

Факультет комп'ютерно-інформаційних систем та програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

на тему: Розробка мобільного застосунку для дій у надзвичайних ситуаціях

Виконав(ла): студент(ка) 4 курсу, групи СПс-41
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Бодя Р. І.

(підпис)

(прізвище та ініціали)

Керівник

Михалик Д.М.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Стоянов Ю. М.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Петрик М. Р.

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота бакалавра на тему «Розробка мобільного застосунку для дій у надзвичайних ситуаціях» розроблена та написана Бодя Русланом Івановичем, студентом Тернопільського національного технічного університету Факультету комп'ютерно-інформаційних систем та програмної інженерії, кафедра програмної інженерії, група СПс-41.

Обсяг роботи складає: сторінок – 62 , рисунків – 38 , додатків – 2 , посилань – 9.

Метою кваліфікаційної роботи є створення мобільного застосунку, який надаватиме користувачам необхідну інформацію та рекомендації щодо дій у різних надзвичайних ситуаціях. Цей застосунок розроблено з урахуванням сучасних вимог до інтерфейсу користувача та функціональних можливостей. Основні функції застосунку включають перегляд інформації про реагування на надзвичайні ситуації, доступ до карт тривоги, ілюстрації дій та інтерактивні тести для перевірки знань.

В рамках роботи проведено аналіз предметної області та інформаційного забезпечення, обрано відповідні засоби та методи вирішення проблеми. Для реалізації застосунку використано ігровий движок Unity та мову програмування C#. Особлива увага приділялася створенню інтуїтивно зрозумілого інтерфейсу користувача та забезпеченню безпеки і конфіденційності даних.

У процесі розробки мобільного застосунку було впроваджено архітектурний підхід Model-View-Controller, що сприяло чіткому розподілу функцій між компонентами застосунку.

Ця робота підтвердила доцільність використання сучасних інструментів та технологій для створення ефективних і зручних інструментів, які допомагають користувачам орієнтуватися та діяти у критичних ситуаціях.

ABSTRACT

Bachelor's qualification thesis on the topic: "Development of a Mobile Application for Actions in Emergency Situations" was developed and written by Bodia Ruslan Ivanovych, a student of Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, group SPs-41.

The volume of the work includes: pages – 62 , figures – 38 , appendices – 2, references – 9.

The aim of the project is to create a mobile application that will provide users with necessary information and recommendations on actions to take in various emergency situations. This application has been developed considering modern requirements for user interface and functionality. The main features of the application include viewing information on emergency response, accessing alert maps, action illustrations, and interactive tests to verify knowledge.

Within the framework of the work, an analysis of the subject area and information support was conducted, appropriate means and methods for solving the problem were selected. The Unity game engine and the C# programming language were used to implement the application. Special attention was paid to creating an intuitive user interface and ensuring the security and confidentiality of user data.

During the development process of the mobile application, the Model-View-Controller (MVC) architectural approach was implemented, which contributed to a clear division of functions among the application's components.

The project has confirmed the feasibility of using modern tools and technologies to create effective and user-friendly tools that help users navigate and act in critical situations.

ЗМІСТ

АНОТАЦІЯ.....	4
ABSTRACT.....	5
ВСТУП.....	8
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ ТА ТЕХНІЧНЕ ЗАВДАННЯ.....	10
1.1 Дослідження предметної області.....	10
1.2 Аналіз інформаційного забезпечення предметної області.....	13
1.3 Засоби та методи вирішення проблеми.....	14
2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СТРУКТУРИ ПРОГРАМНОГО ПРОДУКТУ.....	16
2.1 Розробка архітектури мобільного застосунку.....	16
2.2 Аналіз інструментів для розробки	17
2.3 Середовище проектування Draw іо.....	18
2.4 Розробка діаграми прецедентів.....	19
2.5 Розробка діаграми діяльності	20
2.6 Розробка діаграма класів.....	21
2.7 Розробка діаграма послідовності	23
2.8 Розробка діаграми кооперації.....	25
2.9 Створення діаграми компонентів.....	26
3. КОНСТРУЮВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ...28	
3.1 Вибір мови програмування та інструментів розробки	28
3.2 Розробка інтерфейсу та логіки застосунку.....	30
3.3 Тестування застосунку та оцінка якості.....	44
4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ.....54	
4.1 Соціально-політичні небезпеки, їхні види та характеристики.....	54
4.2 Обґрунтування вибору фарбування стін, підлоги, ферм, колон, вантажо- підйомних кранів, верстатів та іншого обладнання цеху.....	57

ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
ДОДАТКИ.....	63

ВСТУП

В умовах глобальних кліматичних змін, збільшення частоти стихійних лих, техногенних аварій і конфліктів у світі доступний ефективний і доступний інструмент для швидкого реагування та отримання високоактуальних інструкцій. Саме з цією ідеєю пропонується розробити мобільний застосунок, який слугуватиме інструментом для надання важливої інформації та рекомендацій користувачам щодо дій у різних надзвичайних ситуаціях.

Метою цієї кваліфікаційної роботи є створення зручного та зрозумілого інтерфейсу, що дасть змогу користувачеві швидко отримувати необхідну інформацію, переглядати ілюстрації та виконувати інтерактивні тести для перевірки своїх знань. Мобільний застосунок буде створено з урахуванням потреб сучасного користувача та має на меті максимально спростити процес отримання необхідної інформації в екстрених ситуаціях.

Метою дослідження є мобільні застосунки, призначені для інформування та підтримки користувачів у надзвичайних ситуаціях. Основною метою дослідження є розробка ефективного, функціонального та зручного застосунку, який надасть користувачам швидкий доступ до важливої інформації та рекомендацій у разі надзвичайних подій.

Предметом дослідження є розробка архітектури мобільного застосунку, що включає створення інтерфейсу користувача, реалізацію функціональних можливостей, таких як перегляд інформації, відкриття

карт тривоги, перегляд ілюстрацій та проходження інтерактивних тестів, а також аналіз інструментів і технологій, що використовуються для його реалізації. .

Основна увага буде зосереджена на розробці інтерфейсу, який повинен бути інтуїтивно зрозумілим і зручним для користувача. Важливим кроком у роботі стане аналіз можливих ризиків і підготовка плану їх мінімізації та управління ними. Під час розробки будуть враховані потенційні загрози безпеці та конфіденційності інформації користувача, що зберігається в застосунку.

Важливим кроком у цьому процесі буде інтеграція з різними джерелами інформації. Це надасть користувачам доступ до актуальних даних і рекомендацій, які допоможуть їм ефективно реагувати на надзвичайні ситуації та рятувати своє життя та здоров'я.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ ТА ТЕХНІЧНЕ ЗАВДАННЯ

1.1 Дослідження предметної області

Надзвичайні ситуації є невід’ємною частиною нашого життя і варіюються від стихійних лих до техногенних аварій і соціальних конфліктів. В умовах посилення глобалізації та урбанізації питання швидкого реагування на надзвичайні ситуації набуло особливої актуальності. Важливим аспектом забезпечення безпеки є своєчасне надання інформації та чітких інструкцій мешканцям щодо дій у надзвичайній ситуації.

Сучасні технології пропонують широкий спектр рішень для цих завдань, включаючи мобільні застосунки. Мобільні застосунки є ефективним засобом передачі інформації завдяки широкому розповсюдженню та миттєвому доступу до необхідних даних. З огляду на це, дослідження в цьому тематичному напрямку включають аналіз існуючих мобільних рішень та їх ефективності в надзвичайних ситуаціях.

Однією з загальних областей є розробка програм, які попереджають і повідомляють про стихійні лиха, такі як землетруси, повені та урагани. Наприклад, Earthquake Alert!, Hurricane Tracker і FloodMap надають користувачам актуальну інформацію про поточні природні події та рекомендації щодо реагування на надзвичайні ситуації. Вигляд роботи таких програм зображено на рисунку 1.1.

Вони використовують дані національних і міжнародних організацій, таких як USGS (Геологічна служба США) і NOAA (Національне управління океанічних і атмосферних досліджень), щоб забезпечити високу точність і надійність інформації.

Вигляд роботи таких програм зображено на рисунку 1.1.

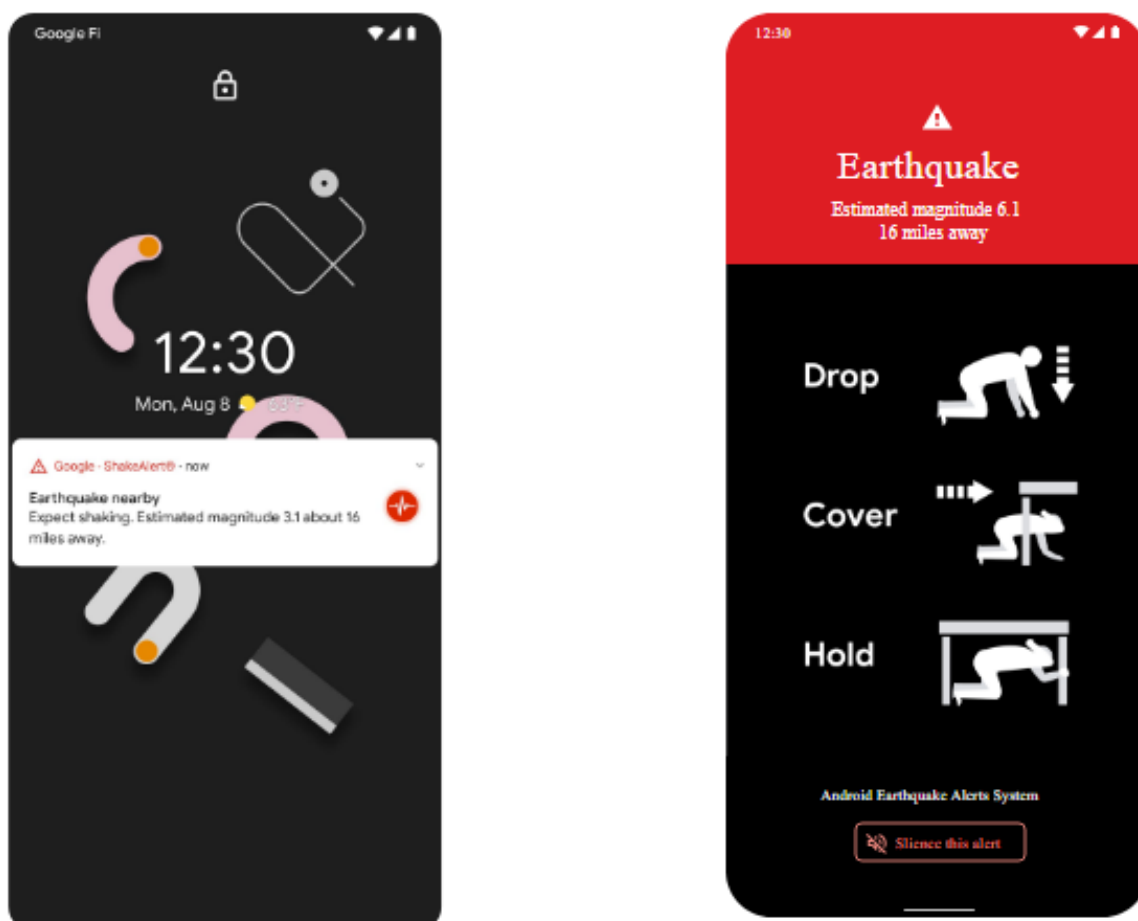


Рисунок 1.1 – Головний вигляд однієї з програм для сповіщення

Деякі застосунки зосереджені на поведінці під час нещасних випадків, спричинених людиною, та соціальних конфліктів. Надзвичайні ситуації: програма Alerts надає користувачам можливість отримувати сповіщення про різні типи надзвичайних ситуацій, як-от аварії на хімічних заводах, радіаційні загрози та терористичні атаки. Також описано методи евакуації та заходи першої допомоги.

Одним таким застосунком є Air Alerts, який використовується мільйонами людей в Україні. Застосунок одразу сповіщає про початок і закінчення тривоги. Даний застосунок доступний на усіх платформах включаючи Android, IOS та браузер версію. Для користування не потрібно ніяких реєстрацій і сам застосунок не збирає даних, що може свідчити про безпеку [1].

Головною перевагою використання мобільної версії є наявність сирени під час тривоги.

Вигляд застосунку зображено на рисунку 1.2.

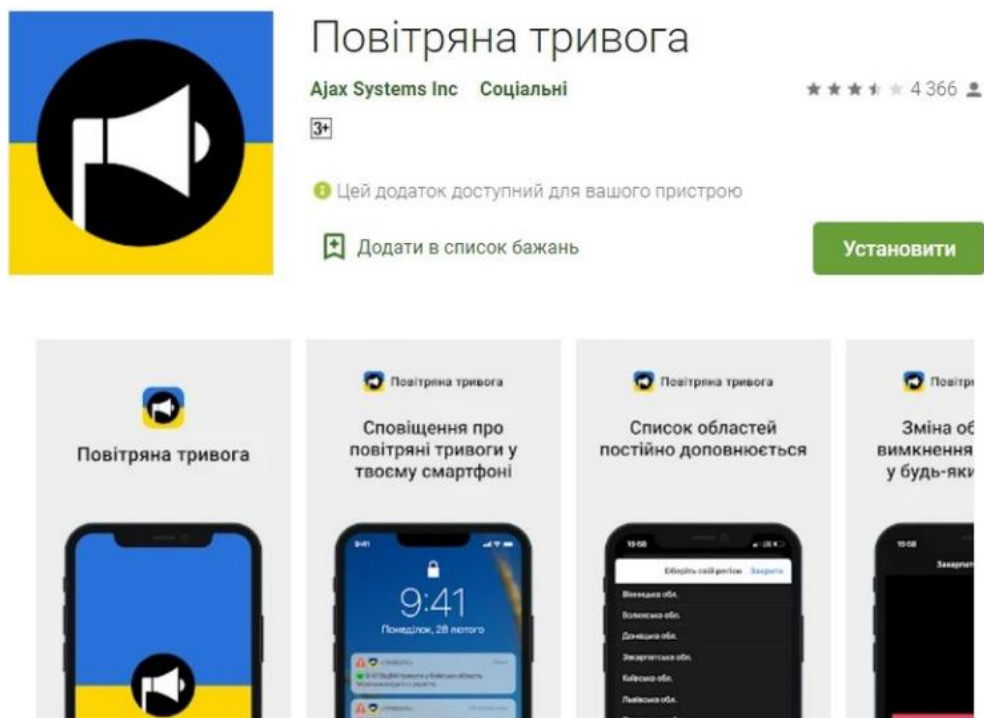


Рисунок 1.2 – Головний вигляд застосунку «Air Alert»

Ще одна важлива категорія – застосунки, які поєднують інформацію та інтерактивність. Наприклад, програма MyShake, розроблена Каліфорнійським університетом у Берклі, використовує дані датчиків смартфона для виявлення сейсмічної активності та сповіщення користувачів про землетруси. Інтерактивні функції дозволяють користувачам не тільки отримувати інформацію, а й брати активну участь у процесі моніторингу надзвичайних ситуацій.

Аналіз існуючих рішень показує, що ефективність мобільних застосунків у попередженні та реагуванні на надзвичайні ситуації залежить від кількох ключових факторів: інформаційної актуальності та надійності, простоти використання та інтерактивності. Враховуючи ці фактори, розробка нового мобільного застосунку для екстрених дій була розроблена таким чином, щоб максимально задовольнити потреби користувачів у швидкому доступі до важливої інформації та чітких інструкцій щодо безпечних дій у різних ситуаціях, які ми надаємо.

Тому основними вимогами до нових програм є інтеграція з надійними джерелами даних, зручний інтерфейс користувача та можливість інтерактивного навчання. Це забезпечує ефективну підготовку та реагування на надзвичайні ситуації та захищає життя та здоров'я користувачів.

1.2 Аналіз інформаційного забезпечення предметної області

Інформаційне забезпечення під час надзвичайних ситуацій є дуже важливим елементом захисту життя та здоров'я населення. Своєчасна та достовірна інформація може значно зменшити втрати та збитки, запобігти паніці та допомогти організувати евакуацію та інші необхідні заходи. Основними джерелами інформації в цій галузі є національні установи, професійні установи, наукові установи та засоби масової інформації.

Державні органи, такі як Державна служба України з надзвичайних ситуацій (ДСНС), відіграють ключову роль у наданні громадськості надзвичайних повідомлень, рекомендацій щодо дій та інформації про поточну ситуацію. Ці установи використовують різні канали для поширення інформації, включаючи телебачення, радіо, веб-сайти та мобільні застосунки. Інформація, яку надає EMS, часто включає прогнози розвитку подій, інструкції щодо евакуації та запобіжні заходи, яких повинні вжити громадяни.

Академічні установи також відіграють важливу роль у забезпеченні готовності до надзвичайних ситуацій. Вони проводять дослідження та розробляють оцінку ризиків надзвичайних ситуацій і плани дій. Науковці публікують результати своїх досліджень у відкритих наукових журналах, що дозволяє інтегрувати наукові знання з реальними заходами безпеки. Наприклад, дослідження в галузі сейсмології допомагають передбачити землетруси, а дослідження в галузі кліматології допомагають попередити про можливі зміни клімату, які можуть призвести до екстремальних погодних умов.

Важливу роль у поширенні інформації про надзвичайні ситуації відіграють засоби масової інформації та соціальні мережі. Вони швидко доносять важливі повідомлення та інструкції до широкої аудиторії. Однак, на відміну від офіційних джерел, інформація ЗМІ може бути менш надійною через ризик поширення недостовірних даних. Особливо це актуально в період кризи, коли швидкість поширення інформації може переважити її точність. Тому важливо, щоб громадяни

користувалися перевіреними джерелами інформації і не лякалися неперевіреними новинами.

Мобільні програми стають все більш популярними як ефективний, зручний та інтерактивний спосіб доставки інформації. Вони дозволяють отримувати дані з офіційних джерел і надавати користувачам персоналізовані сповіщення, покрокові інструкції та зображення. Наприклад, програми можуть попереджати про наближення урагану, пропонувати маршрути евакуації або повідомляти про розташування укриттів. Це дозволяє швидко реагувати на зміну обставин і приймати зважені рішення в надзвичайних ситуаціях.

Таким чином, надання інформації про катастрофу є багаторівневим процесом, який включає збір, аналіз і поширення критично важливих даних з різних джерел з метою забезпечення ефективного реагування на надзвичайні ситуації. Це включає роботу державних органів, професійних організацій, наукових установ і засобів масової інформації, а також використання сучасних технологій, таких як мобільні застосунки, щоб надати громадськості необхідну інформацію у разі небезпеки.

1.3. Засоби та методи вирішення проблеми

У процесі розробки інформаційних систем для надзвичайних ситуацій, велике значення має вибір належних засобів та методів, що гарантують ефективність та надійність програмного забезпечення. У рамках кваліфікаційної роботи, спрямованого на створення мобільного застосунку для дій у надзвичайних ситуаціях, важливо здійснити оптимальний вибір інструментів та підходів, які дозволять розробити інтуїтивно зрозумілий та функціональний продукт.

Одним із ключових моментів при розробці є вибір мови програмування та платформи. Для реалізації мобільного застосунку було вирішено обрати движок Unity та мову програмування C#. Unity - потужний ігровий движок, який надає

широкі можливості для створення інтерактивних застосунків не тільки для ігор, але й для інших сфер, зокрема для екстрених ситуацій. Універсальність, кросплатформна розробка та велика спільнота розробників роблять Unity ідеальним вибором для застосунку.

Мова програмування C# забезпечує високу продуктивність і зручний синтаксис, що дозволяє розробляти складні системи. Використання C# в Unity дозволяє ефективно керувати логікою програми, взаємодіяти з елементами інтерфейсу та працювати з даними.

2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА СТРУКТУРИ ПРОГРАМНОГО ПРОДУКТУ

2.1 Розробка архітектури мобільного застосунку

Архітектура програмного продукту є одним із найважливіших етапів розробки, оскільки вона визначає основні компоненти системи та зв'язки між ними. Це впливає на продуктивність, масштабованість, надійність і ремонтпридатність програмного забезпечення.

Архітектура мобільного застосунку базується на таких принципах:

- Модульність: система складається з незалежних модулів, кожен з яких виконує певні функції.
- Мобільність: застосунок розроблено для роботи на різних мобільних платформах.
- Масштабованість: архітектура дозволяє легко додавати нові функції.
- Надійність: використання перевірених підходів і технологій для забезпечення стабільної роботи.
- Простота використання: інтуїтивно зрозумілий інтерфейс для зручної взаємодії з системою.

Архітектура мобільного застосунку для роботи в надзвичайних ситуаціях складається з наступних компонентів:

1) Основне вікно (Main Form):

- Основний компонент користувацького інтерфейсу.
- Містить чотири основні кнопки навігації.

2) Інформаційна форма (Information Form):

- Відображає інформацію про дії в надзвичайних ситуаціях.
- Містить декілька вкладок для впорядкування вмісту.

3) Форма карти (Map Form):

- Містить кнопки для переходу на різні веб-сайти з картами тривоги.

- Розробляє зовнішні веб-ресурси для відображення карт.

4) Форма ілюстрацій (Illustration Form):

- Показує ілюстрації, пов'язані з реагуванням на надзвичайні ситуації.
- Він має структуру, подібну до форми інформації.

5) Форма вікторини/тест:

- Запускає тест із запитаннями для перевірки знань користувача.
- Відображає результати тесту після завершення.

Розробка архітектури для програмного продукту є важливим кроком у створенні мобільного застосунку для дій під час надзвичайних ситуацій. Він забезпечує основу для розробки, тестування та впровадження системи, а також гарантує її ефективність, надійність і простоту використання.

2.2 Аналіз інструментів для розробки

Розробка мобільного застосунку вимагає використання різноманітних інструментів, які забезпечують ефективну реалізацію функціональності, продуктивність, тестування та підтримку програмного продукту. У цьому розділі розглядаються основні інструменти, які використовуються для створення мобільного плану на випадок надзвичайних ситуацій.

Для розробки застосунку була обрана мова програмування C#, що використовується в середовищі Unity. C# забезпечує високу продуктивність, читабельність і простоту використання, які є важливими аспектами для розробки складних мобільних програм.

Unity — потужне середовище розробки для створення 2D і 3D ігор, а також інтерактивних застосунків. Він надає широкий набір інструментів, які дозволяють створювати інтерфейс користувача та інтеграції з іншими компонентами системи. Unity підтримує кросплатформну розробку, дозволяючи створювати програми для різних мобільних платформ, таких як iOS і Android.

Аналіз засобів розробки показав, що для створення мобільного застосунку для реагування на надзвичайні ситуації використовувалися найрізноманітніші інструменти та технології. Вибір C# і Unity в якості основних засобів розробки забезпечив високу продуктивність і зручність у створенні інтерфейсу користувача, обробці даних і реалізації логіки програми.

2.3 Середовище проектування Draw io

Draw.io, також відомий як diagrams.net, є потужним онлайн-інструментом для створення різноманітних діаграм, включаючи діаграми UML. Завдяки дружньому інтерфейсу та широкому функціоналу він став вибором багатьох розробників та аналітиків у всьому світі.

Draw.io має великий набір інструментів і функцій, які дозволяють легко створювати різноманітні діаграми. Основними перевагами цього середовища є [2]:

- 1) Інтуїтивно зрозумілий інтерфейс: Draw.io має простий і зрозумілий інтерфейс, який дозволяє новачкам швидко навчитися ним користуватися. Завдяки цьому користувачі можуть легко створювати діаграми без особливих зусиль.
- 2) Широкий набір шаблонів: Draw.io пропонує широкий вибір шаблонів для різних типів діаграм. Від діаграм UML до організаційних діаграм і діаграм потоків даних ви знайдете все необхідне для створення високоякісних діаграм.
- 3) Співпраця: Draw.io підтримує співпрацю з діаграмами, дозволяючи кільком користувачам вносити зміни одночасно. Це особливо корисно для командної роботи над проектами.
- 4) Експорт та імпорт: інструмент надає можливість експортувати та імпортувати діаграми в різні формати, включаючи PNG, JPEG, SVG, PDF та інші. Це робить його універсальним інструментом для обміну та

використання діаграм за межами самого середовища.

Для діаграм UML Draw.io пропонує можливість створювати всі основні типи, зокрема:

- діаграми класів**, що показують структуру системи, включаючи класи, їхні властивості та методи.
- Діаграми об'єктів, що показують конкретні екземпляри класів та їхні зв'язки.
- Діаграми варіантів використання, які використовуються для визначення функціональних вимог системи.
- Діаграми послідовностей і зв'язків, які відображають взаємодію об'єктів і компонентів системи.

Таким чином, Draw.io є не тільки потужним інструментом для створення діаграм, а й надійним помічником для аналізу, візуалізації та документування проектів будь-якої складності. Зручний інтерфейс і широкий функціонал роблять його універсальним інструментом для роботи з будь-якими діаграмами.

2.4. Розробка діаграми прецедентів

Діаграма прецедентів є важливим інструментом у розробці програмного забезпечення, який використовується для моделювання взаємодії між користувачами та системою. Це дозволяє зрозуміти, які дії можуть виконувати користувачі та як вони впливають на систему. Основна мета діаграми випадку – визначити функціональні вимоги системи та визначити основні сценарії взаємодії.

Склад:

- 1) Актори: Актори — це зовнішні сутності або системи, які взаємодіють із програмою. Ними можуть бути реальні користувачі або інші системи, які взаємодіють з вашим програмним продуктом.
- 2) Прецедент: Прецеденти – це конкретні дії або сценарії, які виконуються користувачами або системою. Вони визначають функціональні можливості

системи та визначають, як користувачі можуть взаємодіяти з програмним продуктом.

Схема прецедентів для кваліфікаційної роботи показує основні можливості користувачів і те, як вони взаємодіють із програмою в надзвичайних ситуаціях. На діаграмі показано різні сценарії, як-от перегляд інформації про реагування на надзвичайні ситуації, відкриття карти тривоги, перегляд ілюстрацій і виконання тестів. Ця діаграма допоможе зрозуміти, які функції має програма та як користувачі можуть з нею взаємодіяти.

Діаграма прецедентів для застосунку дій у надзвичайних ситуаціях зображена на рисунку 2.4.1.

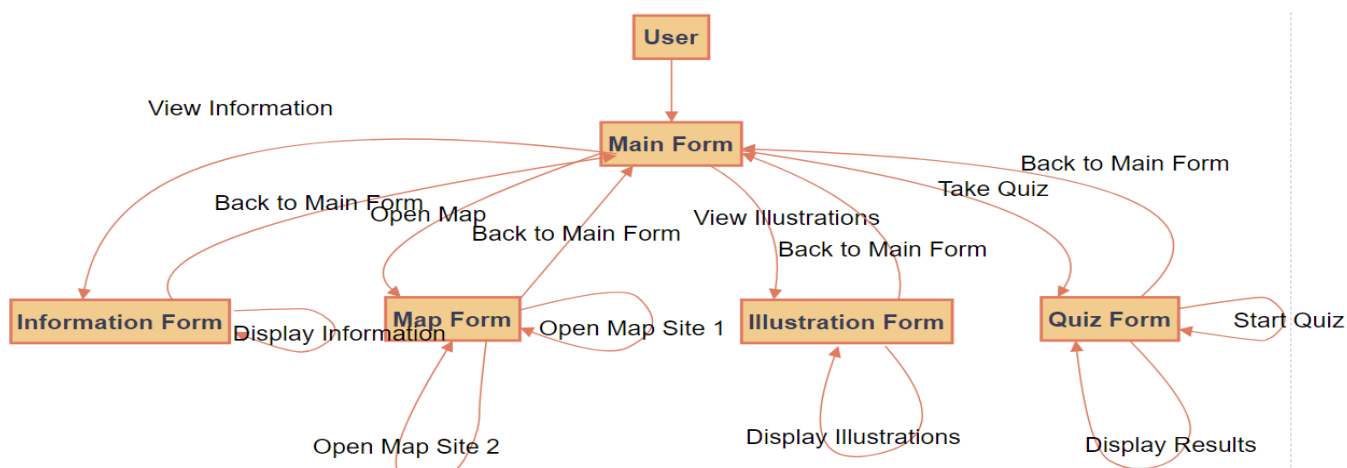


Рисунок 2.4.1 – Діаграма прецедентів для застосунку дій у надзвичайних ситуацій

2.5 Розробка діаграми діяльності

Діаграма діяльності - це графічний інструмент котрий використовують для моделювання послідовності дій та процесів у системі. Він дозволяє візуалізувати взаємодію між об'єктами чи класами та показати послідовність подій у процесі виконання діяльності.

Елементи діаграми:

- 1) Дії: представляють конкретні кроки або дії, які виконуються в процесі.

- 2) Рішення: представити вибір або альтернативи, які можуть виникнути в процесі реалізації.
- 3) Паралельні фрагменти (Parallel Fragments): використовуються для імітації одночасного виконання операцій.
- 4) Умовні розриви: показують умовні переходи або розгалуження в процесі.

Розробка табеля засобів проведення дій у надзвичайних ситуаціях:

Діаграма діяльності для кваліфікаційної роботи демонструє послідовність дій користувача в надзвичайних ситуаціях. Він показує кроки, які користувачі повинні виконати, щоб отримати потрібну інформацію, відкрити картки персонажів, переглянути ілюстрації та виконати тести.

Діаграма діяльності для застосунку дій у надзвичайних ситуаціях зображена на рисунку 2.5.1.

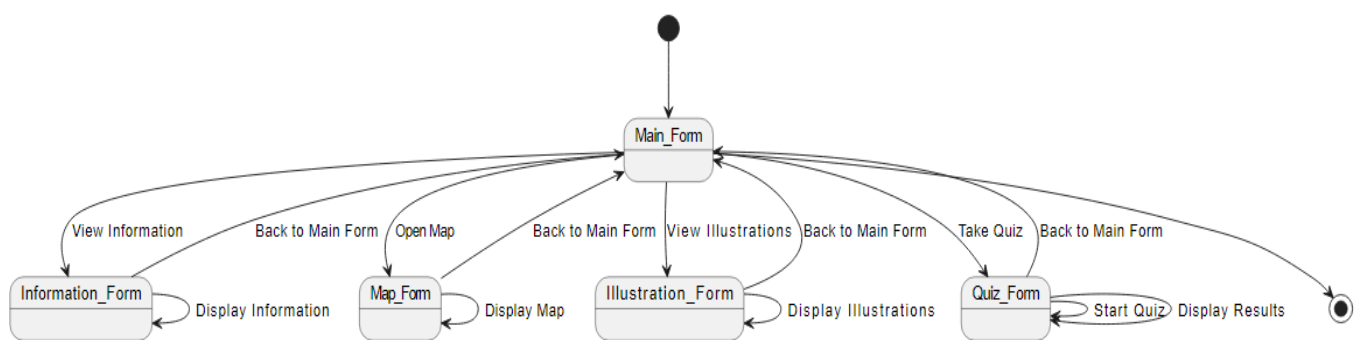


Рисунок 2.5.1 – Діаграма діяльності для застосунку дій у надзвичайних ситуаціях

Ця схема наочно демонструє процес взаємодії користувача з застосунком і допомагає зрозуміти послідовність дій в екстрених ситуаціях.

2.6 Розробка діаграма класів

Діаграма класів є частиною найважливіших діаграм у розробці об'єктно-орієнтованого програмного забезпечення. Вона використовується для

моделювання статичної структури системи, що представляє класи, їх властивості, методи та зв'язки між класами.

Основна мета діаграми класів — проілюструвати та задокументувати архітектуру програмної системи.

Склад:

- 1) Класи (Classes): Основні елементи діаграми, які представляють сутності системи. Класи містять атрибути (дані) і методи (функції або поведінку).
- 2) Атрибути (Attributes): Властивості або змінні, що зберігають стан об'єкта класу.
- 3) Методи (Methods): Операції або функції, що визначають поведінку класу.
- 4) Зв'язки (Relationships): Відображають взаємодію між класами. Основні типи зв'язків:
 - Асоціація (Association): Загальний зв'язок між класами, що показує, як об'єкти одного класу пов'язані з об'єктами іншого класу.
 - Агрегація (Aggregation): Спеціальний вид асоціації, який вказує на "частину-цілого" відношення.
 - Композиція (Composition): Сильний вид агрегації, де частини не можуть існувати без цілого.
 - Наслідування (Inheritance): Відношення між базовим класом і похідним класом, що дозволяє похідному класу успадковувати атрибути і методи базового класу.

Діаграма класів для кваліфікаційної роботи показує структуру вашої програми та зв'язки між її основними компонентами. Він включає такі класи, як MainForm, InformationForm, MapForm, IllustrationForm, QuizForm, а також допоміжні класи для адміністрування тестів і обробки URL-адрес.

Кожен клас містить власні атрибути та методи, які відображають його функціональність і поведінку. Наприклад, клас QuizForm містить властивості для питань, відповідей, результатів тесту та методи обробки відповідей і відображення результатів.

Діаграма класів для застосунку дій у надзвичайних ситуаціях зображена на рисунку 2.6.1.

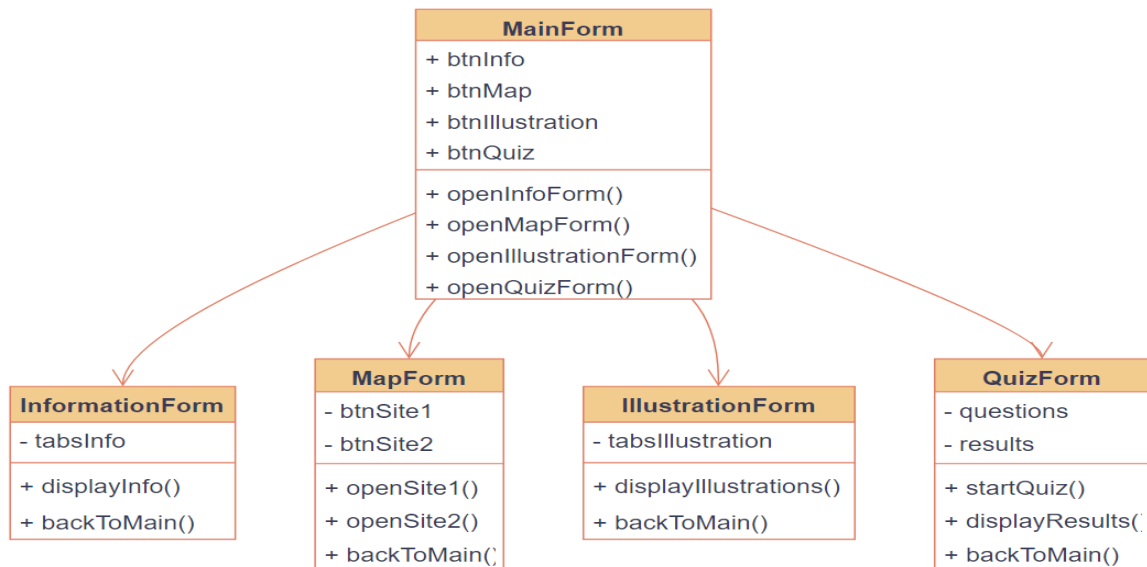


Рисунок 2.6.1 – Діаграма класів для застосунку дій у надзвичайних ситуацій

Ця діаграма допомагає зрозуміти статичну структуру системи, зв'язки між її компонентами та забезпечити правильну реалізацію функціональних вимог.

Розробка діаграми класів є критично важливим етапом у процесі розробки програмного забезпечення, так як вона забезпечує візуальне представлення архітектури системи, допомагає визначити основні компоненти та їх взаємодію, а також служить основою для подальшого впровадження та налагодження програмного продукту.

2.7 Розробка діаграма послідовності

Діаграма послідовності є важливим інструментом у розробці програмного забезпечення, який використовується для моделювання динамічних аспектів системи. Він показує, як об'єкти системи взаємодіють один з одним у певній послідовності подій. Основна мета діаграми послідовності — проілюструвати порядок, у якому викликаються методи та обмінюються повідомленнями між об'єктами під час виконання сценарію.

Діаграма послідовності для кваліфікаційної роботи показує послідовність подій під час взаємодії користувача з програмою в різних сценаріях. Наприклад, діаграма може показати, як користувач переходить від головного вікна до інформаційної форми, відкриває карту сигналів, відображає діаграми або виконує тест.

Діаграма послідовності для застосунку дій у надзвичайних ситуаціях зображена на рисунку 2.7.1.

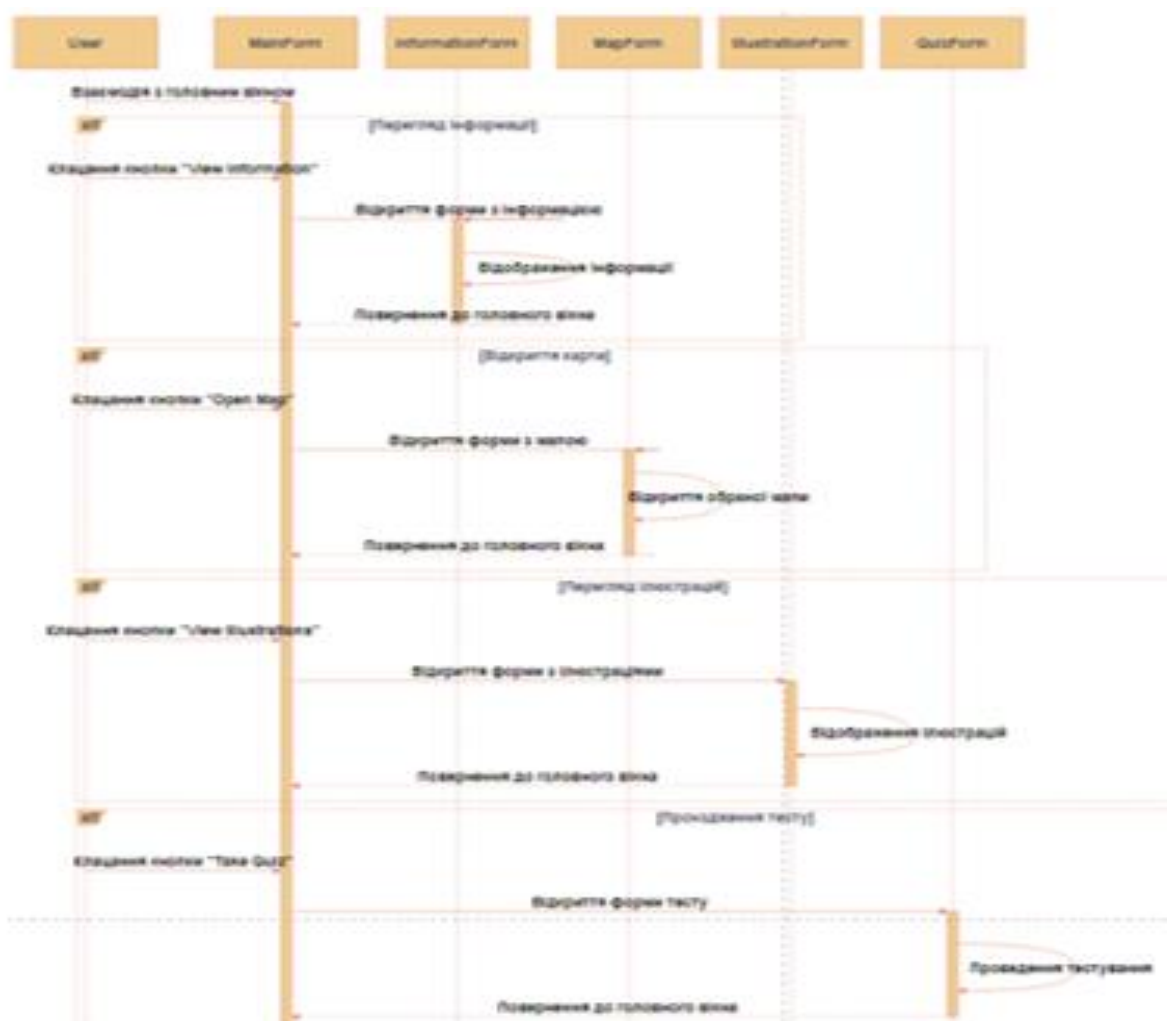


Рисунок 2.7.1 – Діаграма послідовностей для застосунку дій у надзвичайних ситуаціях

Ця діаграма допомагає зрозуміти динаміку взаємодії між об'єктами системи під час різних сценаріїв. Це важливий інструмент для виявлення потенційних

проблем сумісності та забезпечення правильного виконання функціональних вимог.

Розробка діаграми послідовності є важливим етапом у процесі проектування програмного забезпечення, оскільки вона дозволяє створити детальну модель динамічних аспектів системи, ілюструючи послідовність взаємодії між об'єктами та забезпечуючи правильне виконання сценаріїв користувача. Це допомагає зрозуміти, як буде поводитися система в реальних умовах експлуатації, а також вчасно виявити й усунути можливі дефекти.

2.8 Розробка діаграми кооперації

Діаграма кооперацій є важливим інструментом у розробці об'єктно-орієнтованого програмного забезпечення. Він використовується для моделювання взаємодії між об'єктами в системі, зосереджуючись на структурних аспектах цієї взаємодії.

Основна мета діаграми співпраці — показати повідомлення, що передаються між об'єктами, а також показати зв'язки між цими об'єктами.

Діаграма кооперацій для кваліфікаційної роботи показує взаємодію між різними об'єктами в системі під час певних сценаріїв. Він показує, як об'єкти обмінюються повідомленнями для досягнення певної мети, наприклад відображення інформації, відкриття карти сигналів, перегляд діаграм або виконання тестів.

Діаграма кооперацій для застосунку дій у надзвичайних ситуаціях зображена на рисунку 2.8.1.

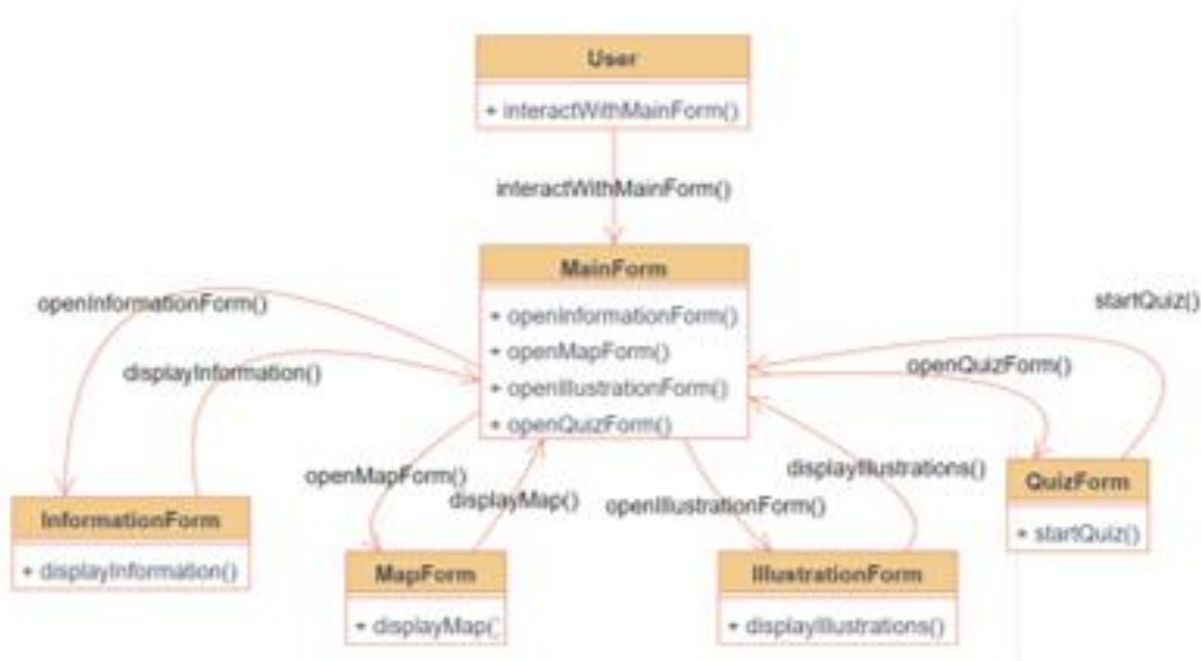


Рисунок 2.8.1 – Діаграма кооперацій для застосунку дій у надзвичайних ситуаціях

Ця діаграма допомагає зрозуміти, як системні об'єкти взаємодіють один з одним під час різних сценаріїв, забезпечуючи правильне виконання функціональних вимог. Це також допомагає виявити та вирішити можливі проблеми у взаємодії між об'єктами.

Розробка діаграми кооперацій є важливим кроком у процесі розробки програмного забезпечення. Він дозволяє моделювати та візуалізувати взаємодії між об'єктами, показуючи, як об'єкти обмінюються повідомленнями для виконання певних дій. Це сприяє кращому розумінню динаміки системи та забезпечує її коректне функціонування в реальних умовах.

2.9 Створення діаграми компонентів

Діаграма компонентів є важливим інструментом у розробці програмного забезпечення, який використовується для моделювання фізичної структури системи. Він показує, як програмні компоненти системи організовані та

взаємодіють один з одним. Основна мета діаграми компонентів — показати статичну структуру системи, включаючи програмні модулі, бібліотеки, файли та інші компоненти, а також їхні взаємозв'язки.

Діаграма компонентів для кваліфікаційної роботи показує основні програмні модулі та їхні зв'язки. Він показує, як різні частини системи організовані та взаємодіють одна з одною для забезпечення функціональності плану в надзвичайних ситуаціях.

Діаграма компонентів для застосунку дій у надзвичайних ситуаціях зображена на рисунку 2.9.1.

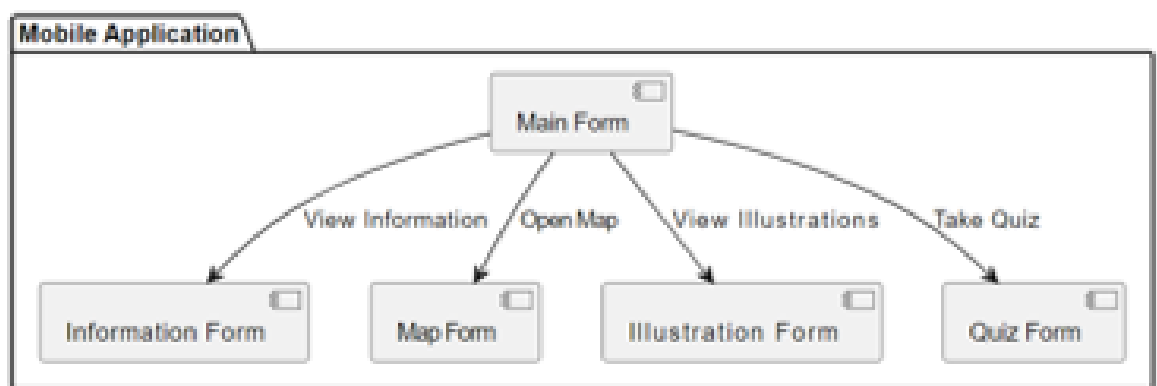


Рисунок 2.9.1 – Діаграма компонентів для застосунку дій у надзвичайних ситуаціях

Ця діаграма допомагає зрозуміти, як організовані програмні модулі системи та як вони взаємодіють один з одним. Це також допомагає виявити та вирішити можливі проблеми в структурі системи.

Створення діаграми компонентів є важливим кроком у процесі розробки програмного забезпечення. Він дозволяє візуалізувати фізичну структуру системи, показуючи, як програмні модулі організовані та взаємодіють один з одним. Це сприяє кращому розумінню архітектури системи та забезпечує її коректне функціонування в реальних умовах.

3. КОНСТРУЮВАННЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ

3.1 Вибір мови програмування та інструментів розробки

При виборі мови програмування та технології розробки для створення мобільного застосунку для дій у надзвичайних ситуаціях було вирішено використовувати платформу Unity та мову програмування C#.

Unity — це гнучкий і потужний механізм, який використовується для створення як 2D, так і 3D проектів для різних платформ, включаючи ПК, мобільні пристрої та консолі. Unity був випущений у 2005 році і з тих пір став одним із найпопулярніших інструментів розробки програмного забезпечення [3].

Однією з ключових особливостей Unity є його універсальність, що робить його привабливим для розробників різних типів програм. Гнучкий графічний механізм Unity дозволяє створювати чудові візуальні ефекти, а потужні інструменти сценаріїв C# дозволяють розробникам реалізувати складну логіку застосунків.

Щоб почати роботу з Unity, потрібно встановити механізм і налагодити середовище. Після встановлення движка необхідно створити новий проект і вибрати його тип (2D або 3D), залежно від потреб вашої програми.

Інтерфейс Unity розділений на кілька панелей, кожна з яких відповідає за певну функцію. Основні плати включають:

- Сцена: тут ви створюєте та редагуєте свої рівні та світи.
- Ієрархія: показує структуру об'єктів у вашій сцені.
- Проект: містить усі ресурси для вашого проекту, такі як текстури, моделі та сценарії.
- Інспектор: дозволяє редагувати властивості вибраного об'єкта.
- Консоль: відображає інформацію про процес розробки та помилки.

Щоб створити мобільний застосунок, потрібно створити об'єкти, додати до них компоненти та визначити їхню поведінку.

Програмування в Unity вимагає навичок написання сценаріїв на основі мови програмування C#. Основи включають взаємодію з Unity API для роботи з об'єктами, фізикою та графікою. Скрипти, створені на основі класу MonoBehaviour, надають доступ до функцій двигуна розробки.

Такі методи, як Start() і Update(), використовуються для керування логікою програми, а компонент Transform дозволяє керувати положенням, поворотом і масштабом об'єктів. Сценарії обробляють взаємодію користувача, події та зміну властивостей об'єктів.

Використання популярних шаблонів програмування, таких як Singleton, Observer і Factory, допомагає ефективно організувати код.

Дотримання найкращих практик, включаючи принципи чистого коду, документування та оптимізації, забезпечує створення ефективних і читабельних програмних продуктів.

Вибір Unity та C# для кваліфікаційної роботи бакалавра зумовлений кількома ключовими факторами:

- 1) Кросплатформенність: Unity дозволяє створювати проекти, які можна легко перенести на різні платформи, включаючи Android та iOS.
- 2) Простота використання: Unity пропонує інтуїтивно зрозумілий інтерфейс і потужні інструменти для швидкого створення прототипів і кінцевого продукту.
- 3) Спільнота та підтримка: Unity має велику та активну спільноту розробників, яка дозволяє швидко знаходити відповіді на запитання та вирішувати проблеми, що виникають під час розробки.
- 4) Потужні можливості створення інтерфейсу користувача: Unity надає різноманітні інструменти для створення та налаштування інтерфейсу користувача, який є критично важливим для мобільних програм.

Таким чином, вибір Unity та C# для розробки мобільного застосунку для дій у надзвичайних ситуаціях виправданий і надає всі інструменти та можливості, необхідні для успішного завершення виконання кваліфікаційної роботи.

3.2 Розробка інтерфейсу та логіки застосунку

Інтерфейс користувача (GUI) і розробка логіки програми є ключовими кроками в процесі створення мобільної програми реагування на надзвичайні ситуації. У цьому розділі детально описано процес створення графічного інтерфейсу та реалізації основних функцій програми за допомогою середовища розробки Unity та мови програмування C#.

Unity надає зручні інструменти для створення інтерфейсів користувача. Основні елементи інтерфейсу, такі як кнопки, текстові поля та вкладки, розміщені на сцені та організовані у відповідних панелях (Canvas). Інтерфейс розділений на кілька основних форм, кожна з яких виконує певні функції. Елементи для створення інтерфейсу знаходяться у вкладці `GameObject>UI`. Вигляд цієї вкладки зображено на рисунку 3.2.1.

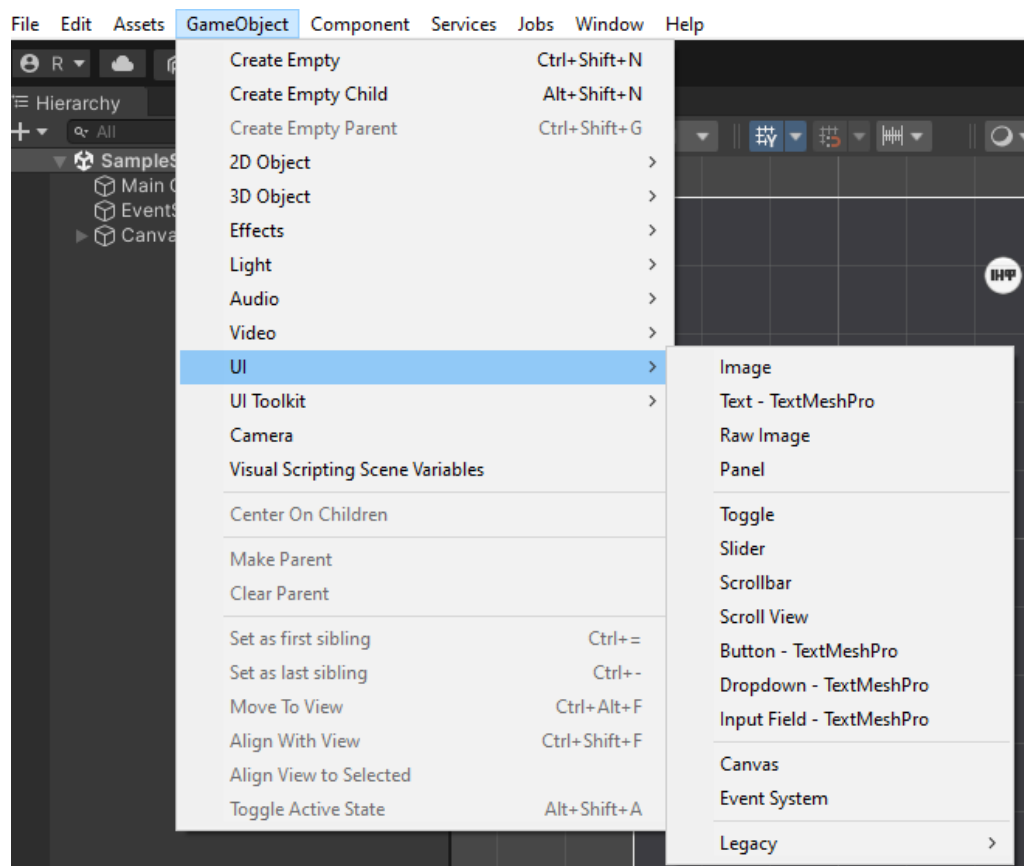


Рисунок 3.2.1 – Вигляд вкладки GameObject

Головна форма застосунку містить чотири кнопки, які переводять користувача в інші форми: відображення інформації, відкриття карти тривоги, відображення ілюстрацій і виконання тесту. Вигляд такої форми зображено на рисунку 3.2.2.

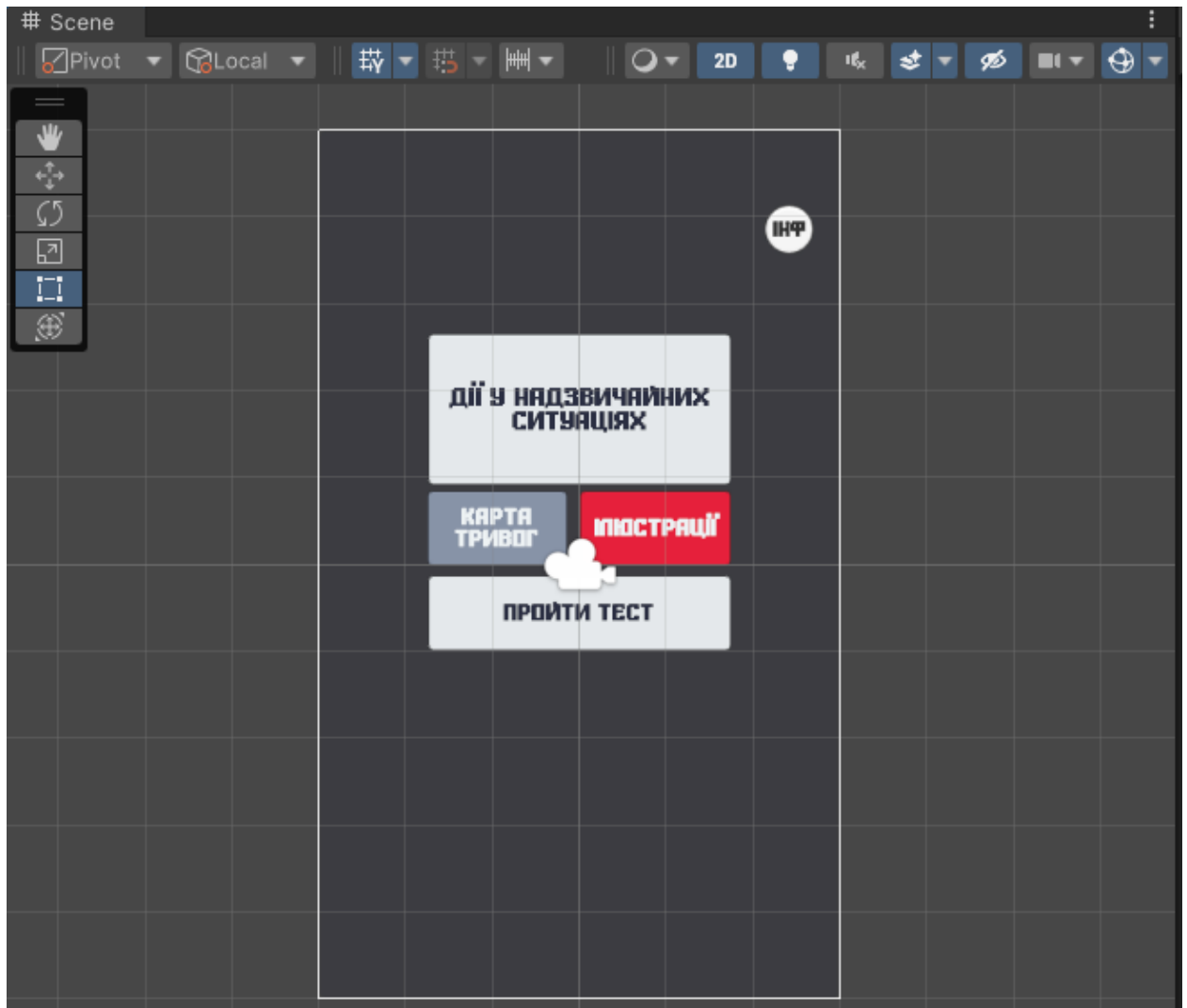


Рисунок 3.2.2 – Головна форма застосунку

Для початку, щоб розмістити на панель об'єкти потрібно спершу помістити панель у ієрархію компонента Canvas. Потім компоненту Canvas у пункті Render Camera потрібно помістити головну камеру проекту. Вигляд компонентів Canvas зображено на рисунку 3.2.3. Вигляд ієрархії головної форми зображено на рисунку 3.2.4.

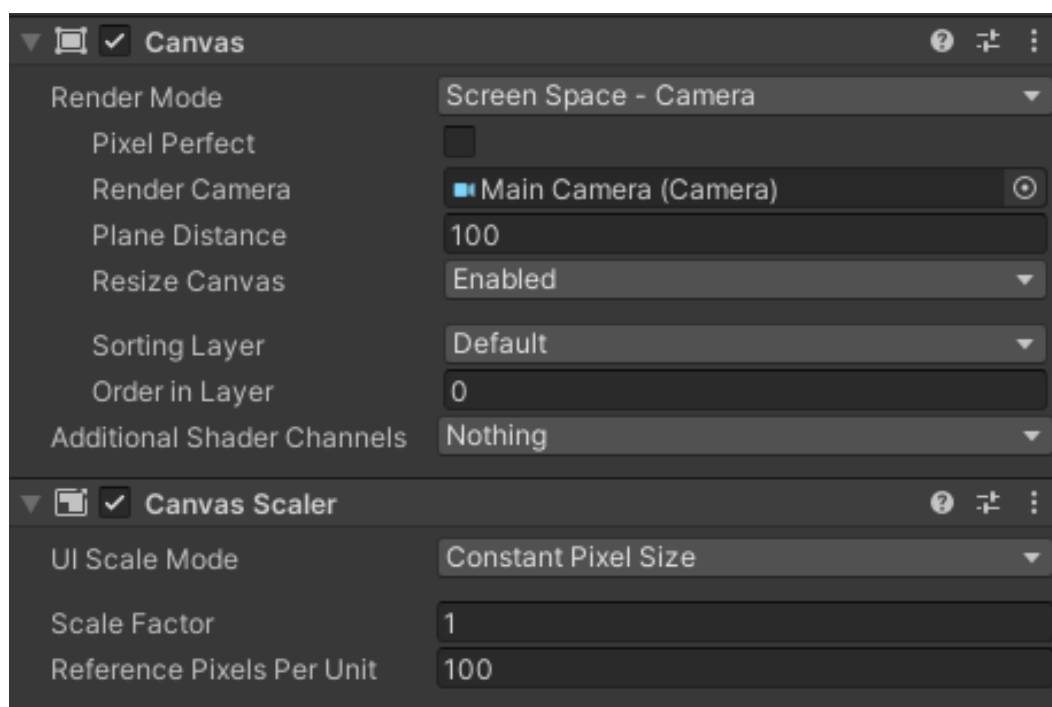


Рисунок 3.2.3 – Елементи компонента Canvas

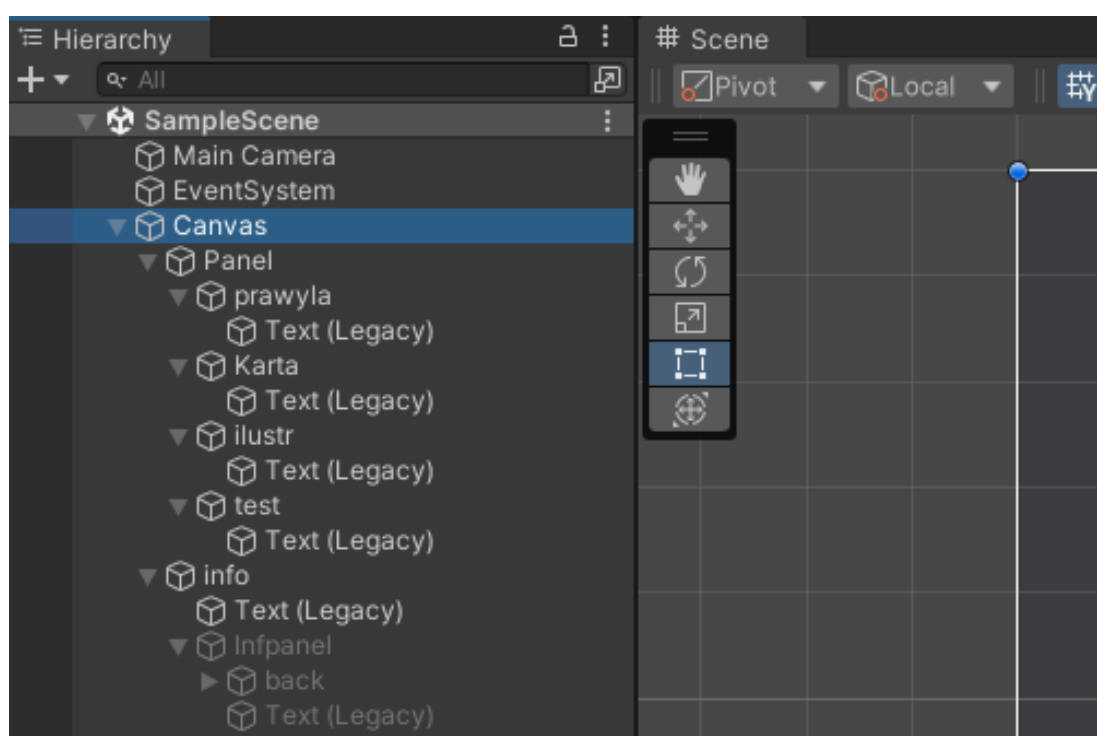


Рисунок 3.2.3 – Вигляд ієрархії головної форми

Елемент панелі отримав відповідний колір і розмір для відповідного відображення на екранах мобільних пристроїв. Зовнішній вигляд елементів компонента панелі наведено на рисунку 2.3.5.

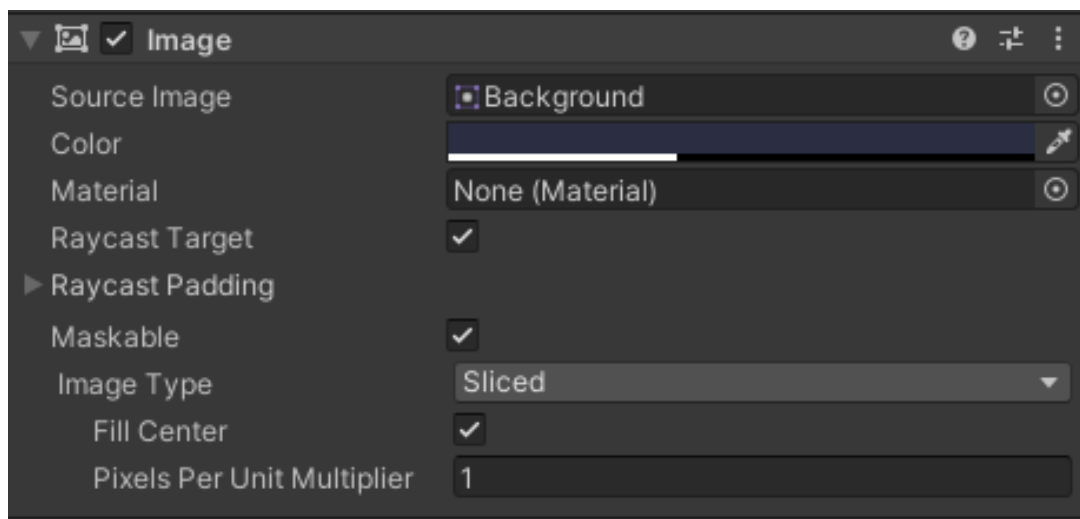


Рисунок 3.2.5 – Вигляд елементів компонента Panel

Головний вигляд форми у режимі «Play» зображено на рисунку 3.2.6.



Рисунок 3.2.6 – Вигляд головної форми в режимі «Play»

На кожній кнопці на події OnClick знаходиться скрипт котрий відповідає за зміну сцени. Вигляд такої події зображено на рисунку 3.2.7.

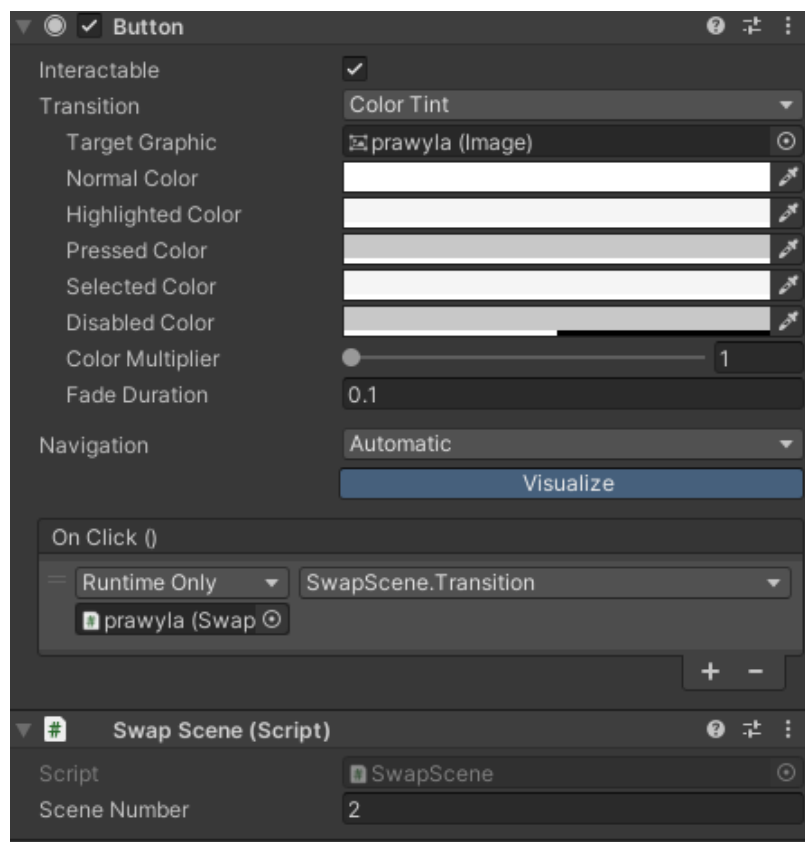


Рисунок 3.2.7 – Вигляд події OnClick у компонента кнопки

Після додавання C# скрипта до елемента кнопки в інспекторі з'являється нове поле під назвою «Scene Number». У цьому полі можна ввести номер сцени, на яку буде переходити застосунок після натискання на кнопку. Кожна сцена в проєкті має свій унікальний порядковий номер.

Щоб дізнатися цей номер, необхідно відкрити меню Build Settings, яке знаходиться за шляхом: File > Build Settings. У цьому меню відображається список усіх сцен, доданих до проєкту, разом із їхніми порядковими номерами. Вигляд цього списку показано на рисунку 3.2.8.

Щоб додати сцену до списку, потрібно спершу створити або відкрити існуючу сцену, а потім додати її до Build Settings. Для цього необхідно натиснути кнопку «Add Open Scenes». Сцена буде додана до списку з унікальним порядковим

номером, який можна використовувати в полі «Scene Number» для налаштування переходів між сценами.

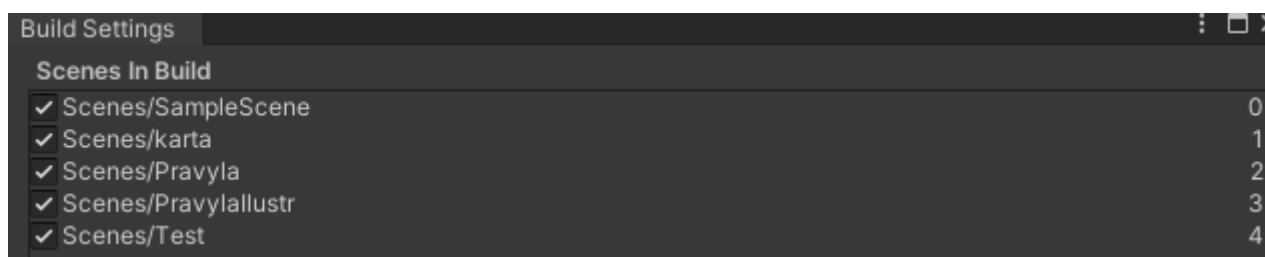


Рисунок 3.2.8 – Вигляд списку сцен

С# код який відповідає за перемикання сцен зображено на рисунку 3.2.9.

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class SwapScene : MonoBehaviour
{
    public int sceneNumber;
    public void Transition()
    {
        SceneManager.LoadScene(sceneNumber);
    }
}
```

Рисунок 3.2.9 – С# код «SwapScene»

Цей код призначений для переходів між сценами в Unity. Це дозволяє легко налаштувати, до якої сцени робити перехід, просто змінивши значення sceneNumber у редакторі. При виклику методу переходу (наприклад, при натисканні кнопки) за допомогою SceneManager.LoadScene завантажується сцена з відповідним номером.

Форма «Карта тривоги» перехід на яку відбувається з кнопки «Карта тривоги» зображена на рисунку 3.2.10.

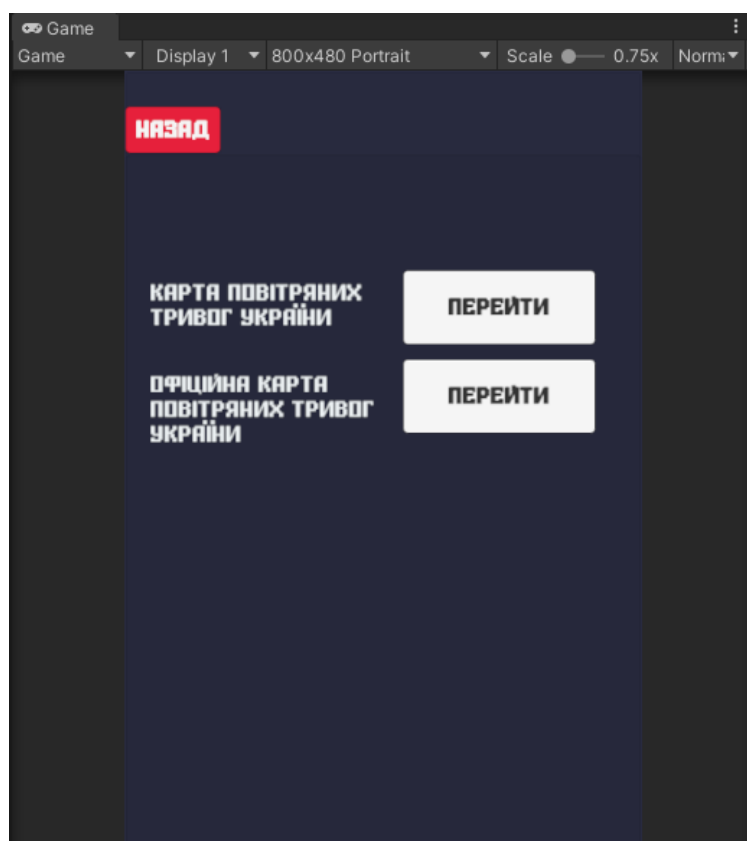


Рисунок 3.2.10 – Вигляд форми «Карта тривог»

На даній формі розміщено дві кнопки, які відразу після кліку пересилають на два альтернативних сайти карт тривоги, що є офіційними в Україні та мають велику популярність.

Ці дві кнопки мають на собі C# скрипт який діє під час події «OnClick». Вигляд елементів однієї з кнопок зображено на рисунку 3.2.11.

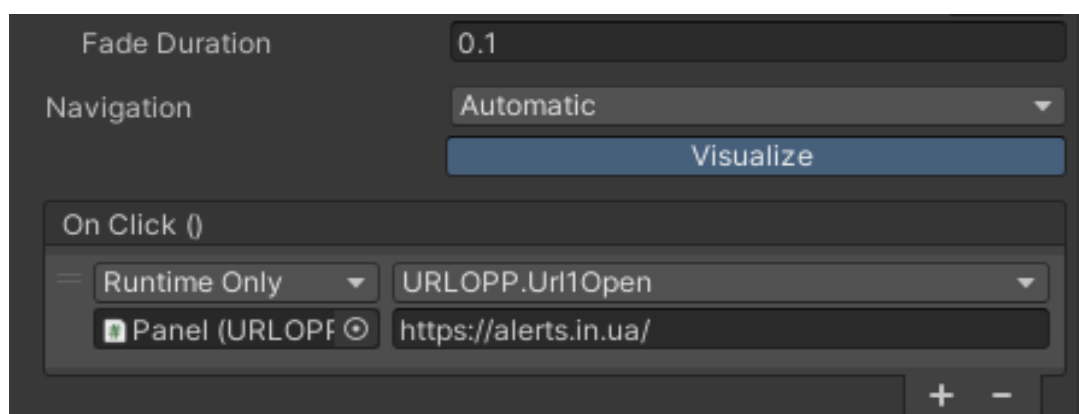


Рисунок 3.2.11 – Вигляд елементів компонента Button кнопки «Перейти»

C# код який відповідає за перехід на сайт з мапами тривоги зображено на рисунку 3.2.12.

```
using UnityEngine;

public class URLOPP : MonoBehaviour
{
    public void Url1Open(string url)
    {
        Application.OpenURL(url);
    }
    public void Url2Open(string url)
    {
        Application.OpenURL(url);
    }
}
```

Рисунок 3.2.12 – C# код «URLOPP»

Форми «Дії у надзвичайних ситуаціях» Вигляд сцени і ієрархії компонентів форми «Дії у надзвичайних ситуаціях» зображено на рисунку 3.2.13.

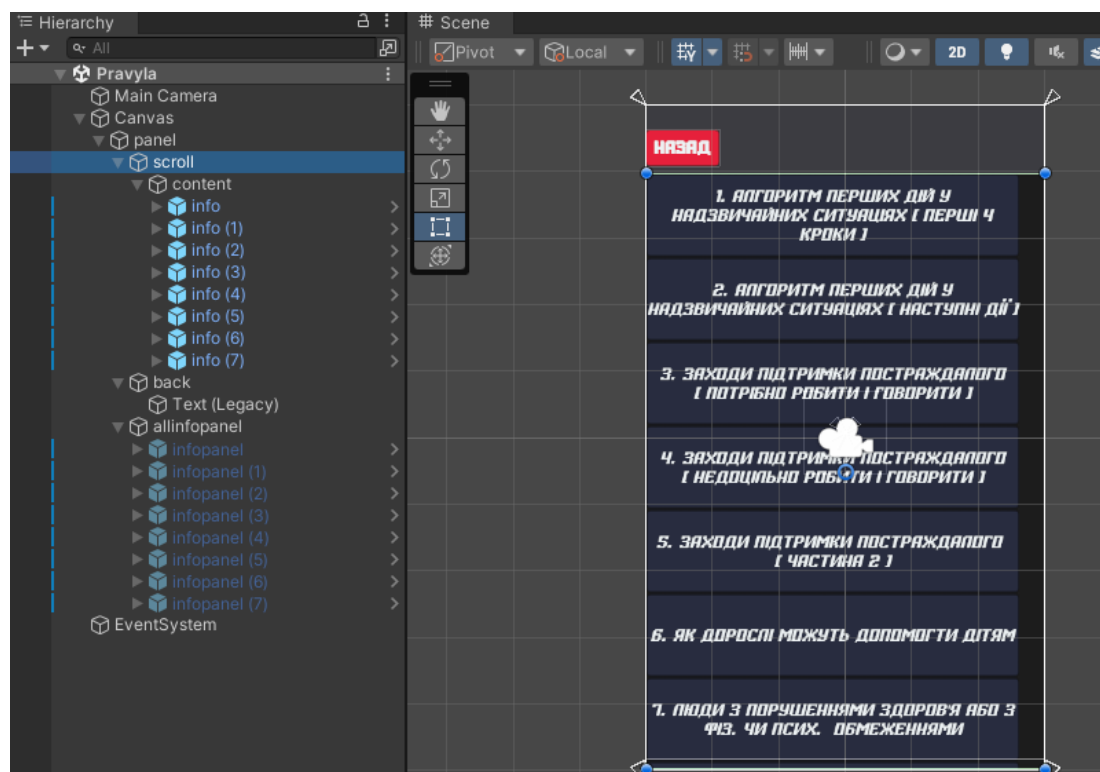


Рисунок 3.2.13 – Вигляд сцени «Дії у надзвичайних ситуаціях»

Ця форма має набір панелей, які дозволяють прокручувати вміст, коли клікнути на кожну з цих панелей, то відкривається вікно з відповідною інформацією.

Щоб створити такий упорядкований ряд панелей на головному сайті, нам потрібно правильно їх розташувати. Для досягнення цієї мети використовуються два основних об'єкти: «прокручування» і «контент».

Перший об'єкт, «Scroll», містить два основні елементи: «Scroll React» і «Mask». «Scroll React» відповідає за можливість прокручувати вміст, а «Mask» відповідає за обмеження області прокручування.

Другий об'єкт, «Вміст», включає «Групу вертикального макету» та «Фільтр розміру вмісту». «Vertical Layout Group» відповідає за вертикальне розташування елементів в контейнері, а «Content Size Filter» - за фільтрацію розміру вмісту.

На рисунках 3.2.14 і 3.2.15 показано зовнішній вигляд цих елементів для кращого розуміння їх функціональності.

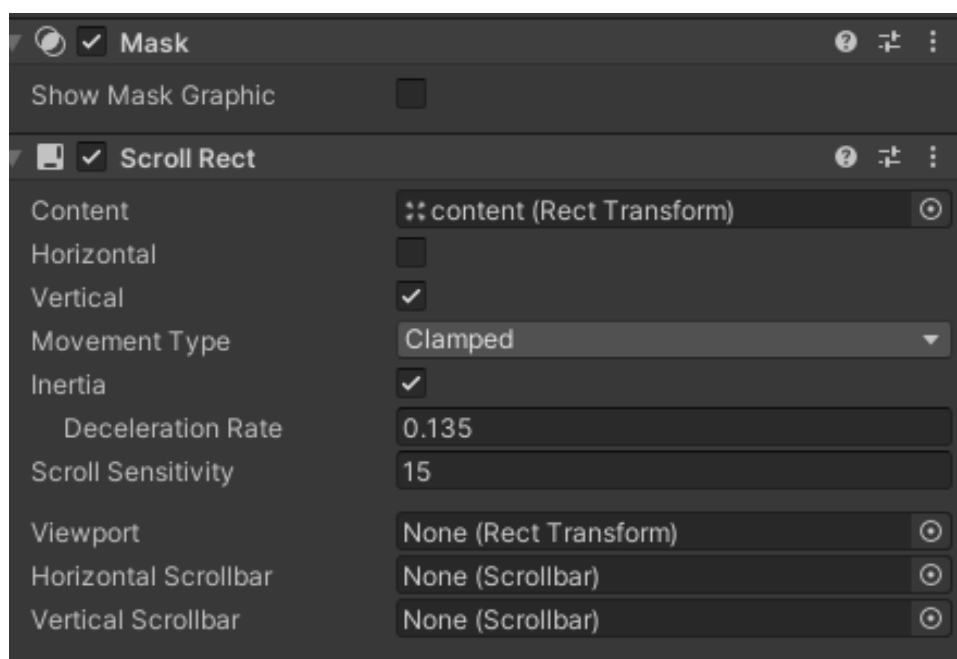


Рисунок 3.2.14 – Вигляд елементів об'єкта «Scroll»

У елементі Content потрібно помістити нашу панель з контентом, котра також має назву «Content».

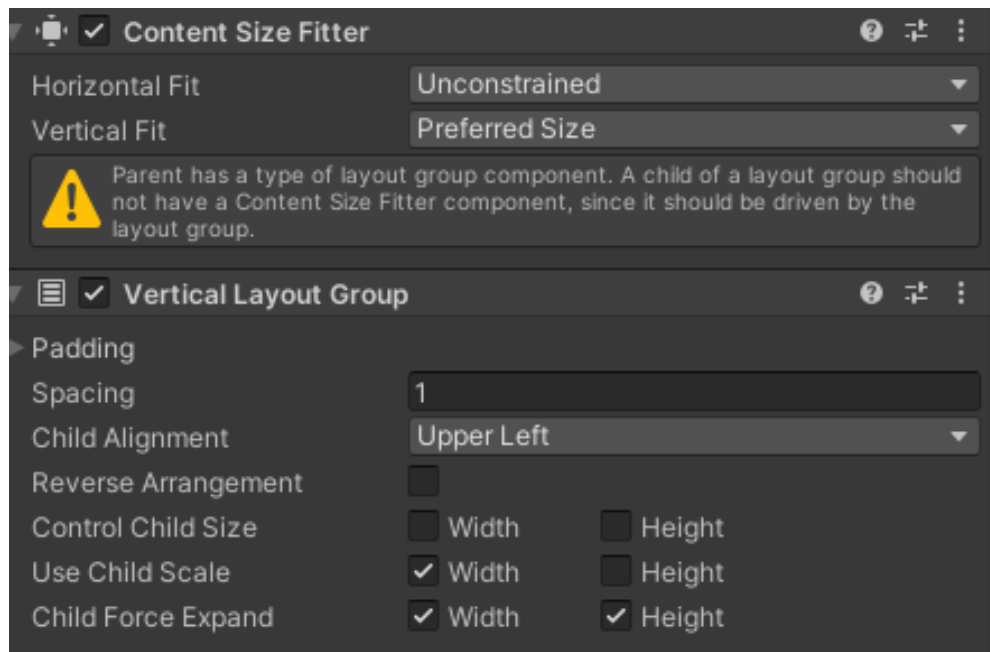


Рисунок 3.2.15 - Вигляд елементів об'єкта «Content»

Схожим чином створюється і панель для скролу інформації. Вигляд такої панелі зображено на рисунку 3.2.16.

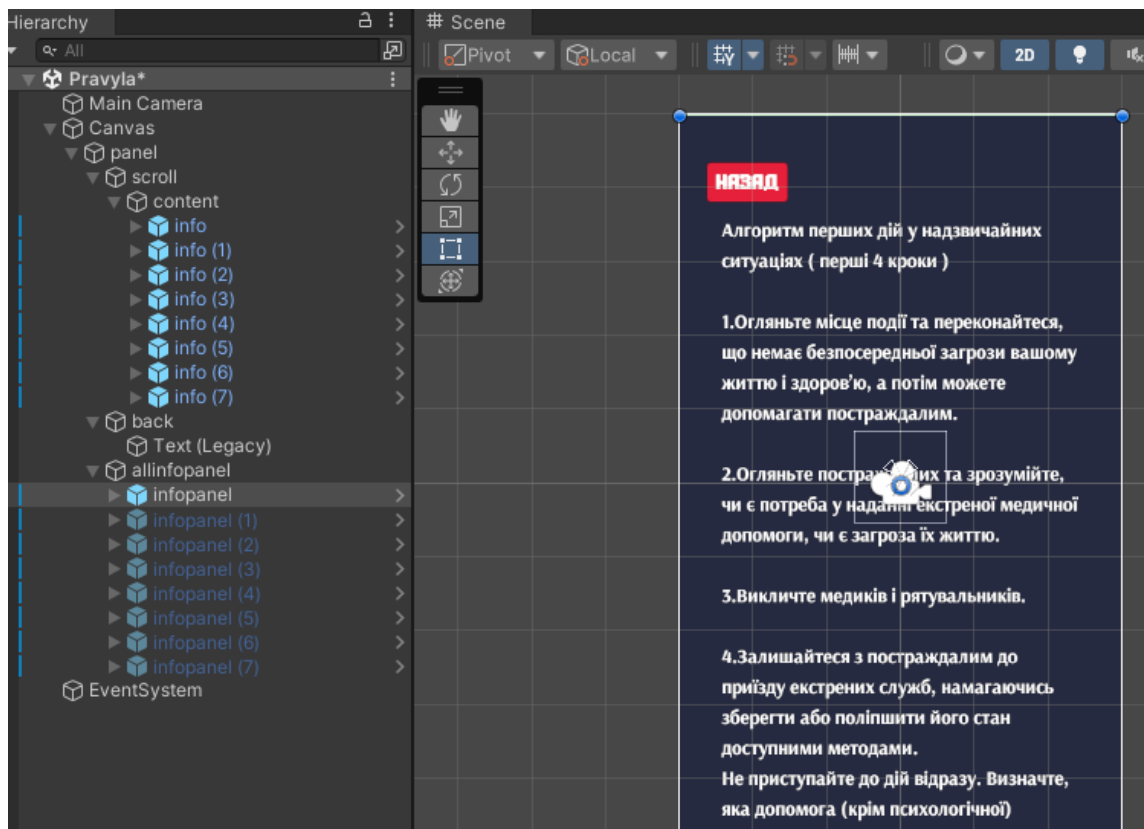


Рисунок 3.2.16 – Вигляд панелі з інформацією

Сцена «ілюстрації» була створена таким самим чином, що і сцена «Дії у надзвичайних ситуаціях». Вигляд сцени «ілюстрації» зображено на рисунках 3.2.17 – 3.2.18.

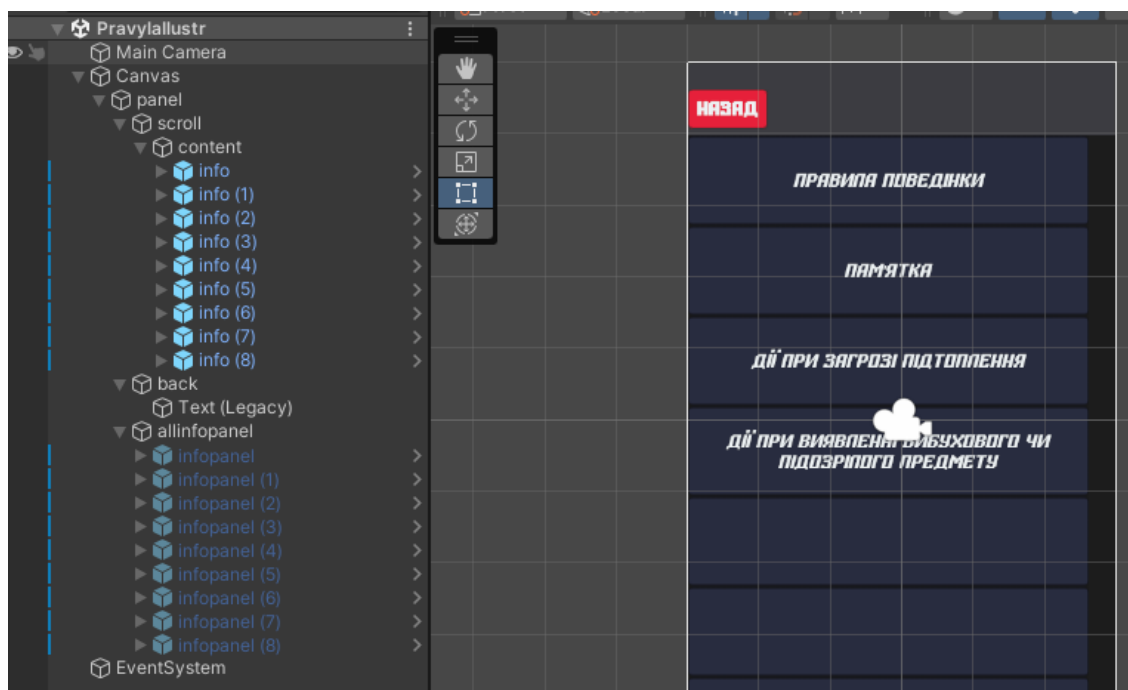


Рисунок 3.2.17 – Вигляд сцени форми «ілюстрації» №1

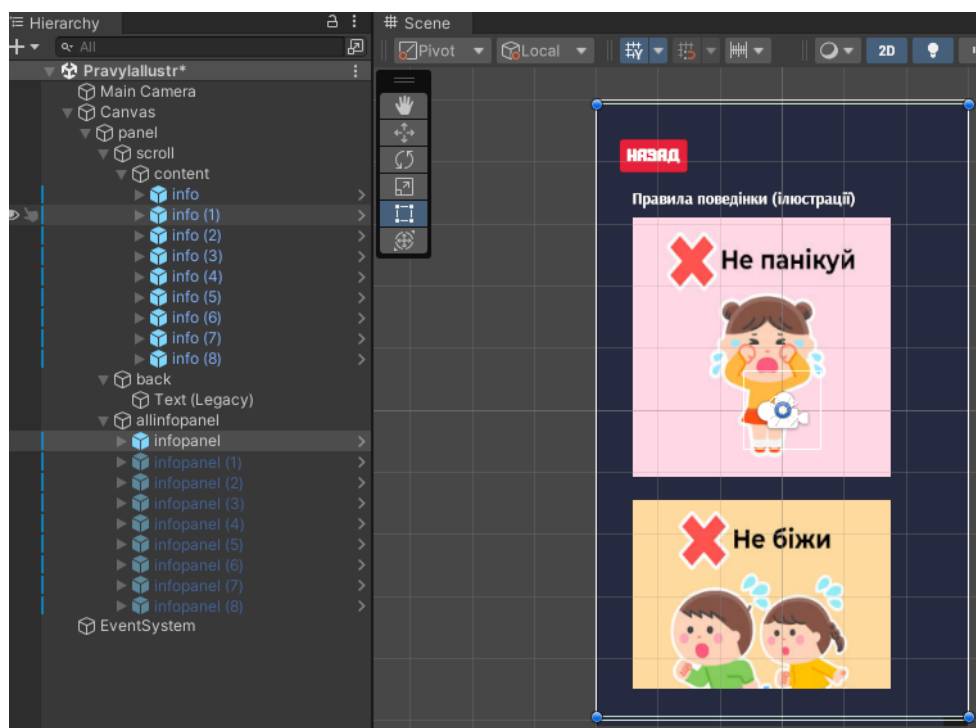


Рисунок 3.2.18 – Вигляд сцени форми «ілюстрації» №2

Остання сцена, сцена форми «Тест». Вигляд даної сцени зображено на рисунку 3.2.19.

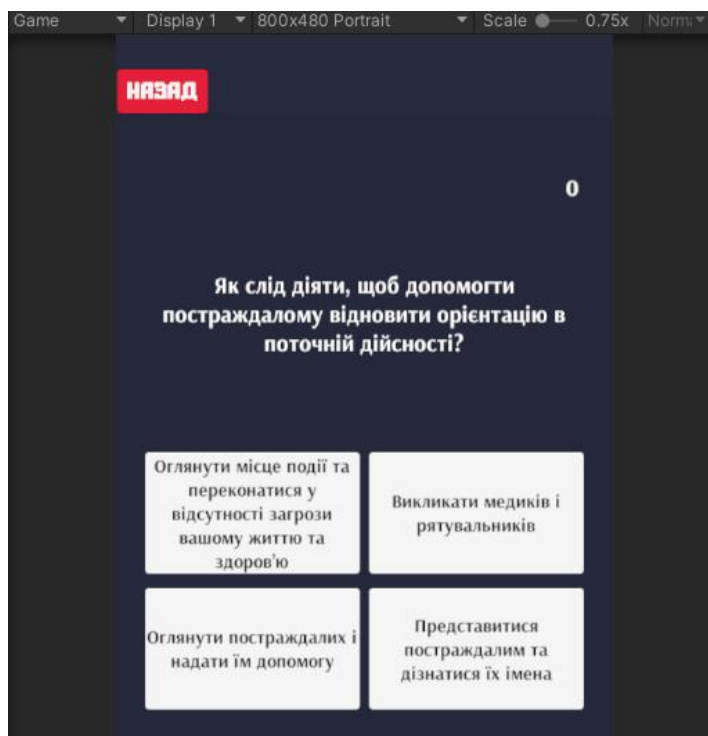


Рисунок 3.2.19 – Вигляд форми сцени «Тест»

Ієрархія компонентів та детальне їх розміщення сцени для форми тесту зображено на рисунку 3.2.20.

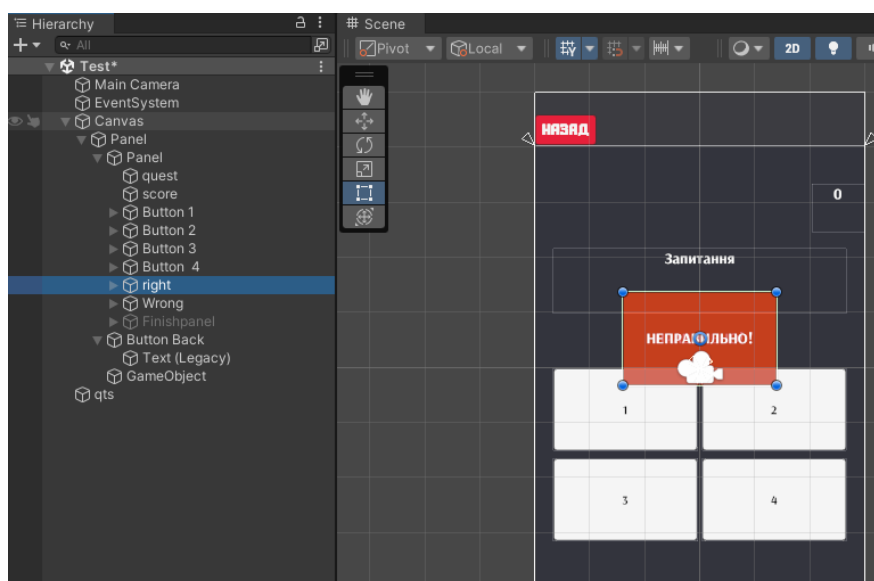


Рисунок 3.2.20 – Вигляд сцени для форми «Тест»

Для того щоб зберігати питання і варіанти відповідей на один з компонентів сцени додається код котрий створює об'єкт ScriptableObject. Вигляд даного C# коду зображено на рисунку 3.2.21.

```
using UnityEngine;

[CreateAssetMenu(fileName = "New QuestionData", menuName = "QuestionData")]
public class QtsData : ScriptableObject
{
    [System.Serializable]
    public struct Question
    {
        public string questionText;
        public string[] replies;
        public int correctReplyIndex;
    }

    public Question[] questions;
}
```

Рисунок 3.2.21 – C# код «QtsData»

Цей код створює шаблон для об'єкта даних, який може зберігати кілька питань вікторини.

Кожне питання складається з тексту питання, набору можливих відповідей і індексу правильної відповіді, що дозволяє системі легко перевіряти правильність відповідей користувачів.

Завдяки атрибуту [CreateAssetMenu], нові об'єкти QtsData можна створювати безпосередньо в редакторі Unity, що значно спрощує процес управління та організації питань вікторини. Це забезпечує гнучкість і зручність при додаванні, редагуванні або видаленні питань, а також робить роботу з вікторинами більш ефективною та зрозумілою для розробників.

Вигляд такого шаблону зображено на рисунку 3.2.22, що демонструє його структуру і розміщення полів для введення даних у редакторі Unity.

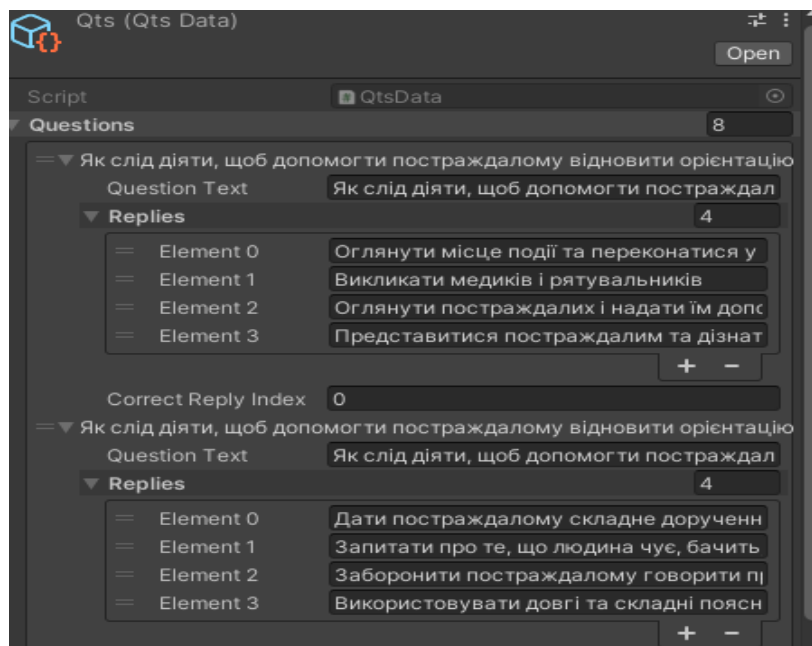


Рисунок 3.2.22 – Шаблон для зберігання даних для форми «Тест»

Щоб створити зв'язок між шаблоном даних, елементами форми та додати логіку до сцени, додається додатковий код C# до одного з елементів. Цей код відповідає за обробку взаємодії користувача з формою, включаючи оновлення даних на основі введення користувача та синхронізацію стану форми з даними. Вигляд такого коду, розміщеного на відповідному компоненті, показано на рисунку 3.2.23.

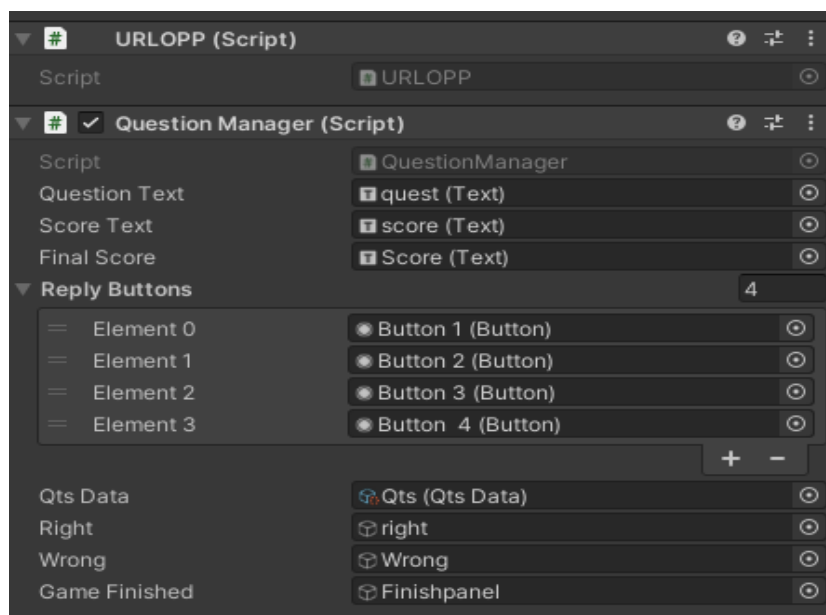


Рисунок 3.2.23 – Вигляд елементів коду «Question Manager» на компоненті «Panel»

На цьому зображенні можна побачити, як саме код інтегрується з компонентом форми, які методи та властивості використовуються для забезпечення необхідної функціональності та які події обробляються для забезпечення динамічної взаємодії з користувачем. Завдяки цьому коду сцена отримує логіку, необхідну для правильного функціонування та взаємодії з користувачем, що дозволяє створити інтуїтивно зрозумілий і зручний інтерфейс.

Повний C# код «Question Manager» знаходиться у Додатку А

3.3 Тестування застосунку та оцінка якості

Тестування відіграє важливу роль у визначенні та забезпеченні якості програмного застосунку і забезпечення в цілому. Така перевірка якості є невід'ємною частиною розробки програмного забезпечення, оскільки дозволяє виявити помилки, недоліки та небажану поведінку програми ще до її випуску.

Тестування застосунку відіграє вирішальну роль у забезпеченні якості та надійності продукції. Він допомагає виявляти та усувати помилки, забезпечує відповідність функціональним і користувацьким вимогам, забезпечує високу стабільність і продуктивність програми.

Функціональне тестування - це тип тестування, метою якого є перевірка функціональності системи відповідно до встановлених вимог і очікувань. Його мета - перевірити, чи програма виконує свої функції та операції відповідно до очікувань користувачів і клієнтів [5].

Під час функціональних тестів перевіряються різні аспекти програмного забезпечення, включаючи правильність обробки даних, адаптацію виконання функцій і операцій до специфікацій і вимог, а також реакцію на вхідні дані та різні сценарії використання.

Метою функціональної перевірки є забезпечення якості програмного забезпечення шляхом виявлення дефектів, помилок і неточностей у роботі

програмного забезпечення до його впровадження та використання. Відповідність програми встановленим вимогам і очікуванням є ключовим аспектом функціонального тестування.

Тестовий приклад - це мінімальний компонент тесту, який зазвичай націлений на один елемент тестового об'єкта. Чим менше тестовий приклад охоплює функціональність, тим очевиднішим буде пошук причини у разі виявлення помилки. Якщо тестовий приклад містить повний сценарій з десятків операцій, стає важко пройти його та визначити причину невдачі, якщо результат негативний. Існує також висока ймовірність отримання хибно негативного результату через помилку на іншому вузлі, вторинному для цього тестового випадку, який бере участь у цьому сценарії. Однак у деяких випадках тестові випадки навмисно роблять об'ємними і складними, якщо цього вимагає специфікація об'єкта тестування.

Функціональні тест-кейси виглядаються наступним чином:

1) Summary: Позитивний сценарій. Перевірка відкриття головного меню після запуску мобільного застосунку.

Preconditions:

- Застосунок встановлено на мобільний пристрій.
- Мобільний пристрій увімкнено і має доступ до інтернету.

Steps to reproduce:

- Відкрити мобільний застосунок.
- Дочекатися завантаження застосунку.

Expected result:

Відкриється головне меню застосунку з доступними функціями (перегляд інформації, карти тривоги, ілюстрації, інтерактивні тести).

2) Summary: Позитивний сценарій. Перевірка перегляду карти тривоги після натискання відповідної кнопки в головному меню.

Preconditions:

- Користувач знаходиться в головному меню застосунку.

- Мобільний пристрій має доступ до інтернету.

Steps to reproduce:

- Відкрити мобільний застосунок.
- Натиснути кнопку "Карти тривоги" у головному меню.

Expected result:

Відкриється екран з картою тривоги, що показує актуальні надзвичайні ситуації.

3) Summary: Позитивний сценарій. Перевірка проходження інтерактивного тесту після натискання відповідної кнопки в головному меню.

Preconditions:

- Користувач знаходиться в головному меню застосунку.
- Мобільний пристрій має доступ до інтернету.

Steps to reproduce:

- Відкрити мобільний застосунок.
- Натиснути кнопку "Інтерактивні тести" у головному меню.
- Вибрати доступний тест і натиснути кнопку "Розпочати тест".

Expected result:

Відкриється інтерфейс для проходження тесту, працююча можливість відповідати на питання і отримати результати після завершення тесту.

4) Summary: Позитивний сценарій. Перевірка доступу до інформації про дії в надзвичайних ситуаціях після натискання відповідної кнопки в головному меню.

Preconditions:

- Користувач знаходиться в головному меню застосунку.
- Мобільний пристрій має доступ до інтернету.

Steps to reproduce:

- Відкрити мобільний застосунок.
- Натиснути кнопку "Дії у надзвичайних ситуаціях" у головному меню.
- Вибрати категорію надзвичайної ситуації.

Expected result:

Відкриється екран з детальною інформацією та рекомендаціями щодо дій в обраній надзвичайній ситуації, включаючи ілюстрації та покрокові інструкції.

Під час функціонального тестування мобільного застосунку для надзвичайних ситуацій були ретельно перевірені всі основні функції та можливості застосунку. У процесі тестування було забезпечено правильність виконання кожної операції та відповідність результатів очікуваним значенням.

Тест включав перевірку правильності обробки даних, належного функціонування інтерфейсу користувача, точності відображення карт сигналізації, коректності інтерактивних тестів та задовільної роботи з ілюстраціями. Перевірено, що програма здатна виконувати свої основні функції відповідно до встановлених вимог і очікувань користувачів.

Завдяки проведеним функціональним тестам було підтверджено, що мобільний застосунок працює стабільно та відповідає всім необхідним вимогам. Це сприяє підвищенню її якості та надійності, що важливо для забезпечення безпеки та ефективності використання програми в екстрених ситуаціях.

Отримані результати функціональних тестів свідчать про готовність програмного забезпечення до наступних етапів тестування та впровадження. Функціональні тести пройшли успішно і підтвердили, що мобільний застосунок відповідає всім вимогам і може забезпечити функціональність, необхідну користувачам в екстрених ситуаціях.

Для подальшого підтвердження якості та надійності тестування мобільного застосунку було проведено ручне тестування функцій програми. Цей етап тестування включав реалізацію реальних сценаріїв використання програмного забезпечення, запис результатів і документування їх скріншотами. Нижче наведено процес ручного тестування та результати для кожної ключової функції програми.

Для початку перевіримо чи головне меню програми відкривається належним чином після запуску застосунку. Кроки для виконання:

- 1) Відкрити мобільний застосунок.
- 2) Перевірити, чи відображається головне меню з основними функціями

(інструкції, карти тривоги, тестування).

В результаті головне меню відкривається належним чином, всі основні функції доступні для вибору. Вигляд активного головного меню зображено на рисунку 3.3.1

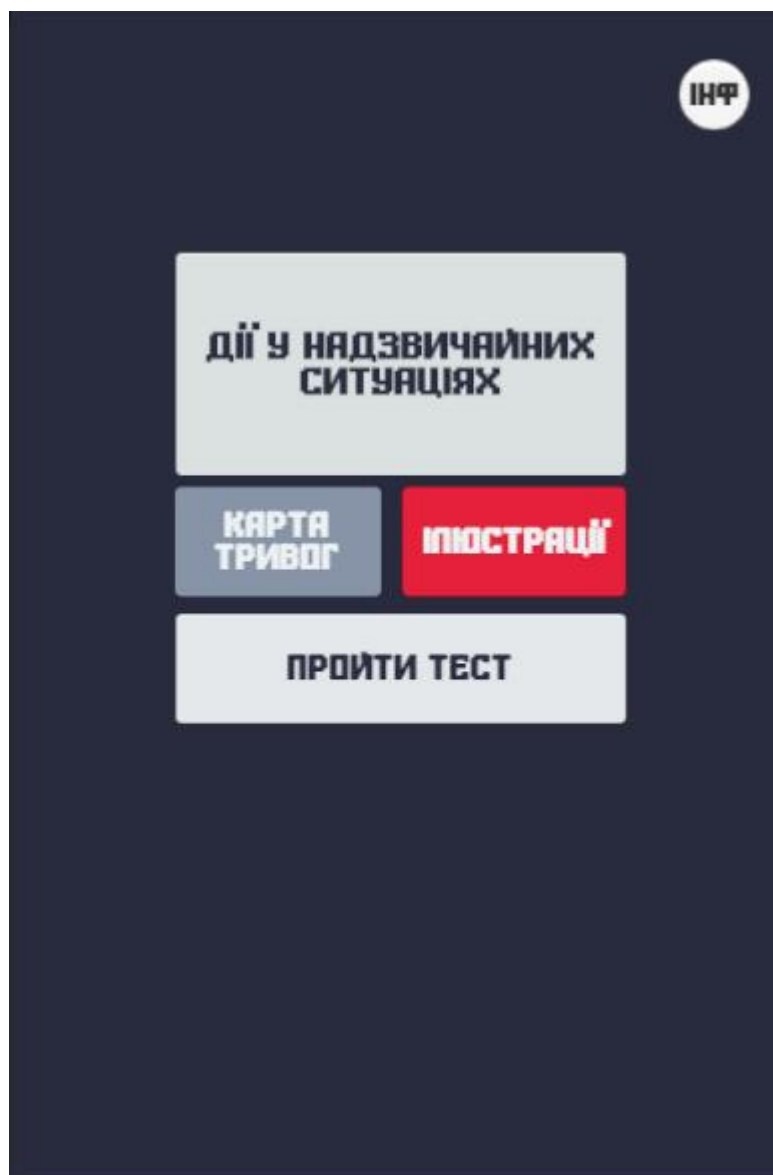


Рисунок 3.3.1 – Активне головне меню

Далі Перевіримо, чи інструкції щодо дій у надзвичайних ситуаціях відображаються коректно. Кроки для виконання:

- 1) Відкрити мобільний застосунок.
- 2) Вибрати розділ "Інструкції".

3) Переглянути кілька інструкцій щодо різних типів надзвичайних ситуацій.

В результаті інструкції відображаються чітко, текст читається легко, ілюстрації відповідають змісту інструкцій. Вигляд працюючих інструкцій зображено на рисунку 3.3.2-3.3.3.

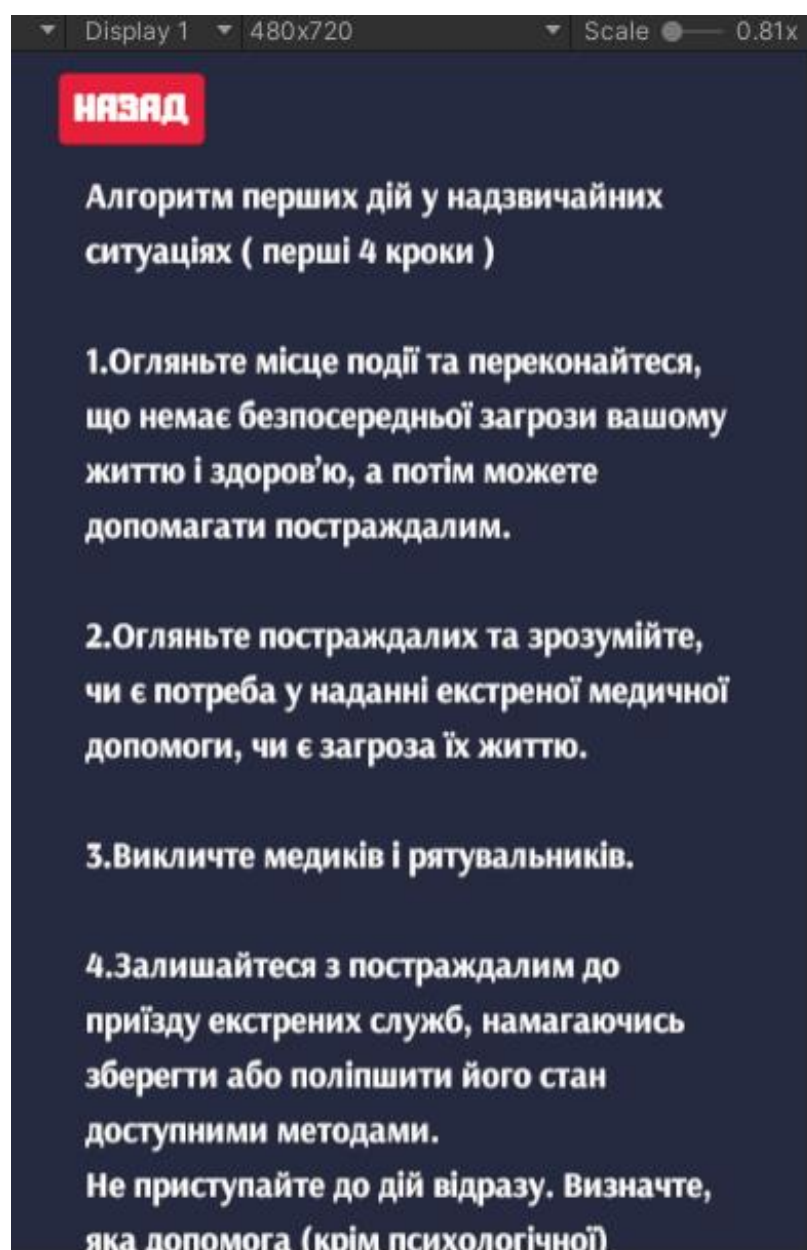


Рисунок 3.3.2 – Вигляд інструкції «Алгоритм перших дій»

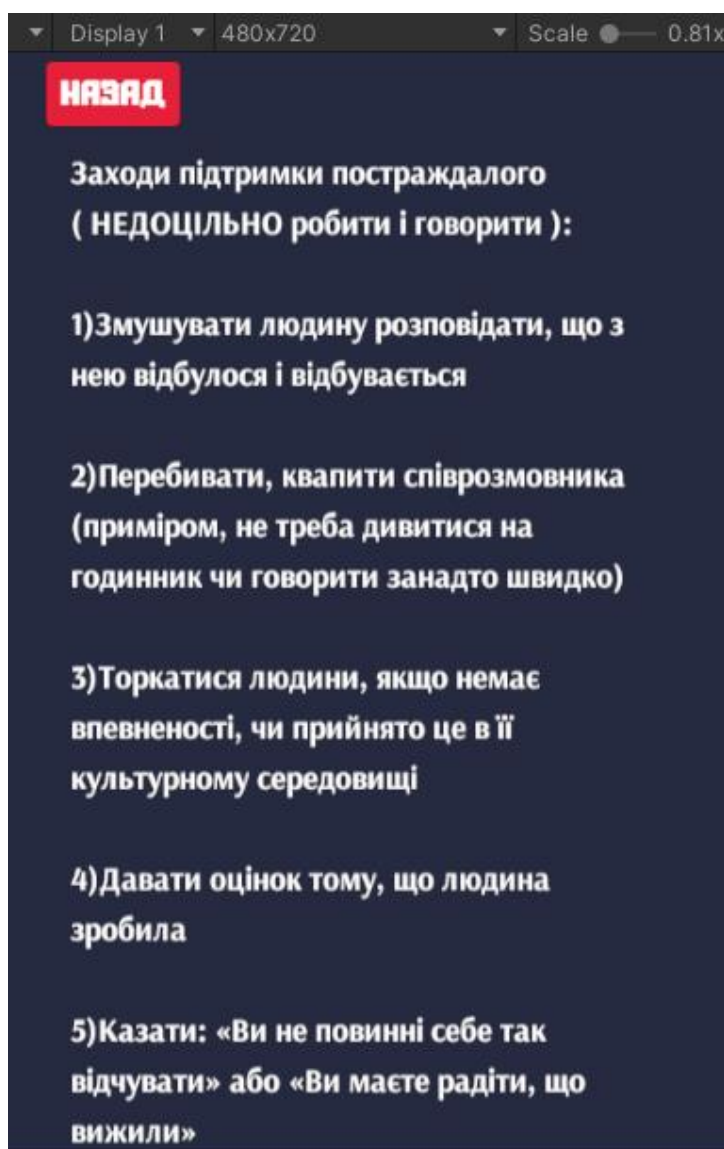


Рисунок 3.3.3 – Вигляд інструкції «заходи підтримки постраждалих»

Наступне перевірка функціональності інтерактивних тестів для оцінки знань користувача. Кроки для виконання:

- 1) Відкрити мобільний застосунок.
- 2) Вибрати розділ "Тестування".
- 3) Пройти один із тестів, відповідаючи на запитання.

У результаті тест проходиться без проблем, результати відображаються правильно, а оцінки відповідають наданим відповідям.

Вигляд початкового вікна проходження тесту та його результатів зображено на рисунку 3.3.4-3.3.5.



Рисунок 3.3.4 – Головне вікно форми «Тест»

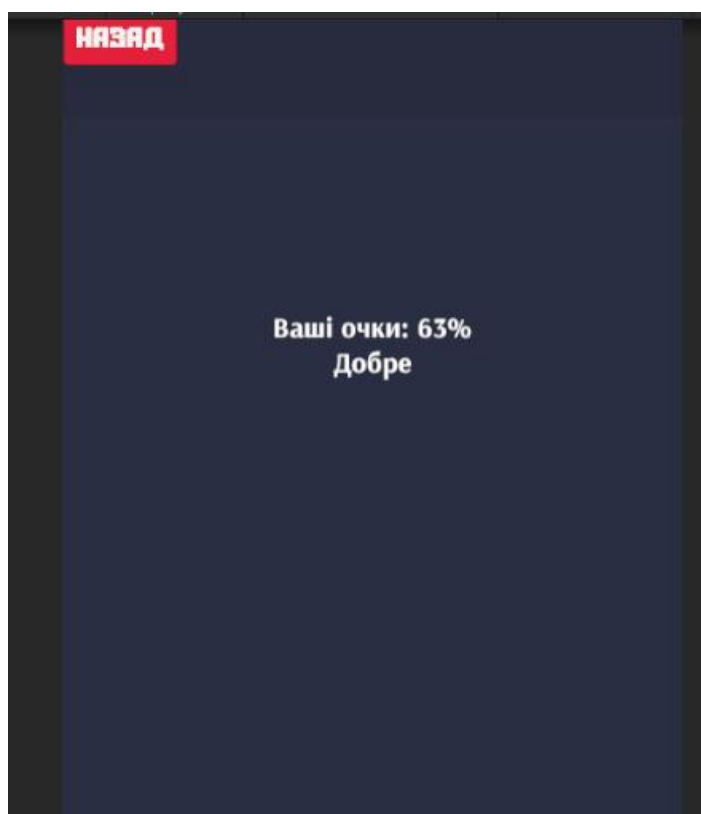


Рисунок 3.3.5 – Результати проходження тесту

На кінець. Перевірка коректності відображення карти тривог після переходу по відповідній кнопці форми «Карта тривог». Кроки для виконання:

- 1) Відкрити мобільний застосунок.
- 2) Вибрати розділ "Карта тривог".
- 3) Переглянути активні тривоги та інформацію про них.

В результаті карта тривог відображається коректно, інформація про тривоги оновлюється в реальному часі, користувач може взаємодіяти з картою для отримання додаткової інформації.

Головне вікно форми «карта тривог» зображено на рисунку 3.3.6, а вигляд відкритої карти тривог яку відкрив застосунок зображено на рисунку 3.3.7.

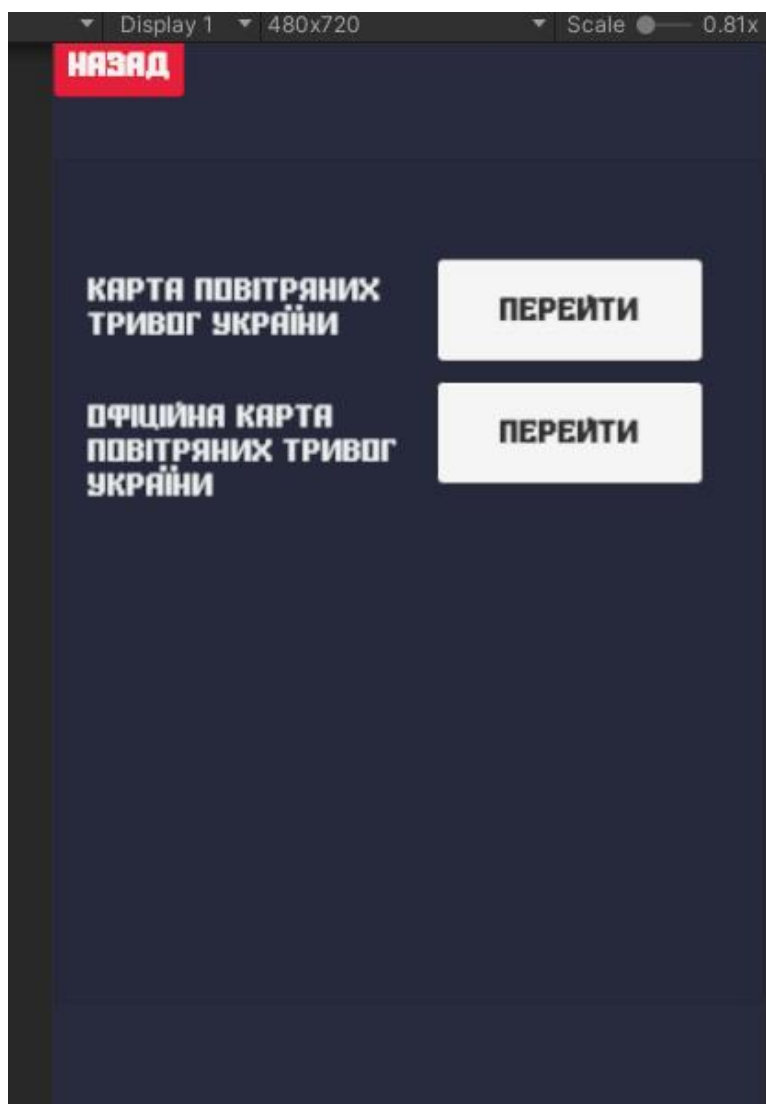


Рисунок 3.3.6 – Головне вікно форми «Карта тривог»

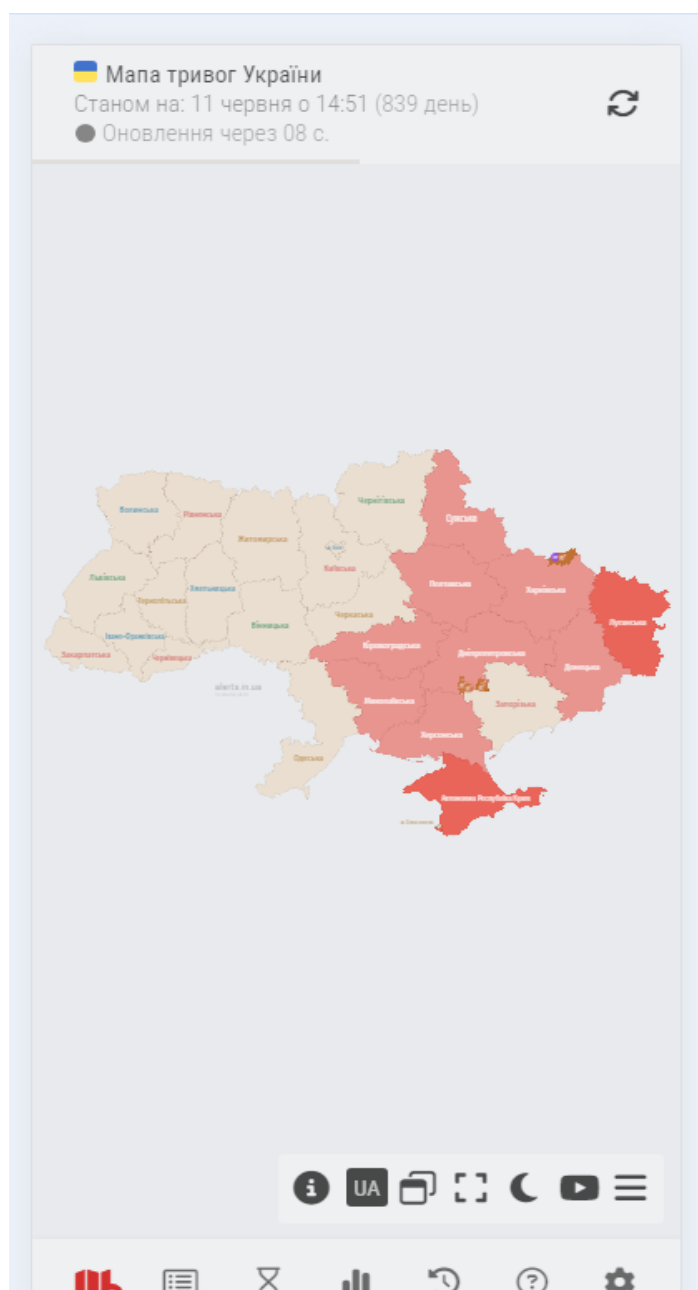


Рисунок 3.3.7 – Карта тривог

Ручне тестування мобільного застосунку для дій у надзвичайних ситуаціях підтвердило, що всі основні функції застосунка працюють належним чином. Проведені тестові випадки, а результати, задокументовані скріншотами, свідчать про високу якість і надійність програми. Додаткові кроки включають більш детальні тести в різних сценаріях використання та застосування програми для широкого кола користувачів.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Соціально-політичні небезпеки, їхні види та характеристики

Безпека життєдіяльності є однією з найважливіших складових сучасного суспільства, адже її метою є забезпечення збереження життя і здоров'я людей у всіх сферах їхньої діяльності.

У сучасному світі, де глобалізація та урбанізація набирають обертів, соціально-політичні небезпеки стають все більш актуальними. Ці небезпеки можуть мати різні форми та впливати на життя мільйонів людей у всьому світі. Серед них політичні конфлікти, соціальна нестабільність, тероризм та інші явища, які можуть спричинити значні людські та матеріальні втрати.

Під час написання кваліфікаційної роботи на здобуття ступеня бакалавра на тему «Розробка застосунку для дій у надзвичайних ситуаціях» особлива увага приділяється вивченню соціально-політичних небезпек та їх впливу на суспільство. Розуміння типів і характеристик цих небезпек має вирішальне значення для створення ефективного програмного продукту, який надаватиме користувачам інформацію та інструкції, необхідні для дій у критичних ситуаціях. Тому аналіз соціально-політичних небезпек є невід'ємною частиною цієї роботи.

Соціально-політичні небезпеки виникають через конфлікти в суспільстві, де стикаються різні інтереси та сили. Ці конфлікти можуть мати політичний, соціальний чи економічний характер і мати серйозні наслідки для стабільності та добробуту суспільства. Політичні конфлікти виникають між політичними системами, партіями чи урядами, часто через боротьбу за владу чи різні підходи до управління в країні. Вони можуть призвести до дестабілізації політичної системи, зміни режиму чи навіть громадянської війни.

Соціальні конфлікти виникають між різними соціальними групами, наприклад між етнічними, релігійними чи культурними спільнотами, а також між класами чи професійними групами. Такі конфлікти часто є результатом нерівного

доступу до ресурсів, дискримінації або боротьби за соціальну справедливість. Вони можуть проявлятися у формі демонстрацій, бунтів чи масових заворушень, що може призвести до значних соціальних змін або, навпаки, ще більшої соціальної напруги.

Коли соціальний конфлікт стає масштабним, він стає соціально-політичним конфліктом, в якому беруть участь політичні інститути та організації. Такі конфлікти можуть зачіпати різні сфери життя суспільства, включаючи міжнаціональні, класові, міжетнічні, релігійні та інші відносини. Відомими прикладами є міжетнічні конфлікти в колишньому Радянському Союзі та Югославії.

До соціально-політичних конфліктів можна віднести громадянські війни, страйки та війни між країнами. Суб'єкти таких конфліктів, усвідомлюючи протиріччя, обирають боротьбу як спосіб їх вирішення. Це можуть бути окремі особи, групи або цілі країни.

Війна – це збройне протистояння між державами (або їх коаліціями), соціальними, етнічними та іншими групами. Війна є крайнім проявом політичної боротьби. Під час військових конфліктів гинуть мільйони людей, руйнуються міста та селища, завдається величезна шкода навколишньому середовищу .

Війни часто спричиняють найбільшу кількість жертв з політичних причин. Наприклад, під час Другої світової війни в Радянському Союзі (1945-1941) загинуло приблизно 55 мільйонів людей, повністю зруйновано 1710 міст і 70 тисяч сіл. Під час війни у В'єтнамі 1960-х років загинуло близько 7 мільйонів місцевих жителів і 57 тисяч американців. Крім людських втрат і широкомасштабних руйнувань, військові конфлікти завдають серйозної шкоди навколишньому середовищу [5].

Тероризм є формою політичного екстремізму і часто використовується для досягнення політичних або релігійних цілей. Тероризм включає напади на

державні установи, захоплення заручників, вибухи та масові вбивства. Основними видами тероризму є політичний, релігійний і кримінальний .

Найбільш поширеними терористичними актами є [5]:

- Напади на державні або промислові об'єкти;
- Захоплення державних установ або посольств;
- Захоплення літаків або інших транспортних засобів;
- Викрадення або насильницькі дії проти жертви;
- Політичні вбивства;
- Вибухи та масові вбивства, розраховані на залякування;
- Розповсюдження інфекційних хвороб.

Міжнародна спільнота веде активну боротьбу з тероризмом. Генеральна Асамблея ООН щороку обговорює питання запобігання тероризму, приймає декларації та створює комітети для вирішення цієї проблеми.

Серед найбільших викликів сучасного суспільства можна відзначити проблему алкоголізму. Алкогольна залежність стала серйозною хворобою, від якої страждають мільйони людей у всьому світі. Лише за офіційною статистикою в Україні зареєстровано понад 690 тисяч випадків захворювання. Практично кожен орган в організмі людини відчуває на собі негативний вплив алкоголю. Перш за все, алкоголь викликає серйозні проблеми з печінкою, які можуть призвести до цирозу та інших захворювань. Крім того, він також може негативно впливати на роботу нервової системи, викликаючи депресію та інші психічні розлади [6].

Тривале вживання алкоголю може призвести до фізичної залежності, що ускладнює процес відмови від алкоголю. Люди, які страждають на алкоголізм, часто відчувають труднощі зі здоров'ям і соціальною адаптацією. Це може вплинути на їхню роботу та особисте життя, а також на стосунки з родиною та іншими.

Вживання алкоголю також має негативні наслідки для суспільства в цілому. Це призводить до зростання ДТП, домашнього насильства та злочинності. Крім

того, алкоголь може впливати на продуктивність праці та економічний розвиток країни, знижуючи працездатність і збільшуючи витрати на медичне обслуговування.

Для боротьби з цією проблемою необхідно вживати комплексних заходів, таких як підвищення обізнаності населення про наслідки вживання алкоголю, підтримка програм реабілітації алкогольних залежних, проведення правильної політики щодо регулювання виробництва та продажу алкогольних напоїв. Лише спільними зусиллями громади та держави можна досягти значного зменшення випадків алкоголізму та його негативних наслідків для суспільства.

Куріння – ще одна серйозна проблема. В Україні курить значна частина населення, що призводить до високої смертності та захворювань, пов'язаних з палінням. Тютюновий дим містить багато токсичних речовин, які шкідливо впливають на здоров'я.

Таким чином, соціально-політичні небезпеки становлять серйозну загрозу суспільству, і їх вирішення потребує спільних зусиль як національних урядів, так і міжнародної спільноти.

4.2 Обґрунтування вибору фарбування стін, підлоги, ферм, колон, вантажопідйомних кранів, верстатів та іншого обладнання цеху

У сучасному виробничому середовищі, де безпека праці та оптимальні умови праці мають особливе значення, вибір кольорів для фарбування різних елементів цеху стає не лише питанням естетики, але й центральним елементом у створенні безпечної та комфортної робочої зони .

Вибір кольорів для фарбування стін і підлоги визначається не тільки дизайнерськими перевагами, але і функціональними вимогами. Світлі та нейтральні відтінки, такі як білий, світло-сірий або бежевий, використовуються для

забезпечення візуального комфорту співробітників. Вони роблять приміщення більш просторим і світлим, а також дозволяють краще виявляти забруднення і дефекти. Відповідно до стандарту ДСТУ 7245:2011 «Дизайн і ергономіка», вибір світлих кольорів сприяє зниженню рівня стресу та втоми працівників, що важливо для підвищення їх продуктивності та ефективності [7].

Фарбування ферм і колон відіграє центральну роль у забезпеченні безпеки та організації робочого простору. Яскраві кольори, такі як червоний та жовтий, використовуються для позначення небезпечних зон та евакуаційних виходів відповідно до стандарту ДСТУ EN ISO 3864-4:2010 «Графічні символи. Кольори та знаки безпеки.». Це допомагає прискорити реагування працівників у надзвичайній ситуації та знизити ризик травм. Крім того, правильно підібрані кольори для фарбування небезпечних зон забезпечують визначеність і зрозумілість інструкцій персоналу в надзвичайних ситуаціях [8].

Вибір кольорів для фарбування кранів має вирішальне значення для забезпечення безпеки експлуатації та запобігання потенційним небезпекам. Відповідно до стандарту червоний і жовтий кольори використовуються для позначення небезпечних зон та надання інструкцій щодо безпечної експлуатації кранів. Це забезпечує максимальну видимість компонентів сигналу та допомагає запобігти можливим нещасним випадкам і травматичним ситуаціям. Вибір правильних кольорів також враховує різноманітність освітлення та погоду, що може гарантувати ефективну роботу кранів за будь-яких обставин.

Освітлення та маркування обладнання та механізмів має важливе значення для забезпечення безпечної та ефективної роботи працівників. Кольорове кодування машин та іншого обладнання допомагає встановити їх функціональне призначення, а також вказує на потенційну небезпеку та вимоги безпеки. Відповідно до ДСТУ EN ISO 12100:2016 «Безпека машин. Загальні принципи проектування. Оцінка та зменшення ризиків», правильне фарбування обладнання може визначити небезпечні зони, режими роботи та необхідні заходи безпеки для персоналу [9].

Стіни там, де встановлені лазерні пристрої, мають бути виготовлені з матеріалу, який не пропускає лазерне випромінювання. Щодо внутрішніх поверхонь, їх рекомендується фарбувати у матовий колір з мінімальним коефіцієнтом відбивання, що сприяє максимальному розсіюванню світла залежно від довжини хвилі лазерного випромінювання. Побілку стін і стель приміщень термічних цехів рекомендується робити регулярно, не рідше одного разу на рік [10].

Вибір кольорів для елементів фарбувального цеху базується на стандартах безпеки та ергономіки, які визначають необхідність використання певних кольорів для позначення небезпечних зон, виходів та інших ключових елементів. Світлі та нейтральні кольори використовуються для стін і підлоги для підвищення комфорту та безпеки працівників, тоді як світлі кольори використовуються для позначення безпеки та виходів. Це дозволяє створити чітке та безпечне робоче середовище, що допомагає зменшити ризик травм і покращити загальну ситуацію безпеки на робочому місці.

Ретельний аналіз та обґрунтування вибору кольорів для фарбування різних елементів цеху є невід'ємною частиною підвищення безпеки та комфорту при роботі в промислових умовах. Це завдання вимагає не тільки знань у сфері ергономіки та безпеки, а й практичного бачення потреб робочого середовища.

Сучасні стандарти безпеки та ергономіки враховують роль кольорів у виробничому середовищі. Правильно підібрані кольори можуть значно полегшити сприйняття робочого середовища та зробити його безпечнішим для співробітників. Наприклад, яскраві кольори часто використовуються для позначення небезпечних зон або обладнання, які потребують особливої уваги. Це допомагає працівникам легше розрізняти різні елементи середовища та уникати потенційно небезпечних ситуацій.

Крім того, вибір кольорів може вплинути на емоційний стан співробітників і їх загальну продуктивність. Наприклад, використання приємних і спокійних кольорів може створити більш комфортну атмосферу в робочому середовищі, що допомагає знизити рівень стресу та підвищити ефективність роботи.

Одним із ключових аспектів у виборі кольорів є їх відповідність вимогам безпеки та нормам. Наприклад, стандарти можуть визначати певні кольори для позначення різних типів легкозаймистих речовин або для визначення різних типів небезпеки. Дотримання цих стандартів допомагає створити безпечніше робоче середовище та зменшити ризик нещасних випадків.

Узгодження колірних схем також може бути важливим аспектом при організації промислового простору. Наприклад, використання однакової колірної палітри для всіх елементів навколишнього середовища може допомогти створити однорідний і згуртований вигляд цеху, що сприяє загальному враженню професіоналізму та обізнаності про безпеку серед працівників.

Тому усвідомлений вибір кольорів для фарбування різних елементів є центральним кроком у створенні безпечного, комфортного та продуктивного робочого середовища. Правильно підібрані кольори сприяють не тільки підвищенню безпеки.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досягнуто основної мети - розробка мобільного застосунку, призначеного для надання важливої інформації та рекомендацій користувачам щодо дій у надзвичайних ситуаціях. Проаналізовано та реалізовано всі необхідні функціональні можливості, що забезпечують ефективну підтримку користувачів у разі надзвичайних подій.

Розроблено зручний та інтуїтивно зрозумілий інтерфейс користувача, що включає компоненти, які дозволяють швидко орієнтуватися у застосунку. Застосовано підхід до розробки інтерфейсу, який забезпечує зручність використання на мобільних пристроях. Реалізовано функціональні можливості, серед яких перегляд інформації про дії у надзвичайних ситуаціях, відкриття карт тривоги та географічних даних для орієнтування у просторі, перегляд ілюстрацій, що демонструють дії у критичних ситуаціях, а також проходження інтерактивних тестів для перевірки знань та готовності до дій у надзвичайних ситуаціях.

Забезпечено актуальність та достовірність інформації шляхом інтеграції з різними джерелами інформації, що дозволило отримувати актуальні дані та рекомендації. Враховано потреби сучасних користувачів щодо швидкого доступу до важливої інформації. Для розробки застосунку було обрано платформу Unity та мову програмування C#, що забезпечило високу ефективність розробки.

Таким чином, виконання цієї кваліфікаційної роботи підтвердило доцільність використання сучасних інструментів та технологій для розробки мобільних застосунків, призначених для інформування та підтримки користувачів у надзвичайних ситуаціях. Результати роботи можуть бути використані як основа для подальших досліджень та розробок у цій сфері.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. alerts.in.ua - онлайн сервіс [Електронний ресурс] - Режим доступу:
<https://uk.wikipedia.org/wiki/Alerts.in.ua>
2. Draw io – онлайн-сервіс для створення діаграм [Електронний ресурс] –
Режим доступу: <https://vchymo.com/application/Drawio>
3. Unity 3d – движок [Електронний ресурс] - Режим доступу:
<https://foxminded.ua/dvyzhok-unity/>
4. Функціональне тестування [Електронний ресурс] – Режим доступу до
ресурсу: <https://qalight.ua/baza-znaniy/funktsionalne-testuvannya/>
5. Зеркалов В. М. БЖД та ОП: Навчальний посібник / В. М. Зеркалов. – Київ,
2016. – 268 с.
6. Мелех Л. В. БЖД та ОП: Навчальний посібник / Л. В. Мелех. – Львів, 2022.
– 220 с.
7. ДСТУ EN ISO 3864-4:2010. Графічні символи. Кольори та знаки безпеки
[Електронний ресурс] - Режим доступу:
https://online.budstandart.com/ua/catalog/doc-page?id_doc=55060.
8. ДСТУ 7245:2011. Дизайн і ергономіка. Кодування зорової інформації.
Загальні вимоги ергономіки [Електронний ресурс] - Режим доступу:
https://online.budstandart.com/ua/catalog/doc-page?id_doc=60262.
9. ДСТУ EN ISO 12100:2016. Безпечність машин. Загальні принципи
проектування. Оцінювання ризиків та зменшення ризиків [Електронний ресурс] -
Режим доступу: https://online.budstandart.com/ua/catalog/doc-page?id_doc=71627.
10. Про затвердження Правил охорони праці при термічній обробці металів
Держгірпромнагляд; Наказ, Правила від 18.12.2007 № 315 [Електронний ресурс] -
Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0066-08#Text>

ДОДАТКИ

ДОДАТОК А – Лістинг коду розробленого застосунку

Лістинг коду 1 – Код відкривання URL-адреси

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class URLOPP : MonoBehaviour
{
    public void Url1Open (string url)
    {
        Application.OpenURL(url);
    }
    public void Url2Open (string url)
    {
        Application.OpenURL(url);    }    }
```

Лістинг коду 2 – Код перемикання між сценами

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class SwapScene : MonoBehaviour
{
    public int sceneNumber;
    public void Transition()
    {
        SceneManager.LoadScene(sceneNumber);
    }
}
```

Лістинг коду 3 – Код для зберігання даних тесту

```
using UnityEngine;

[CreateAssetMenu(fileName = "New QuestionData", menuName = "QuestionData")]
public class QtsData : ScriptableObject
{
    [System.Serializable]
    public struct Question
    {
        public string questionText;
        public string[] replies;
        public int correctReplyIndex;
    }
    public Question[] questions;
}
```

Лістинг коду 4 – Код для керування питаннями для тесту

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class QuestionManager : MonoBehaviour
{
    public Text questionText;
    public Text scoreText;
    public Text finalScore;
    public Button[] replyButtons;
    public QtsData qtsData;
    public GameObject right;
    public GameObject wrong;
    public GameObject gameFinished;
```

```

private int currentQuestion = 0;
private static int score = 0;

void Start()
{
    SetQuestion(currentQuestion);
    right.SetActive(false);
    wrong.SetActive(false);
    gameFinished.SetActive(false);
}

void SetQuestion(int questionIndex)
{
    questionText.text =
qtsData.questions[questionIndex].questionText;
    foreach (Button r in replyButtons)
    {
        r.onClick.RemoveAllListeners();
    }
    for (int i = 0; i < replyButtons.Length; i++)
    {
        replyButtons[i].GetComponentInChildren<Text>().text =
qtsData.questions[questionIndex].replies[i];
        int replyIndex = i; // Fix: Assign i to replyIndex
        replyButtons[i].onClick.AddListener(() =>
        {
            CheckReply(replyIndex);
        });
    }
}

void CheckReply(int replyIndex)
{
    if (replyIndex ==
qtsData.questions[currentQuestion].correctReplyIndex) // Fix:
Correct the comparison operator
    {

```



```

        score++;
        scoreText.text = "" + score;
        right.SetActive(true);
        foreach (Button r in replyButtons)
        {
            r.interactable = false;
        }
        StartCoroutine(Next());
    }
    else
    {
        wrong.SetActive(true);
        foreach (Button r in replyButtons)
        {
            r.interactable = false;
        }
        StartCoroutine(Next());
    }
}

IEnumerator Next()
{
    yield return new WaitForSeconds(2);
    currentQuestion++;

    if (currentQuestion < qtsData.questions.Length) // Fix:
Correct Length spelling
    {
        Reset();
    }
    else
    {
        gameFinished.SetActive(true);
        float scorePercentage = (float)score /
qtsData.questions.Length * 100;
        finalScore.text = "Ваші очки: " +
scorePercentage.ToString("F0") + "%";
    }
}

```

```

        if (scorePercentage < 50)
        {
            finalScore.text += "\nЦього разу не дуже";
        }
        else if (scorePercentage < 60)
        {
            finalScore.text += "\nСпробуйте ще";
        }
        else if (scorePercentage < 70)
        {
            finalScore.text += "\nДобре";
        }
        else if (scorePercentage < 80)
        {
            finalScore.text += "\nБездоганно!";
        }
        else
        {
            finalScore.text += "\nХороший результат!";
        }
    }
}

public void Reset()
{
    right.SetActive(false);
    wrong.SetActive(false);

    foreach (Button r in replyButtons)
    {
        r.interactable = true;
    }

    SetQuestion(currentQuestion);
}
}

```

ДОДАТОК Б – Диск із розробленим застосунком