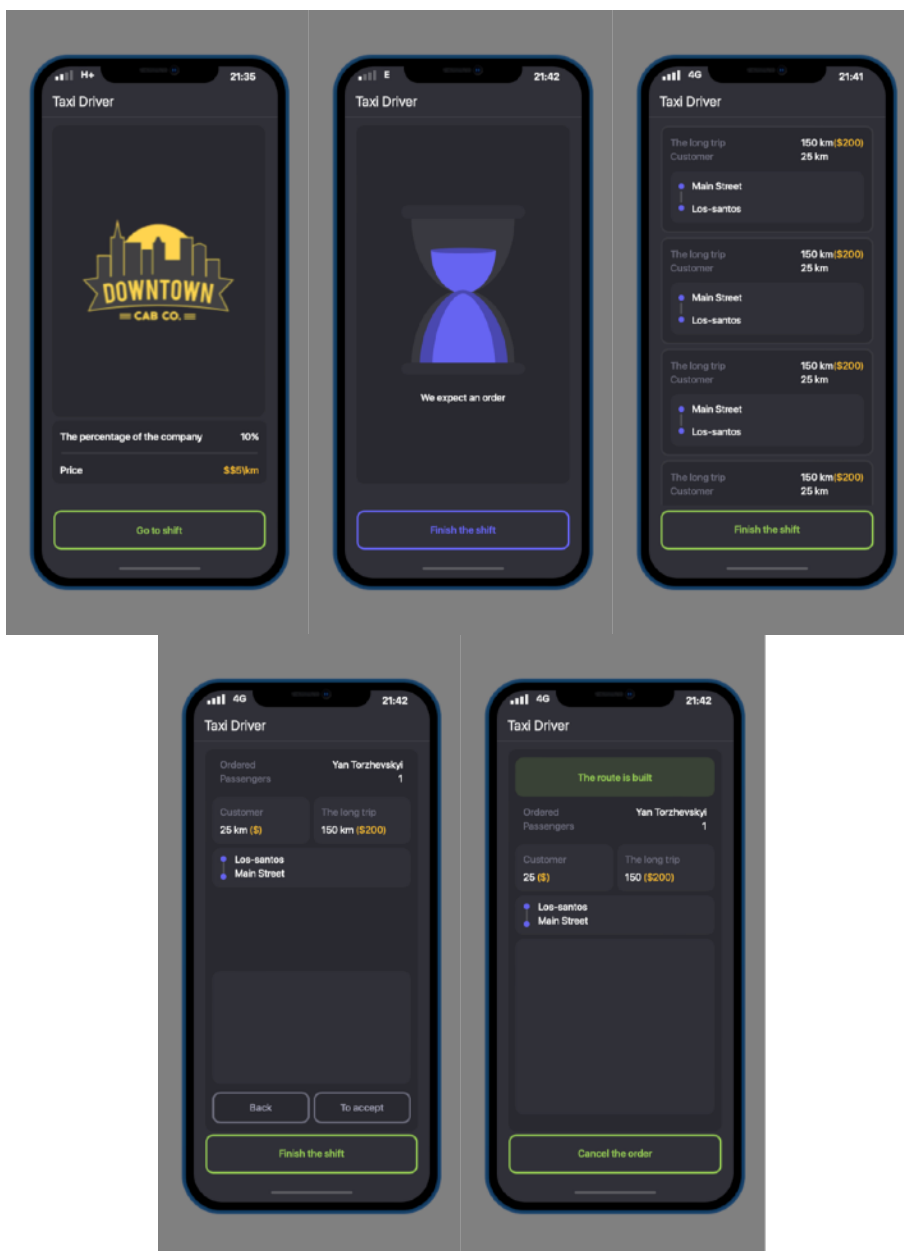


Рис 4.8 Екранні форми застосунку «Робота таксиста»

Розробка меню взаємодії. Меню взаємодії є доповненням до інших систем взаємодії користувачів. Дана система дозволяє гравцю органічним чином взаємодіяти з транспортом, нерухомістю та іншими користувачами. Меню складається з опціональних блоків, які обираються в залежності від цілі взаємодії. Алгоритм роботи меню представлено на рисунку 4.9.

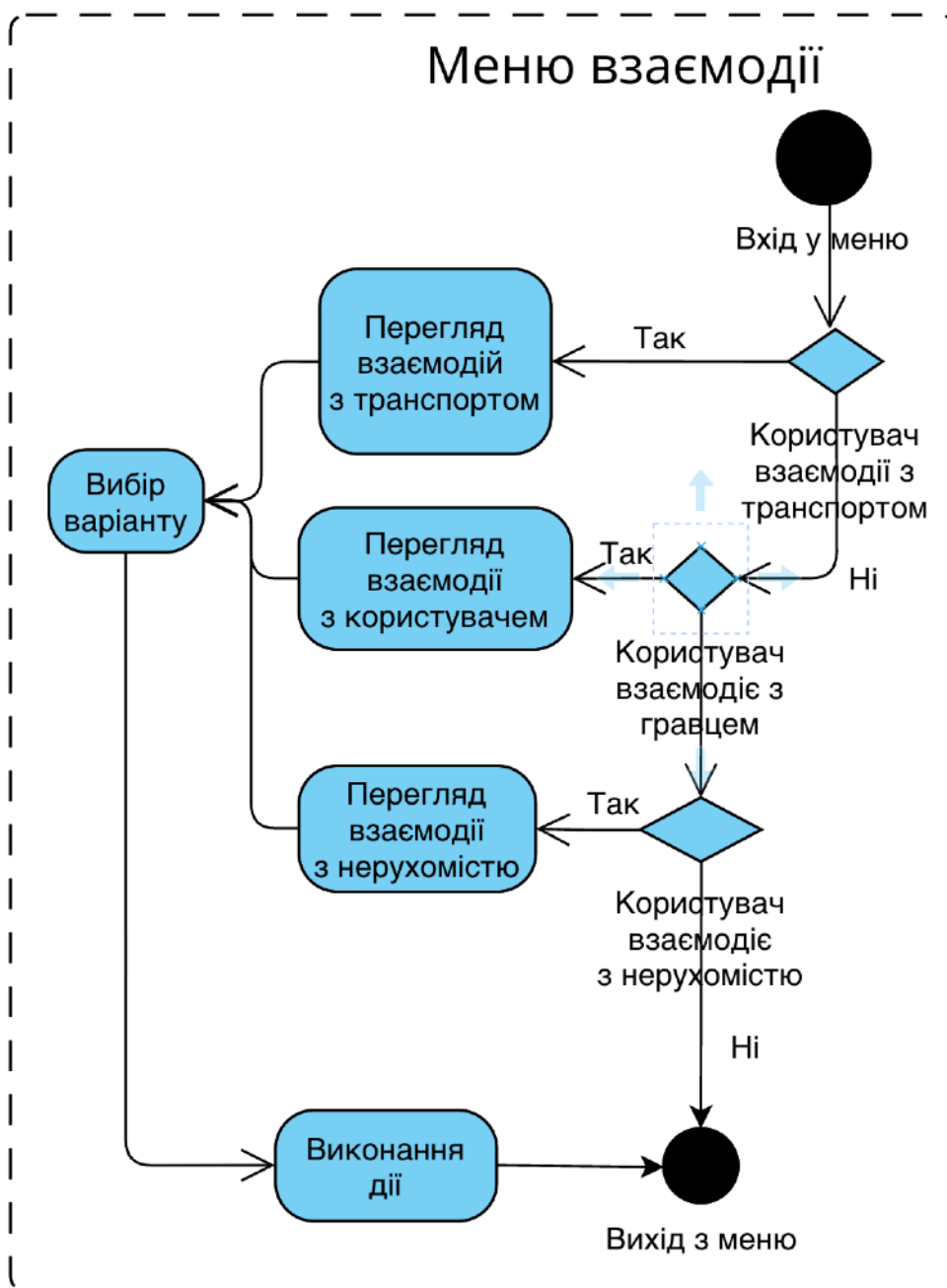


Рис 4.9 Алгоритм роботи меню взаємодії

Екранна форма системи представлена на рисунку 4.10.



Рис 4.10 Екранна форма взаємодії з транспортом

Розробка системи діалогів. Система діалогів допомагає отримувати необхідну інформацію не порушуючи атмосферу рольової гри на сервері. Система діалогів може мати безліч блоків, переходів між ними, та виконувати певні дії. Алгоритм роботи системи діалогів зображено на рисунку 3.8.

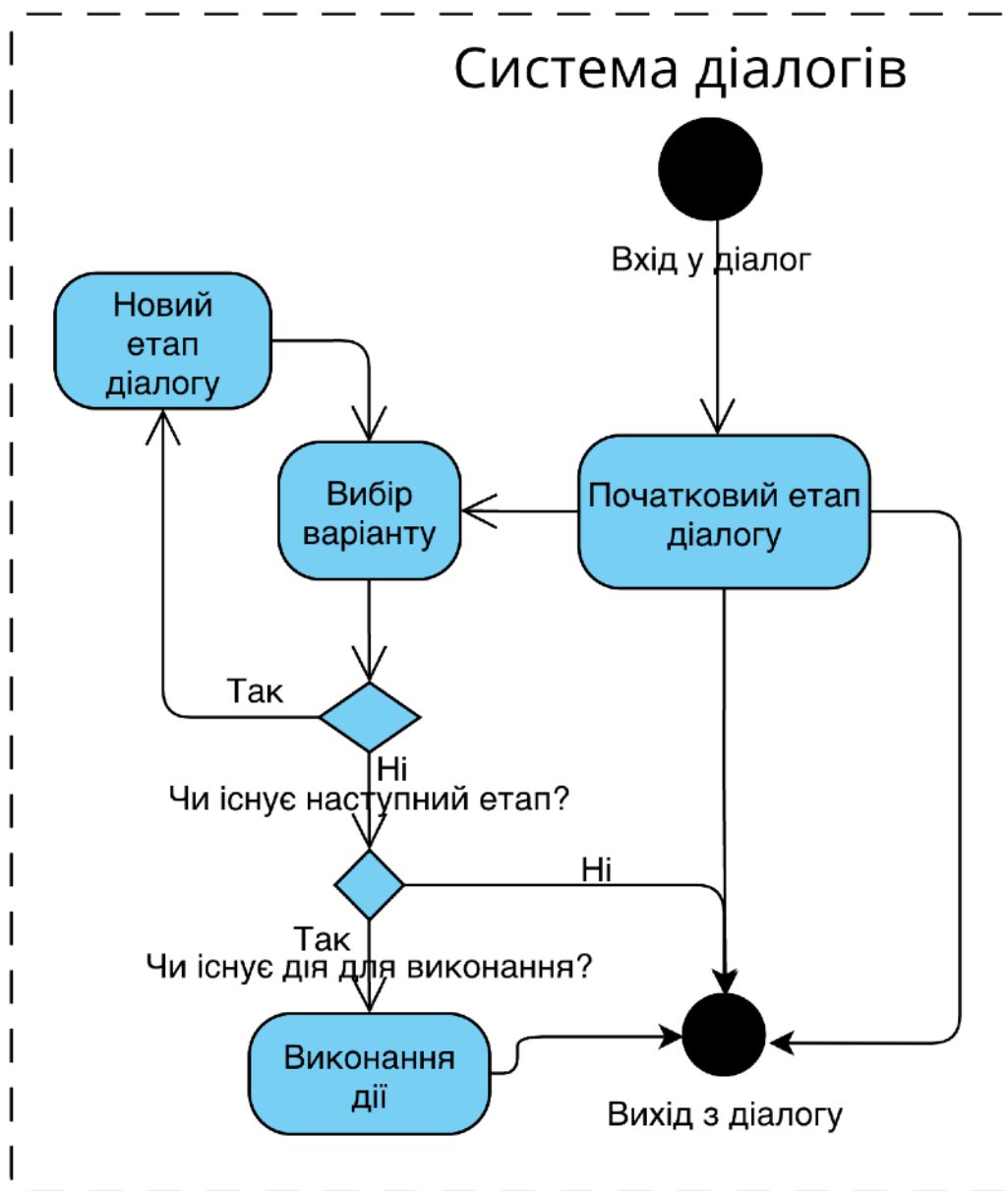


Рис 3.8 Алгоритм роботи системи діалогів

Екранна форма розробленого меню представлена на рисунку 3.9.

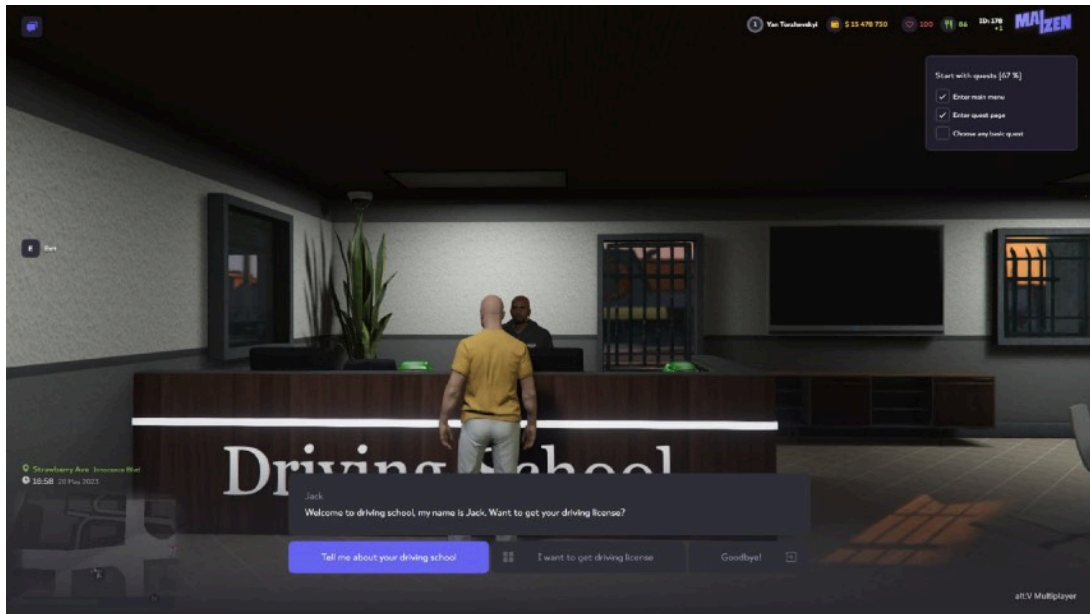


Рис 3.9 Екранна форма системи діалогів

ВИСНОВКИ

1. В ході аналізу ігрової індустрії було визначено, що головним вектором розвитку сучасних ігор є використання рольового жанру як основного, або наявність у грі великої кількості рольових механік.
2. В результаті дослідження ігрової індустрії було визначено що, рольові сервери на базі гри GTAV набувають неймовірної популярності та користуються попитом.
3. Проведено аналіз вимог користувачів до комп'ютерних рольових ігор, в результаті дослідження було визначено, що користувачі зацікавлені у наявності глибокої персоналізації, наявності унікальних систем та механік, а також адаптивного та швидкого користувацького інтерфейсу.
4. Обрано технології та платформи для розробки. Для створення макетів інтерфейсу було використано платформу Figma. Для розробки користувацької frontend-системи було використано мову програмування JavaScript, та фреймворк React.js. В якості платформи для розміщення було обрано рушій AltV, що забезпечує роботу frontend-системи з грою GTAV.
5. Визначено функціональні вимоги до багатокористувацької рольової гри. Серед функціональних можливостей є такі:
 - Реєстрація, та авторизація користувача.
 - Створення, та управління створеними персонажами.
 - Використання магазинів одягу, їжі, зброї, транспорту.
 - Використання телефону, та застосунків.
 - Створення, та управління компанією.
 - Придбання, продажу, та управління нерухомістю.
 - Отримання основної ігрової інформації через головний інтерфейс.
 - Взаємодія з іншими гравцями через меню взаємодії.
 - Використання ігрового чату.
 - Використання заправної станції.
 - Отримання, прийняття чи відхилення угоди.

- Використання меню для налаштувань гри, придбання ігрових предметів, та отримання статистики гравця.
- Використання панелі управління нерухомістю.
- Використання системи конструювання об'єктів одягу.
- Використання панелі відповіді на зворотний зв'язок користувачів.

6. Визначено нефункціональні вимоги до багатокористувацької рольової гри. Серед нефункціональних можливостей є такі:

Продуктивність: Frontend-система має забезпечувати високу продуктивність з мінімальним часом відгуку, оптимізацією використання ресурсів та ефективним завантаженням. Використання React.js забезпечує високу продуктивність за рахунок віртуального DOM і "lazy loading". Метрика: час відгуку менше 200 мс.

Сумісність: Frontend-система має бути сумісною з AltV і GTA V, забезпечуючи гладке використання коду та візуальну гармонію з дизайном GTA V. Сумісність досягнута за допомогою тісної співпраці з командою розробників AltV і використання Figma для проектування UI. Метрика: 100% функціоналу працює без помилок.

Надійність: Frontend-система має бути стабільною і відповідати на дії користувача без помилок та затримок. Досягається через детальне тестування і безпечне кодування. Метрика: 99,9% часу без збоїв.

Безпека: Всі дані користувачів мають бути захищені за допомогою сучасних методів шифрування. Час виявлення та виправлення хиб в безпеці не повинен перевищувати 24 години.

Масштабованість: Інтерфейс має бути готовим до масштабування, зберігаючи свою продуктивність та стабільність при зростанні числа користувачів. Метрика: здатність підтримувати стабільну роботу при збільшенні кількості користувачів, значення навантаження серверу не повинна перевищувати 30%.

Підтримка: Frontend-система повинна бути легко підтримуваним, включаючи простоту внесення змін, додавання нового функціоналу, виправлення помилок і оновлень. Метрика: час потрібний для виправлення помилок менше однієї доби.

Юзабіліті: Інтерфейс має бути інтуїтивно зрозумілим для користувачів, дозволяючи їм легко взаємодіяти з грою. Метрика: 95% користувачів відзначають високий рівень зручності використання за допомогою анкетування.

7. Розроблено frontend-систему багатокористувацької рольової гри згідно визначених вимог.
8. Проведено тестування роботи frontend-системи, та виправлення знайдених хиб.
9. Проведено інтеграцію frontend-системи з backend-системою багатокористувацької рольової гри.

ПЕРЕЛІК ПОСИЛАНЬ

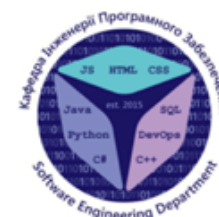
1. «The Usability of Massively Multiplayer Online Roleplaying Games: Designing for New Users» Steve Cornett [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/221518580_The_Usability_of_Massively_Multiplayer_Online_Roleplaying_Games_Designing_for_New_Users
2. «Designing Interfaces: Patterns for Effective Interaction Design» Jenifer Tidwell, Charles Brewer and Aynne Valencia [Електронний ресурс] – Режим доступу: https://www.amazon.com/Designing-Interfaces-Patterns-Effective-Interaction/dp/1492051969/ref=sr_1_1?dchild=1&keywords=Jenifer+Tidwell,+Charles+Brewer+and+Aynne+Valencia&qid=1622014247&sr=8-1
3. Платформа статистики та аналітики стримінгової платформи Twitch [Електронний ресурс] – Режим доступу: <https://streamscharts.com/games/grand-theft-auto-v>
4. Платформа AltV [Електронний ресурс] – Режим доступу: <https://altv.mp/#/>
5. FiveM [Електронний ресурс] – Режим доступу: <https://docs.fivem.net/docs/>
6. RageMP [Електронний ресурс] – Режим доступу: https://wiki.rage.mp/index.php?title=Main_Page
7. React.js [Електронний ресурс] – Режим доступу: <https://uk.legacy.reactjs.org/>
8. Figma [Електронний ресурс] – Режим доступу: <https://help.figma.com/hc/en-us/sections/13148565160471-Figma-design-basics>
9. Документація роботи WebView платформи AltV [Електронний ресурс] – Режим доступу: <https://docs.altv.mp/js/api/alt-client.WebView.html>
10. NoPixel [Електронний ресурс] – Режим доступу: <https://www.nopixel.net/upload/index.php>

ДОДАТОК А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
 ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка frontend-частини багатокористувацької комп'ютерної гри GTA V з використанням рушія AltV та фреймворку React JS

Виконав студент 4 курсу
 групи ПД-43
 Торжєвський Ян Русланович

Керівник роботи
 Щербина Ірина Сергіївна
 доцент,
 доцент кафедри інженерії програмного забезпечення,
 кандидат технічних наук

Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи - покращення взаємодії користувача з сервером багатокористувацької рольової гри засобами frontend компонентів на платформі AltV з використанням фреймворку React.js.

Об'єкт дослідження - геймплей у багатокористувацькій комп'ютерній рольовій грі.

Предмет дослідження - frontend-частина багатокористувацької комп'ютерної гри на базі GTA V з використанням рушія AltV.

ЗАВДАННЯ БАКАЛАВРСЬКОЇ РОБОТИ

Передо мною була поставлена задача створення та проектування frontend-системи багатокористувацької рольової гри на основі GTA V. Для цього було виконано:

1. Аналіз сучасної ігрової індустрії.
2. Розроблено функціональні та нефункціональні вимоги.
3. Розроблено алгоритми роботи взаємодії та інтерфейсів користувача.
4. Розроблено frontend-систему багатокористувацької рольової гри.

3

АНАЛІЗ FRONTEND-СИСТЕМ ІГОР КОНКУРЕНТІВ

Назва пункту	Eclipse RP	Echo RP	NoPixel	Maizen RP
Масштабованість	+	-	+	+
Адаптивний дизайн	+	+	-	+
Використання WebSocket для передачі даних	-	+	-	+
Підтримка «безшовного» оновлення даних	-	+	-	+
Оптимізація шляхом використання «lazy loading»	-	+	-	+
Резервне відображення для мультимедіа	-	-	-	+
Використання CSS-анимацій	+	+	-	+

4

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



JavaScript



Figma



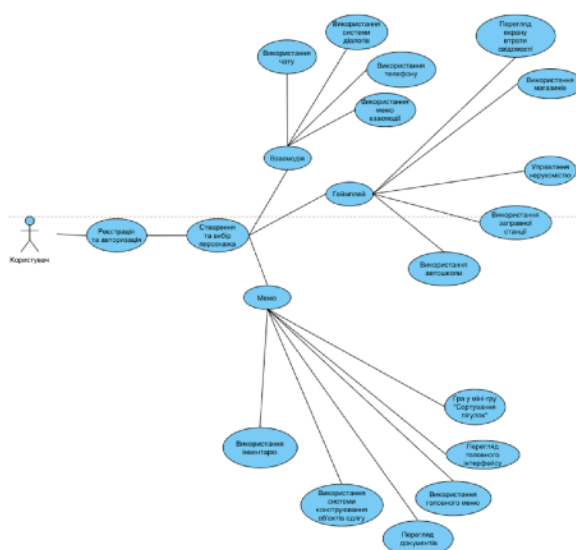
React JS



AltV

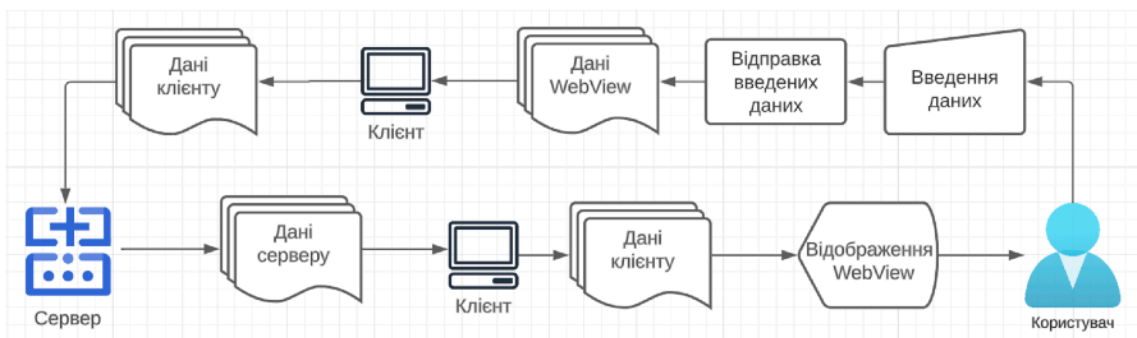
5

УЗАГАЛЬНЕНА ДІАГРАМА ПРЕЦЕДЕНТІВ



6

ДІАГРАМА ВЗАЄМОДІЇ КОРИСТУВАЧА З СЕРВЕРОМ



АЛГОРИТМ ТА ЕКРАННА ФОРМА СИСТЕМИ ІНВЕНТАРЮ

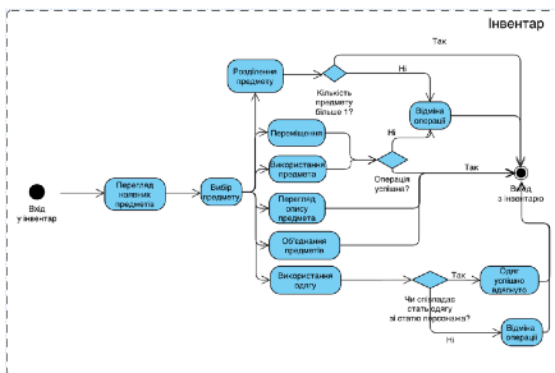


Схема роботи системи інвентарю



Екранна форма системи інвентарю

АЛГОРИТМ ТА ЕКРАННА ФОРМА СИСТЕМИ ДІАЛОГІВ

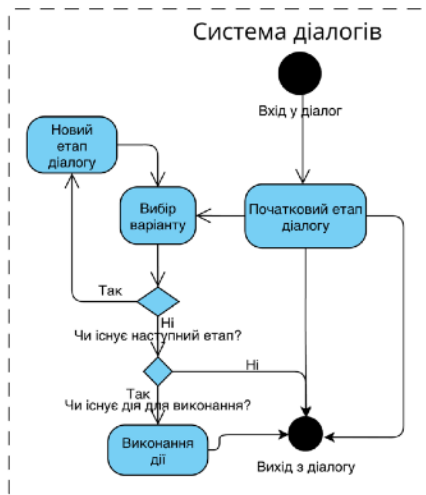
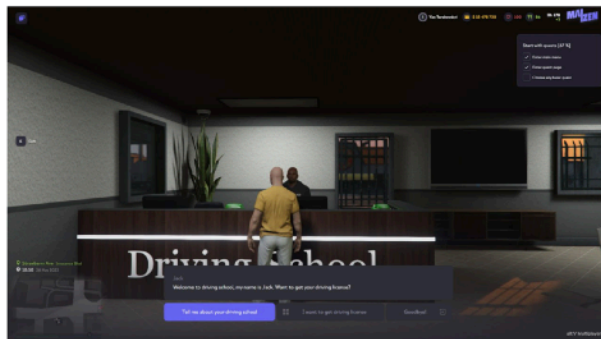


Схема роботи системи діалогів



Екранна форма системи діалогів

9

АЛГОРИТМ ТА ЕКРАННА ФОРМА ГОЛОВНОГО ІНТЕРФЕЙСУ

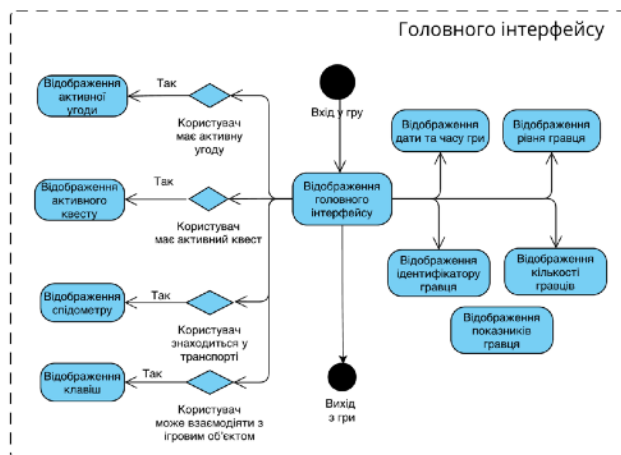


Схема роботи головного інтерфейсу



Екранна форма головного інтерфейсу

10

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1) Торжевський Я. Р. Аналіз впливу метавсесвітів та рольових режимів на розвиток інфокомунікаційних технологій / Торжевський Я. Р. , Щербина І. С. // Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ». Збірник тез 20.04.2023 ДУТ, м. Київ — К.: ДУТ, 2023. — С. 115.

2) Торжевський Я. Р. Аналіз використання метавсесвітів та рольових режимів у сфері освіти / Торжевський Я. Р. , Щербина І. С. // III Всеукраїнська Науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті». Збірник тез 16.05.2023 ДУТ, м. Київ — К.: ДУТ, 2023. Подано до друку.

ВИСНОВКИ

1. Проведено аналіз сучасної індустрії відеоігор.
2. Проведено аналіз та дослідження ігор в рольовому жанрі.
3. Обрано інструменти для розробки frontend-у рольової гри.
4. Визначено функціональні вимоги до рольової гри.
5. Визначено нефункціональні вимоги до рольової гри.
6. Розроблено frontend-систему для взаємодії користувача з сервером рольової гри згідно до вимог.
7. Проведено тестування роботи frontend-компонентів розробленої системи інтерфейсу користувача.
8. Проведено інтеграцію frontend-у з backend-ом рольової гри.

ДЯКУЮ ЗА УВАГУ!

ДОДАТОК Б