

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

ПОЯСНЮВАЛЬНА ЗАПИСКА
до кваліфікаційної випускної роботи

освітній ступінь бакалавр

спеціальність 126 „Інформаційні системи та технології”
(шифр і назва спеціальності)

спеціалізація „Інформаційні системи та технології”
(назва спеціалізації)

на тему „Розробка інтелектуальної гри з використанням Unity/C”

Виконав: студент групи ІСТ-206д _____ А.А Ніценко
(підпис) (ініціали і прізвище)

Керівник _____ Д.В Ратов
(підпис) (ініціали і прізвище)

Завідувач кафедри _____ О.І Захожай
(підпис) (ініціали і прізвище)

Рецензент О.І Захожай

Київ – 2024

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

Освітній ступінь бакалавр

спеціальність 126 „Інформаційні системи та технології”
(шифр і назва спеціальності)

спеціалізація 126 „Інформаційні системи та технології”
(назва спеціалізації)

ЗАТВЕРДЖУЮ

Завідувач кафедри ПМ,

д.т.н., доцент

_____ Захожай О.І
“___” _____ 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ ВИПУСКНУ РОБОТУ СТУДЕНТУ

Ніценко Анна Андріївна

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтелектуальної гри з використанням Unity/C

Керівник роботи _____ Ратов Денис Валентинович, к.т.н, доцент

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

затверджений наказом університету від “06” травня 2024 року №171/15.15-С

2. Строк подання студентом роботи 10.06.2024

3. Вихідні дані до роботи Об'єктом даної роботи є процес розробки інтелектуального ігрового застосунку з використанням логічних операцій на базі ігрового рушія Unity.

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ. Аналітичний огляд, з висвітленням наступних питань: характеристика Unity Engine та Unreal Engine, їх відмінності, модель, основні класи. Основна частина в якій треба висвітлити: створення та реалізація проекту. Висновки. Перелік використаних джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 30 березня 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання кваліфікаційної випускної роботи	Строк виконання етапів	Примітка
1	Одержання завдання на виконання роботи	30.03.24	виконано
2	Укладання і погодження з керівником плану і етапів виконання роботи	06.04.24	виконано
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	14.04.24	виконано
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	22.04.24	виконано
5	Укладання та тестування програмного продукту	28.04.24	виконано
6	Укладання, оформлення та погодження пояснювальної записки з керівником	02.05.24	виконано
7	Здача готової пояснювальної записки на кафедрі	10.06.24	виконано
8	Укладання доповіді і презентації	12.06.24	виконано

Студент _____
(підпис)

А.А Ніценко _____
(ініціали і прізвище)

Керівник роботи _____
(підпис)

Д.В Ратов _____
(ініціали і прізвище)

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
дипломної роботи студента гр. ІСТ-20бд Ніценко А.А.

Науковий керівник

Доцент, к.т.н.

Ратов Д.В.

Оцінка наукового керівника: _____

Рецензент

ПІБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК

підпис

Меняйленко О.С

РЕФЕРАТ

Робота містить: 49 сторінок основного тексту, 22 сторінок додатків, 10 рисунків, 10 використаних джерел, 2 таблиці.

Метою випускної кваліфікаційної роботи є вивчення особливостей розробки інтелектуального ігрового застосунку на основі кросплатформеного інструменту Unity Engine за допомогою мови програмування C# та його ігрового рушія.

В ході розробки ігрового застосунку були проаналізовані відмінності Unity Engine та Unreal Engine , визначені існуючі види інструментів для розробки кросплатформених ігор , ознайомилися з середою розробки, була створена гра , було проведено тестування гри. Вироблено опис процесу розробки і тестування системи. Реалізовано, а також описаний користувальницький інтерфейс, зроблені знімки екранних форм програмного засобу. Продемонстровано результат виконаної роботи.

Система задовольняє всім вимогам, пред'явленим в технічному завданні.

Зміст

ВСТУП	7
1. АНАЛІТИЧНИЙ ОГЛЯД	8
1.1 Класифікація ігрових застосунків	8
1.3 Відмінності Unity та Unreal Engine при розробці ігор	9
1.4 Мова програмування C#.....	12
1.5 Технічне завдання на розробку гри	12
2. ПРЕДМЕТНА ГАЛУЗЬ. ВИДИ ТА АНАЛІЗ ІГОР.ПРОБЛЕМИ ТА РИЗИКИ	14
2.1 Поняття ігор.....	14
2.2 Існуючі види ігор на основі Unity	14
2.3 Аналіз ігор	15
2.4 Проблеми та ризики які можуть виникнути	16
3. СТВОРЕННЯ ТА ПРОЕКТУВАННЯ ГРИ ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS.....	18
3.1 Встановлення програмного забезпечення Unity та UnityHub	18
3.2 Реалізація та створення проекту.....	20
ВИСНОВКИ.....	26
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	27
ДОДАТОК А.....	28

ВСТУП

Актуальність. Індустрія розробки комп'ютерних ігор – це новий напрямок та маловивчений. Застосування цифрових технологій та поширення комп'ютерних ігор серед молодшого покоління дозволяють зрозуміти чи цікава сфера програмування саме для них . Комп'ютерні ігри сприяють самостійному вивченню, мотивують їх на вдосконалення навичок та комунікативного спілкування. Завдяки особливостям ігрових засобів відображення дійсності, дитина у грі вперше долучається до абстрактного мислення та розвиває свої інтелектуальні здібності та свою інтуїцію.

Об'єкт дослідження: застосунок на мові програмування C# з застосуванням ігрової бібліотеки Unity.

Предмет дослідження: ігровий додаток інтелектуальної комп'ютерної гри на Unity.

Мета дослідження: створення інтелектуальної гри.

Задачі дослідження:

1. Розробка графічного візуального інтерфейсу для ігрового застосунку;
2. Реалізація програмного модулю гри;
3. Тестування гри та її об'єктів.

1. АНАЛІТИЧНИЙ ОГЛЯД

1.1 Класифікація ігрових застосунків

У сучасному світі комп'ютерні ігри є однією з найпопулярніших розваг. Ринок ігрових застосунків розвивається з неймовірною швидкістю, і щороку з'являються нові ігри та жанри. Для того щоб чіткіше визначитися в цьому розмаїтті, існує класифікація ігрових застосунків. У таблиці 1.1 наведено опис жанрів ігрових застосунків.[1]

Таблиця 1.1 – Класифікація ігрових застосунків

Жанр	Опис
Стратегії	Передбачає планування та керування різними ресурсами з метою розвитку, будівництва та керування власними територіями та арміями, а також стратегічне планування бойових дій з метою перемоги над противником.
Головоломки	Передбачає вирішення складних логічних завдань та головоломок, які вимагають від гравця терпіння та кмітливості.
Шутери	Передбачає геймплей, зосереджений на використанні різноманітної зброї та стрільбі на ворогів або інших цілей.

1.2 Аналіз інструментальних засобів розробки

Як середовище розробки було обрано платформу Unity. В ході порівняння вона була обрана, як найкраща для нашого дипломного проекту. Було проаналізовано такі середовища розробки :

- Одна із найпопулярніших платформ для створення ігор – Unity. Вона з'явилася у 2005 році і досі не втрачає попиту серед гейм – розробників. Основна причина, чому програмний комплекс управління так довго тримається на плаву – це величезна спільнота і безліч офіційних і неофіційних відео уроків, які допомагають створювати ігри та вирішувати багато проблем.
- Ще один лідер у рейтингу ігрових двигунів. Unreal Engine – це платформа, орієнтована на 3D, із системою візуального програмування. Unreal Engine використовує спеціальну мову сценаріїв під назвою Blueprint, який дозволяє створювати візуальний скриптинг без написання коду. Крім того, Unreal Engine підтримує C++ для більш просунутого програмування та оптимізації, що робить його кращим вибором для розробників із досвідом роботи з C++.
- CryEngine дозволяє створювати ігри із фотореалістичною графікою. При належне вміння проекти, розроблені з його допомогою, перевищують по якості картинки будь-які ігри. CryEngine відомий своїми винятковими можливостями у сфері графіки. [2]

1.3 Відмінності Unity та Unreal Engine при розробці ігор

Ігровий двигун – це основний набір програмних компонентів і візуальних інструментів, який дозволяє створювати і запускати інтерактивні додатки з графічним забезпеченням, яке обробляється в реальному часі.

Використання ігрових двигунів значно спрощує розробку ігор шляхом надання готових інструментів та ресурсів, що зменшує необхідність у написанні коду з нуля і прискорює процес розробки.

Вони забезпечують широкі можливості для реалізації творчих ідей розробників, сприяють підвищенню якості графіки, фізики, звуку та інших аспектів гри, що забезпечує більш затягуючий геймплей та задоволення користувачів. Загалом, можна зробити висновок, що обраний вами ігровий движок повинен дати вам можливість виконувати вищезазначені завдання з меншими зусиллями на кодування.

Unity та Unreal Engine – великі імена у розробці ігор. Вони обидва вільні у використанні, але є також інші. Наприклад, Godot – безкоштовна платформа для розробки ігор з відкритим кодом. Головним мінусом Godot можна назвати малу порівняно з Unity кількість документації та відео уроків.

Unity – це середовище для розробки комп'ютерних ігор, в якій об'єднані різні програмні засоби, що використовуються при створенні ПЗ - текстовий редактор, компілятор, відладчик і так далі. При цьому, завдяки зручності використання, Unity робить створення ігор максимально простим і комфортним, а мультиплатформеність движка дозволяє розробникам ігор охопити якомога більшу кількість ігрових платформ і операційних систем.

Unity найчастіше використовують для розробки мобільних ігор на IOS, Android та Windows, а Unreal Engine 4 для розробки ігор для таких платформ, як PlayStation, MacOS та Xbox. Зустрічається думка, що найлегше починати розвиватися у сфері GameDev шляхом вивчення такого способу взаємодії із середовищем розробки як Visual Scripting.

Для створення складного проекту потрібні глибокі знання C#. Unity доступний безкоштовно, що відкриває перед незалежними розробниками двері в ігрову індустрію. [3]

Також можна зазначити про можливість використання базової версії програми на безкоштовній основі. Не зважаючи на те що вона не надає доступ до всіх функцій інших версій вона дозволяє розробникам – початківцям використовувати усі потужності для створення гри, обмежуючи їх тільки деякими мережевими функціями. Всього існує три версії основного рушія – безкоштовна Personal, а також Pro і Plus. Усі вони надають доступ до

використання офіційного магазину як платних, так і вільних у використанні асетів і їх використання без додаткових виплат.

Unreal Engine – це популярний ігровий рушій, розроблений Epic Games, який надає набір потужних інструментів для створення та розробки ігор на широкому спектрі платформ, включаючи ПК, консолі та мобільні пристрої. Він широко визнаний як один з найпотужніших і найгнучкіших ігрових рушіїв на сьогоднішній день, що пропонує розробникам можливість створювати високоякісні, захоплюючі ігри з приголомшливою графікою і реалістичною фізикою. Рушій також пропонує підтримку передових фізичних симуляцій, анімації та візуальних ефектів, що дає розробникам можливість створювати високодеталізоване та реалістичне ігрове середовище.

Загалом, Unreal Engine справив значний вплив на ігрову індустрію, надавши розробникам потужний та гнучкий інструментарій для створення інноваційного та захопливого ігрового досвіду. Вивчивши його особливості, можливості та проблеми, ви зможете глибше зрозуміти роль, яку відіграють ігрові програми в середовищі Unreal Engine у формуванні ігор. [4]

Також можна проаналізувати ці два засоби розробки у вигляді таблиці.

Таблиця 1.3 – Порівняння засобів розробки

Критерії	Unity	Unreal Engine
Доступність	Безкоштовно	Безкоштовно
Набір інструментів	Безліч	Безліч
Спільнота (доступність вивчення та відео – уроків)	Багато матеріалів	Багато матеріалів
Мова програмування	C#	C++

1.4 Мова програмування C#

C# – об'єктно – орієнтована мова програмування. Розроблена в 1998-2001 роках групою інженерів компанії Microsoft під керівництвом Андерса Хейлсберга і Скотта Вільтаумота як мова розробки додатків для платформи Microsoft .NET Framework. Згодом був стандартизований як ECMA-334 та ISO/IEC 23270.

Переїнявши багато від своїх попередників – мов C++, Delphi, Модула, Smalltalk і, особливо, Java – C#, спираючись на практику їх використання, виключає деякі моделі, що зарекомендували себе проблематичні при розробці програмних систем, наприклад, C# на відміну від C++ не підтримує множинне спадкування класів (між тим допускається множинна реалізація інтерфейсів).

Вона є найбільш популярною між безлічі інших мов та на цій мові можна швидко навчитися складати різні програми.

C# відноситься до сім'ї мов з C-подібним синтаксисом, їх синтаксис найбільш близький до C++ і Java. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (у тому числі операторів явного та неявно-го приведення типу), делегати, атрибути, події, змінні, властивості, узагальнені типи та методи, ітератори, анонімні функції з підтримкою замикань, LINQ, винятки, коментарі у форматі XML. [5]

1.5 Технічне завдання на розробку гри

В результаті виконання даного дипломного проекту повинна бути створена інтелектуальна гра на основі ігрового рушія Unity та мови програмування C#. Гра повинна бути цікава та розвивати логіку, ерудицію та інтуїцію у користувачів та особливо дітей. Гра призначена для застосування в операційній системі Windows але може бути використана і в інших операційних системах.

При розробці гри для операційної системи Windows потрібно вирішити наступні завдання:

- Проаналізувати відмінності Unity та Unreal Engine при розробці інтелектуальної гри

- Визначити існуючі види ігор
- Провести дослідження яка гра може зацікавити користувача
- Ознайомитися з середою розробки
- Реалізація ігрового проекту для операційної системи Windows
- Тестування гри

2. ПРЕДМЕТНА ГАЛУЗЬ. ВИДИ ТА АНАЛІЗ ІГОР. ПРОБЛЕМИ ТА РИЗИКИ

2.1 Поняття ігор

Комп'ютерна гра – це тип відеоігор, в які грають на персональному комп'ютері (ПК), а не на ігровій консолі чи аркадному автоматі. Це обумовлено рядом причин, досліджених у дипломній роботі. Гра, як і будь – який програмний продукт проходить певні визначені етапи від задумки до готового продукту та має свої особливості щодо створення та реалізації.

Люди грають в ігри не стільки для самої гри, скільки для досвіду, який гра створює: захоплюючий прилив адреналіну, загадкові пригоди, психологічний виклик. Люди грають в ігри, щоб створити короточасні емоції та переживання, чи подолають вони складні ігрові виклики, шукаючи позбавлення від щоденних турбот.

Нові технології – від соціальних медіа та систем GPS до штучного інтелекту – роблять планету, на якій ми живемо, абсолютно новою і створюють новий цифровий світ.

Ігрова індустрія багатогранна та відкрита для нових ідей, технологій та досягнень. На сьогоднішній день ігри – це не тільки спосіб цікаво провести час, це яскраві емоції, враження, незабутній досвід, це спільноти, що стрімко розвиваються, це професія і це спорт. Розробка ігор – це мистецтво. [6]

2.2 Існуючі види ігор на основі Unity

Одним із жанрів ігор, які можна створити за допомогою Unity, є аркада. В аркадах основний акцент робиться на активні дії гравця і швидкий геймплей. Завдяки функціоналу Unity розробники можуть реалізувати механіку, що дозволяє гравцеві керувати персонажем, набирати бали, проходити рівні та змагатися з іншими гравцями [7].

Ще одним жанром, який можна створити за допомогою Unity, є RPG (рольова гра). RPG характеризується наявністю розвитку персонажа, взаємодією з іншими персонажами та виконанням завдань. Unity дозволяє створювати складні системи розрахунку навичок та характеристик персонажів, а також створювати власні світи з унікальними особливостями.

2.3 Аналіз ігор

Зараз, у 2024 році, геймерам неймовірно пощастило мати широкий спектр платформ для їхніх улюблених ігор. Найпоширенішими є 3 основні ігрові платформи: мобільна, ПК та консольна.

Їх поширеність у відсотках можна відобразити у вигляді рисунку.

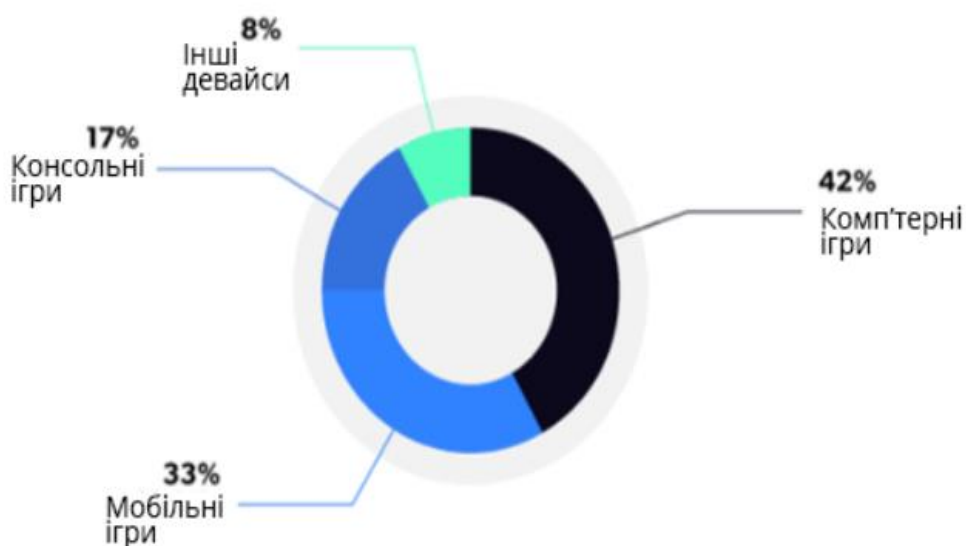


Рисунок 2.3 – Статистика популярності ігор

Мобільні ігри. Кількість користувачів смартфонів по всьому світу продовжує зростати і в 2020 році перевищила 3,5 мільярда. Усі ці люди мають доступ до світу ігор у своїй кишені і можуть грати в них практично скрізь, у будь – який час. Мова більше не йде лише про такі прості ігри, як Snake, незважаючи на те, що ця гра стала легендарною.

Більшість ігор на мобільних пристроях є коротшими і набагато менш графічно насиченими, ніж версія для ПК або консолі, але це не означає, що вони не можуть бути такими ж захоплюючими. [8]

В цілому, мобільні ігри є набагато більш портативним варіантом, ніж будь – яка інша платформа з іграми, які, як правило, набагато дешевші.

Комп'ютерні ігри. Багато геймерів твердо переконані, що ПК – це єдиний істинний спосіб грати в ігри. Звичайно, це не так, але ПК має досить вагомі підстави, щоб бути найкращою ігровою платформою в цілому.

Ігри на комп'ютері пропонують дуже високий рівень налаштування, що є рушійною силою його популярності. Геймери можуть підняти свій ігровий досвід на будь – який бажаний рівень. Можна почати роботу із стандартною установкою або заглибитися у світ будівництва комплектацій, налаштування та вдосконалення.

Консольні ігри. Консоль – це спеціалізований настільний комп'ютер, який використовується для відеоігор. Дві найпопулярніші консолі – це PlayStation від Sony та Xbox від Microsoft. Wii від Nintendo також є претендентами, які імітують фізичну участь у таких активностях, як боулінг та гра в теніс.

Відеоігрові консолі зазвичай працюють від операційних систем і процесорів, що відрізняються від настільних комп'ютерів. Консолі контролюються відповідними виробниками, а програмне забезпечення адаптоване до можливостей машини. Ігри не взаємозамінні з іншими ігровими приставками або настільними комп'ютерами, хоча розробники програмного забезпечення можуть розробляти ігри для кількох платформ. [9]

2.4 Проблеми та ризики які можуть виникнути

Так, розробка гри самостійно цілком можлива, і це може бути захоплюючим і творчим процесом. Сучасні технології та доступність

різноманітних інструментів дозволяють одному розробнику створити гру від ідеї до реалізації.

Однак варто зазначити, що розробка гри самостійно може бути викликом, особливо для новачків в індустрії.

Ось основні проблеми та ризики які можуть виникнути при розробці ігор:

- **Обсяг і складність проекту.** Якщо планується розробляти невеличку та просту гру, таку як проста головоломка чи аркада, то виконання проекту самостійно може бути досить реальним. Однак, якщо йдеться про більшу і складнішу гру з високими вимогами до графіки, звуку і механік, можливо, знадобиться більш чисельна команда розробників.

- **Навички та досвід.** Розробка ігор вимагає знань у різних галузях, як програмування, дизайн, графіка і звук. Якщо у вас уже є досвід або хороші навички в одній або декількох із цих галузей, це може значно полегшити процес.

- **Управління проектом.** Розробка гри може бути часозатратним процесом, особливо під час роботи наодинці. Важливо добре планувати й керувати проектом, щоб дотримуватися термінів і не перевантажувати себе.

- **Підтримка та оновлення.** Після завершення гри може знадобитися підтримка і випуск оновлень для виправлення помилок і додавання нового контенту. Переконайтеся, що ви готові забезпечити підтримку своєї гри після її випуску.

Самостійна розробка гри вимагає наполегливості, творчого підходу і готовності до вивчення нових технологій. Це захопливий досвід, який дає змогу вам повністю втілити свої ідеї в реальність і поділитися своєю творчою роботою зі світом.

3. СТВОРЕННЯ ТА ПРОЕКТУВАННЯ ГРИ ДЛЯ ОПЕРАЦІЙНОЇ СИСТЕМИ WINDOWS

3.1 Встановлення програмного забезпечення Unity та UnityHub

Для початку розробки необхідно було завантажити спеціальне програмне забезпечення. Насамперед, завантажити платформу Unity з офіційного сайту. [10]

Далі потрібно вибрати версію Unity. На сторінці завантаження потрібно вибрати операційну систему, на якій планується використовувати Unity. В рамках даного дипломного проекту будемо використовувати версію для Windows.

Може знадобитися вибрати модулі або плагіни, які відповідають потребам розробки ігор. Після завершення завантаження потрібно запустити інсталяційний файл Unity і слідувати інструкціям на екрані, щоб закінчити процес встановлення. Під час встановлення може знадобитися вибрати компоненти, які потрібно встановити, та вказати шлях встановлення. Після завершення інсталяції можна запускати Unity.

При першому запуску Unity може знадобитися увійти до свого облікового запису або створити новий.

Після проходження реєстрації облікового запису та встановлення Unity, можна розпочинати створення проекту в UnityHub.

UnityHub дозволяє знайти корисні статті, огляди, поради та приклади, які допоможуть розробникам освоїти різноманітні аспекти роботи з Unity. Тут можна поставити питання та отримати допомогу від досвідчених розробників, а також поділитися своїми проектами та ідеями.

UnityHub є майданчиком для обміну знаннями та досвідом у спільноті розробників Unity. Тут можна отримати підтримку, знайти натхнення і бути в курсі останніх новин та трендів у світі розробки ігор на платформі Unity.

Також UnityHub служить для створення та збереження ігрового проекту гри.

Після встановлення всього необхідного можна приступати до створення проекту в Unity.

Для цього у попередньо встановленому UnityHub потрібно натиснути кнопку "New project" для створення нового проекту. Далі необхідно вибрати, скільки просторів буде в грі, що розробляється, і написати ім'я проекту, як представлено на рисунку 3.1.

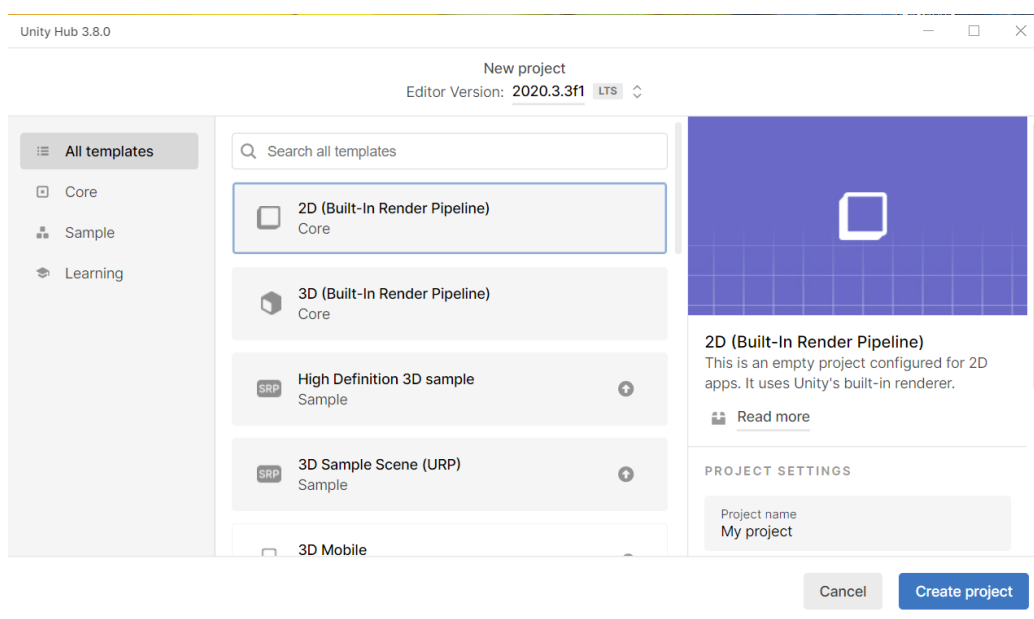


Рисунок 3.1 – Створення проекту гри

Після натискання на кнопку "Create project" створиться порожній проект Unity.

Після створення порожнього проекту необхідно підключити Microsoft Visual Studio для роботи зі скриптами у проекті. Що б це зробити потрібно зайти в меню "Edit", а потім у "Project settings".

3.2 Реалізація та створення проекту

Отже для створення проекту потрібно для початку розробити графічний інтерфейс та визначити основну сторінку для того щоб користувач міг керувати основними параметрами інтелектуальної гри у жанрі вікторини, а саме темою питань, їх кількістю під час проходження гри та виділеного для цього часу.

Обов'язковою умовою для початку гри є обрання теми з трьох запропонованих на вибір. Також варто зазначити що мова для проходження гри англійська, тому що саме ця мова найбільш популярна у світі та дає можливість грати у цю гру користувачам які володіють англійською. За замовчуванням для користувача буде виділена одна хвилина, але за бажанням користувач може збільшити цей час до двох або трьох хвилин. За замовчуванням для користувача задано 15 питань різної складності. За бажанням користувач може змінити кількість питань до п'яти або десяти.

Для більш чіткого та зрозумілого представлення проекту його можна представити структурно у вигляді блок – схеми на рисунку 3.2

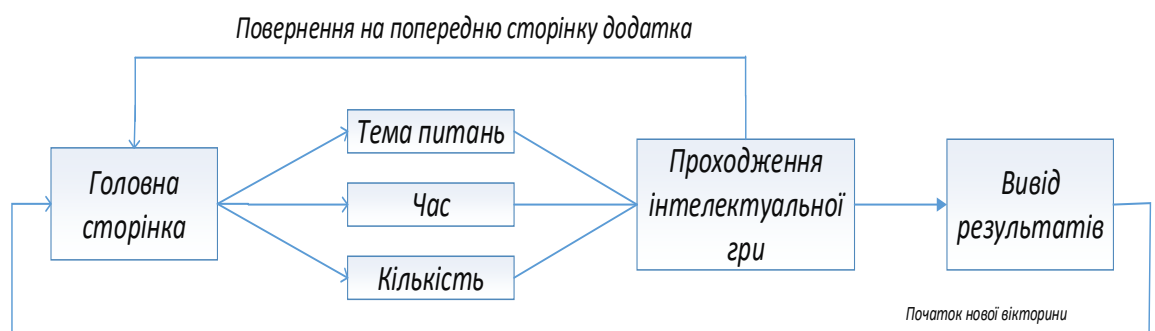


Рисунок 3.2 – Загальна структура гри

Завдяки цій схемі можна зрозуміти основну ідею та логічні операції мого дипломного проекту. Вигляд основної сторінки яка буде реалізована у цьому проекті можна продемонструвати на рисунку 3.3

Developed by Anna Nitsenko

Quick question - quick answer

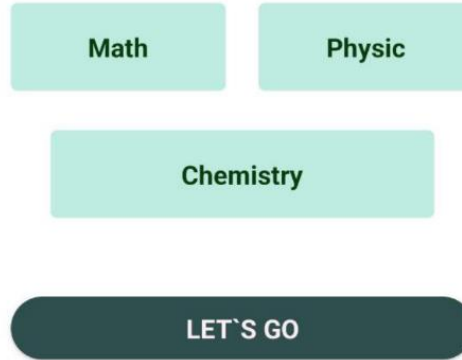


Рисунок 3.3 – Вигляд основної сторінки гри

Для того щоб розпочати гру потрібно вибрати одну з трьох запропонованих на вибір тем. Натиснувши на обрану тему вона буде підсвічена іншим кольором. Процедура вибору можна побачити на рисунку 3.4

Developed by Anna Nitsenko

Quick question - quick answer

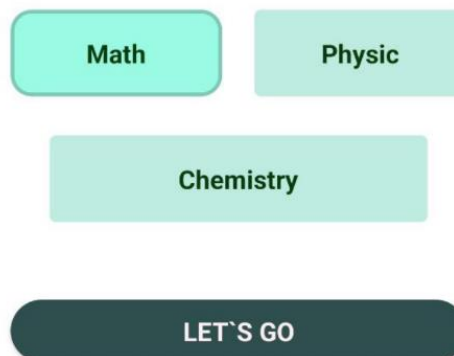


Рисунок 3.4 – Процедура вибору теми

За замовчуванням якщо користувач не обрав жоден з запропонованих варіантів часу, на проходження гри таймер встановлюється на одну хвилину та для користувача починається зворотній відлік. Зворотній відлік під час проходження гри в 1 хвилину можна побачити на рисунку 3.5

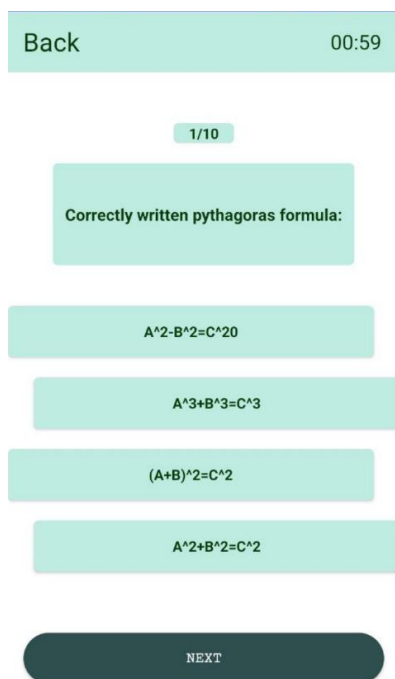


Рисунок 3.5 – Зворотній відлік при встановленні часу проходження в 1 хвилину

Для зручності користувача на сторінку гри в верхній лівий кут було додано кнопку для повернення на попередню сторінку, якщо користувач зробив хибний вибір параметрів та бажає змінити їх.

Кількість питань що буде виводитися на екран буде регулюватися з попереднім вибором. У випадку, якщо користувач не обрав бажану кількість питань за замовчуванням буде виведено усі питання з обраної теми.

Під час проходження гри користувач може побачити номер питання, його текст та чотири варіанти відповідей.

Після обрання варіанту відповіді користувач не може змінити її. У випадку правильного варіанту відповіді, він буде позначений зеленим

кольором. У випадку хибності вибраного варіанту, він буде визначений м'ятним кольором а вірний варіант автоматично зеленим кольором.

Приклади зміни кольору наведені на рисунках 3.6 та 3.7

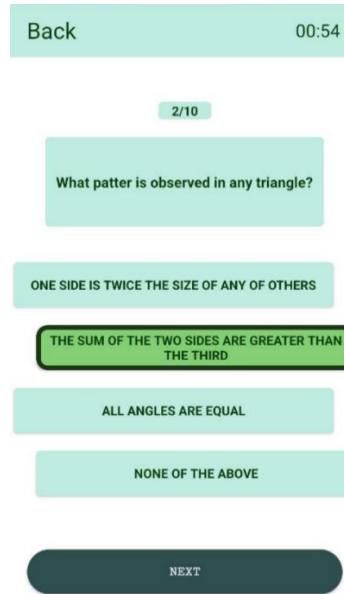


Рисунок 3.6 – Виділення правильної відповіді

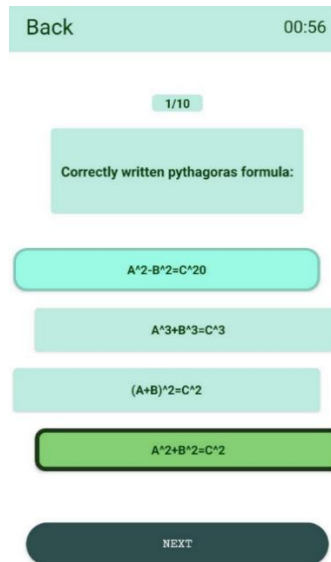


Рисунок 3.7 – Виділення хибної відповіді та позначення вірної

У разі якщо користувач намагається перейти до наступного питання не відповівши на поточне буде попередження у вигляді спливаючого

повідомлення про необхідність обов'язково дати відповідь. Вигляд попередження наведений на рисунку 3.8. Після проходження гри відбудеться вивід правильних та неправильних відповідей їх кількість та перехід до початку нової гри. Результат завершення гри можна побачити на рисунку 3.9

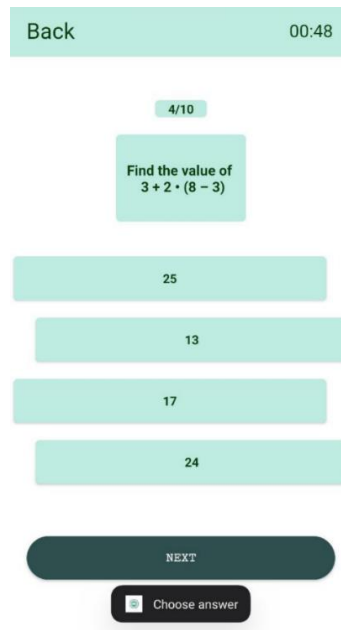


Рисунок 3.8 – Попередження користувача про надання відповіді

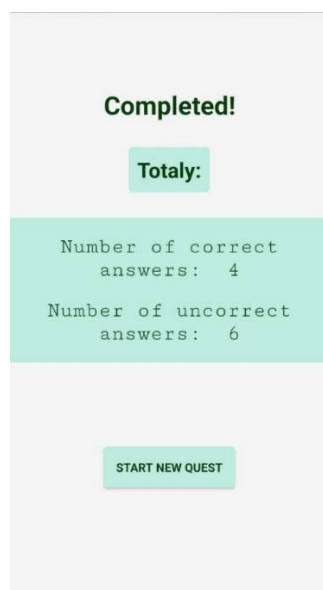


Рисунок 3.9 – Завершення гри

Отже була створена інтелектуальна гра вікторина яка повністю задовольняє користувача та виконує усі функції які були потрібні для коректної праці гри а саме: має усі описані в розділі активності, реалізує можливість вибору параметрів користувачем для проходження гри, реалізує проходження гри та має підказки для зручності користувача.

Лістинг коду програми буде наведений у додатку А.

ВИСНОВКИ

У рамках виконання дипломної роботи була розроблена інтелектуальна гра у вигляді вікторини для операційної системи Windows.

Методом аналізу та визначенням переваг та недоліків було обране найбільш вдале програмне забезпечення та мову програмування для майбутньої гри.

Основною платформою для розробки дипломного проекту був ігровий рушій Unity через низку переваг, а саме: популярність, доступність програмного забезпечення для розробки, широка ігрова бібліотека для створення проектів, широке коло підтримки розробників та багато літератури для вивчення цього ігрового рушія.

Перед початком було сформульовано задачу та перелік основних вимог до гри та інтерфейсу.

Мовою програмування гри було обрано мову програмування C# через низку переваг, а саме: широку бібліотеку для створення проектів, простий і доступний інтерфейс та численну літературу і посібники з основ та більш детального розгляду особливостей програмування на C#.

Розроблений ігровий застосунок на платформі Unity успішно демонструє геймплейну цікавість та може бути використаний для розваги, навчання, соціального впливу та інших цілей.

У підсумку під час розробки гри було проведено аналіз предметної області поставленої задачі та створено дипломний проект що відповідає обговореним вимогам та функціоналу.

Проект демонструє вміння працювати з різноманітними компонентами гри, створювати інтерактивність та надавати користувачам незабутні враження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бородкіна І., Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів. / І. Бородкіна, Г. Бородкін.; Київ: Видавництво "Центр учбової літератури", 2018. – 590 с.
2. CryEngine [Електронний ресурс] / режим доступу: <https://www.cryengine.com/>.
3. Unity 5.5 [Електронний ресурс] / режим доступу: <https://www.digitaltrends.com/>.
4. Unity Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unity3d.com/Manual/index.html>.
5. Дібрівний О.А. Вступ до об'єктно – орієнтованого програмування C# / О. А. Дібрівний, В. В. Гребенюк; Державний університет телекомунікацій. – Київ: Видавництво Державного університету телекомунікацій, 2018. – 190 с.
6. Комп'ютерна гра [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/PC_game.
7. 2D на Unity, докладний посібник [Електронний ресурс] – Режим доступу до ресурсу: <http://websketches.ru/blog/2d-igra-na-unity-podrobnoye-rukovodstvo-p1>.
8. How video games are made: the game development process [Електронний ресурс] / режим доступу: <https://www.cgspectrum.com/>.
9. Top Game Development Technologies and Trends in 2022 and in The Coming Years [Електронний ресурс] / режим доступу: <https://hackernoon.com/>.
10. Unity Game Engine [Електронний ресурс] – Режим доступу до ресурсу: <https://unity.com/ru>.

ДОДАТОК А ПРОГРАМНИЙ КОД

question_activity

```
package com.example.easyquest;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.List;
import java.util.Timer;
import java.util.TimerTask;
public class QuestActivity extends AppCompatActivity {
    private TextView question_num;
    private TextView question_text;
    private AppCompatButton answ_one, answ_two, answ_three, answ_four;
    private AppCompatButton answ_did;
    private Timer tic_tac;
    private int sec = 0;
    private int total_time = 1;
    private List<QuestionsList> question_list;
    private int current_question_num = 0;
    private String answ_by_user = "";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_quest);
final TextView back_button = findViewById(R.id.back_button);
final TextView timer = findViewById(R.id.timer);
final String get_what_topic = getIntent().getStringExtra("what_topic");
StartTimer(timer);
question_num = findViewById(R.id.questions_num);
question_text = findViewById(R.id.questions_text);
answ_one = findViewById(R.id.answ_one);
answ_two = findViewById(R.id.answ_two);
answ_three = findViewById(R.id.answ_three);
answ_four = findViewById(R.id.answ_four);
answ_did = findViewById(R.id.answ_did);
question_list = QuestionsBank.getQuestion(get_what_topic);
question_num.setText((current_question_num+1)+"/"+question_list.size());
question_text.setText(question_list.get(0).getQuestion_text());
answ_one.setText(question_list.get(0).getAnsw_one());
answ_two.setText(question_list.get(0).getAnsw_two());
answ_three.setText(question_list.get(0).getAnsw_three());
answ_four.setText(question_list.get(0).getAnsw_four());
back_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        tic_tac.purge();
        tic_tac.cancel();
        startActivity(new Intent(QuestActivity.this, MainActivity.class));
        finish();
    }
});

```

```

answ_one.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(answ_by_user.isEmpty()) {
            answ_by_user = answ_one.getText().toString();
            answ_one.setBackgroundResource(R.drawable.chosen_topic);
            RevealAnswer();
            question_list.get(current_question_num).setChosed_answ(answ_by_user);
        }
    }
});

answ_two.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(answ_by_user.isEmpty()) {
            answ_by_user = answ_two.getText().toString();
            answ_two.setBackgroundResource(R.drawable.chosen_topic);
            RevealAnswer();
            question_list.get(current_question_num).setChosed_answ(answ_by_user);
        }
    }
});

answ_three.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(answ_by_user.isEmpty()) {
            answ_by_user = answ_three.getText().toString();
            answ_three.setBackgroundResource(R.drawable.chosen_topic);
            RevealAnswer();
            question_list.get(current_question_num).setChosed_answ(answ_by_user);
        }
    }
});

```

```

        }
    }
});
answ_four.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(answ_by_user.isEmpty()) {
            answ_by_user = answ_four.getText().toString();
            answ_four.setBackgroundResource(R.drawable.chosen_topic);
            RevealAnswer();
            question_list.get(current_question_num).setChosed_answ(answ_by_user);
        }
    }
});
answ_did.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(answ_by_user.isEmpty()){
            Toast.makeText(QuestActivity.this,"Choose
answer",Toast.LENGTH_SHORT).show();
        }
        else {
            NextQuestion();
        }
    }
});
}
private void StartTimer (TextView timerView) {
    tic_tac = new Timer();
    tic_tac.scheduleAtFixedRate(new TimerTask() {

```

```

@Override
public void run() {
    if (sec == 0 && total_time == 0){
        tic_tac.purge();
        tic_tac.cancel();
        //Toast.makeText(QuestActivity.this,"Time is over!",
Toast.LENGTH_LONG).show();

        Intent intent = new Intent(QuestActivity.this, QuestResults.class);
        intent.putExtra("correct", GetCorrectAnsw());
        intent.putExtra("uncorrect",GetUncorrectAnsw());
        startActivity(intent);
        finish();
    }
    else if(sec == 0){
        total_time = total_time - 1;
        sec = 59;
    } else {
        sec--;
    }
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            String final_minutes = String.valueOf(total_time);
            String final_seconds = String.valueOf(sec);
            if(final_minutes.length()==1){
                final_minutes = "0"+final_minutes;
            }
            if(final_seconds.length()==1){
                final_seconds="0"+final_seconds;
            }
        }
    });
}

```



```

        timerView.setText(final_minutes + ":" + final_seconds);
    }
});
}
},1000,1000);
}
private int GetCorrectAnsw(){
    int correct_answers = 0;
    for(int i = 0;i<question_list.size();i++){
        final String get_user_chosed_answ = question_list.get(i).getChosed_answ();
        final String get_answer = question_list.get(i).getCorr_answ();
        if(get_user_chosed_answ.equals(get_answer) ){
            correct_answers++;
        }
    }
    return correct_answers;
}
private int GetUncorrectAnsw(){
    int uncorrect_answers = 0;
    for(int i = 0;i<question_list.size();i++){
        final String get_user_chosed_answ = question_list.get(i).getChosed_answ();
        final String get_answer = question_list.get(i).getCorr_answ();
        if(!get_user_chosed_answ.equals(get_answer) ){
            uncorrect_answers++;
        }
    }
    return uncorrect_answers;
}
@Override
public void onBackPressed() {

```

```

tic_tac.purge();
tic_tac.cancel();
startActivity(new Intent(QuestActivity.this, MainActivity.class));
finish();
}
private void RevealAnswer() {
    final String get_answer = question_list.get(current_question_num).getCorr_answ();
    if (answ_one.getText().toString().equals(get_answer)) {
        answ_one.setBackgroundResource(R.drawable.green_right);
    }
    else if (answ_two.getText().toString().equals(get_answer)) {
        answ_two.setBackgroundResource(R.drawable.green_right);
    }
    else if (answ_three.getText().toString().equals(get_answer)) {
        answ_three.setBackgroundResource(R.drawable.green_right);
    }
    else if (answ_four.getText().toString().equals(get_answer)) {
        answ_four.setBackgroundResource(R.drawable.green_right);
    }
}
private void NextQuestion () {
    current_question_num++;
    if((current_question_num+1) == question_list.size()){
        answ_did.setText("Admit");
    }
    if(current_question_num < question_list.size()) {
        answ_by_user = "";
        answ_one.setBackgroundResource(R.drawable.round_white);
        answ_two.setBackgroundResource(R.drawable.round_white);
        answ_three.setBackgroundResource(R.drawable.round_white);
    }
}

```

```

        answ_four.setBackgroundResource(R.drawable.round_white);
        question_num.setText((current_question_num+1)+"/"+question_list.size());
        question_text.setText(question_list.get(current_question_num).getQuestion_text());
        answ_one.setText(question_list.get(current_question_num).getAnsw_one());
        answ_two.setText(question_list.get(current_question_num).getAnsw_two());
        answ_three.setText(question_list.get(current_question_num).getAnsw_three());
        answ_four.setText(question_list.get(current_question_num).getAnsw_four());
    }
    else {
        Intent intent = new Intent(QuestActivity.this, QuestResults.class);
        intent.putExtra("correct", GetCorrectAnsw());
        intent.putExtra("uncorrect", GetUncorrectAnsw());
        startActivity(intent);
        finish();
    }
}
}
}

```

main_activity

```

package com.example.easyquest;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    private String what_topic = "";

```

@Override

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    final LinearLayout math = findViewById(R.id.math_quest);
    final LinearLayout chemistry = findViewById(R.id.chem_quest);
    final LinearLayout physic = findViewById(R.id.physic_quest);
    final Button start = findViewById(R.id.start_button);
    math.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            what_topic = "math";
            math.setBackgroundResource(R.drawable.chosen_topic);
            chemistry.setBackgroundResource(R.drawable.round_white);
            physic.setBackgroundResource(R.drawable.round_white);
        }
    });
    chemistry.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            what_topic = "chemistry";
            chemistry.setBackgroundResource(R.drawable.chosen_topic);
            math.setBackgroundResource(R.drawable.round_white);
            physic.setBackgroundResource(R.drawable.round_white);
        }
    });
    physic.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            what_topic = "physic";
```

```

        physic.setBackgroundResource(R.drawable.chosen_topic);
        math.setBackgroundResource(R.drawable.round_white);
        chemistry.setBackgroundResource(R.drawable.round_white);
    }
});
start.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(what_topic.isEmpty()) {
            Toast.makeText(MainActivity.this, "Chose quest`s topic",
Toast.LENGTH_SHORT).show();
        }
        else {
            Intent intent = new Intent(MainActivity.this, QuestActivity.class);
            intent.putExtra("what_topic", what_topic);
            startActivity(intent);
            finish();
        }
    }
});
}
}

```

result_activity.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"

```

```
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
tools:context=".QuestResults"  
android:background="#F5F5F5">
```

```
<TextView  
    android:id="@+id/congrad"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="15dp"  
    android:layout_marginTop="90dp"  
    android:layout_marginEnd="15dp"  
    android:fontFamily="sans-serif"  
    android:gravity="center"  
    android:text="Completed!"  
    android:textSize="30sp"  
    android:textStyle="bold"  
    android:textColor="#083c0c"/>
```

```
<TextView  
    android:id="@+id/res_text"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/congrad"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="30dp"  
    android:background="@drawable/round_white"  
    android:fontFamily="sans-serif"  
    android:padding="10dp"  
    android:text="Totally:"  
    android:textSize="25sp"
```

```
android:textStyle="bold"  
android:textColor="#083c0c"/>
```

```
<LinearLayout
```

```
    android:id="@+id/results"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical"  
    android:layout_centerHorizontal="true"  
    android:layout_below="@id/res_text"  
    android:layout_marginTop="30dp"  
    android:background="@drawable/round_white">
```

```
<TextView
```

```
    android:id="@+id/corr_num"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp"  
    android:gravity="center_horizontal"  
    android:padding="10dp"  
    android:text="0"  
    android:textColor="#083c0c"  
    android:fontFamily="serif-monospace"  
    android:textSize="25sp"/>
```

```
<TextView
```

```
    android:id="@+id/uncorr_num"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="10dp"  
    android:gravity="center_horizontal"  
    android:padding="10dp"  
    android:text="0"
```

```

        android:textColor="#083c0c"
        android:fontFamily="serif-monospace"
        android:textSize="25sp"/>
</LinearLayout>
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/start_new_one"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/results"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="100dp"
    android:background="@drawable/round_white"
    android:fontFamily="sans-serif"
    android:padding="15dp"
    android:text="Start new quest"
    android:textColor="#083c0c"
    android:textStyle="bold" />
</RelativeLayout>

```

question_activity.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".QuestActivity"

```



```

android:orientation="vertical"
android:background="@color/white">
<RelativeLayout
    android:id="@+id/control_button"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:background="#beece1">
<TextView
    android:id="@+id/back_button"
    android:layout_width="65dp"
    android:layout_height="match_parent"
    android:layout_centerVertical="true"
    android:layout_marginStart="15dp"
    android:layout_marginEnd="15dp"
    android:fontFamily="sans-serif"
    android:gravity="center"
    android:text="@string/back"
    android:textColor="#083c0c"
    android:textSize="25sp" />
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_alignParentEnd="true"
    android:layout_centerVertical="true"
    android:layout_marginEnd="15dp"
    android:gravity="center"
    android:orientation="horizontal">
<TextView
    android:id="@+id/timer"
    android:layout_width="65dp"

```

```
        android:layout_height="match_parent"
        android:text=""
        android:textColor="#083c0c"
        android:textSize="20sp"
        android:fontFamily="sans-serif"
        android:gravity="center"/>
    </LinearLayout>
</RelativeLayout>
<TextView
    android:id="@+id/questions_num"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center"
    android:layout_marginTop="50dp"
    android:background="@drawable/round_white"
    android:elevation="1dp"
    android:paddingStart="15dp"
    android:paddingEnd="15dp"
    android:text="1/10"
    android:textColor="#083c0c"
    android:fontFamily="sans-serif"
    android:textStyle="bold"/>
<TextView
    android:id="@+id/questions_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    android:layout_marginTop="20dp"
```

```
android:layout_gravity="center"
android:background="@drawable/round_white"
android:gravity="center"
android:minHeight="100dp"
android:text="Questions will be here soon!"
android:fontFamily="sans-serif"
android:padding="12dp"
android:textSize="16sp"
android:textStyle="bold"
android:textColor="#083c0c"/>
```

```
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/answ_one"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginStart="5dp"
    android:layout_marginEnd="30dp"
    android:layout_marginTop="40dp"
    android:background="@drawable/round_white"
    android:text="Answer one"
    android:fontFamily="sans-serif"
    android:textColor="#083c0c"
    android:textStyle="bold"/>
```

```
<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/answ_two"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginStart="30dp"
    android:layout_marginEnd="5dp"
    android:layout_marginTop="20dp"
    android:background="@drawable/round_white"
```

```
android:text="Answer two"  
android:fontFamily="sans-serif"  
android:textStyle="bold"  
android:textColor="#083c0c"/>
```

```
<androidx.appcompat.widget.AppCompatButton  
  android:id="@+id/answ_three"  
  android:layout_width="match_parent"  
  android:layout_height="50dp"  
  android:layout_marginStart="5dp"  
  android:layout_marginEnd="30dp"  
  android:layout_marginTop="20dp"  
  android:background="@drawable/round_white"  
  android:text="Answer three"  
  android:fontFamily="sans-serif"  
  android:textStyle="bold"  
  android:textColor="#083c0c"/>
```

```
<androidx.appcompat.widget.AppCompatButton  
  android:id="@+id/answ_four"  
  android:layout_width="match_parent"  
  android:layout_height="50dp"  
  android:layout_marginStart="30dp"  
  android:layout_marginEnd="5dp"  
  android:layout_marginTop="20dp"  
  android:background="@drawable/round_white"  
  android:text="Answer four"  
  android:fontFamily="sans-serif"  
  android:textStyle="bold"  
  android:textColor="#083c0c"/>
```

```
<androidx.appcompat.widget.AppCompatButton  
  android:id="@+id/answ_did"
```

```
android:layout_width="match_parent"
android:layout_height="50dp"
android:layout_marginStart="20dp"
android:layout_marginEnd="20dp"
android:layout_marginTop="60dp"
android:background="@drawable/start_but"
android:text="Next"
android:textColor="@color/white"
android:fontFamily="serif-monospace"
android:textStyle="bold"/>
```

```
</LinearLayout>
```

main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/white"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/text_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
android:layout_marginStart="20dp"
android:layout_marginTop="100dp"
android:layout_marginEnd="20dp"
android:layout_marginBottom="30dp"
android:fontFamily="sans-serif"
android:gravity="center_horizontal"
android:text="Quick question - quick answer"
android:textAlignment="center"
android:textColor="#083c0c"
android:textSize="30sp"
android:textStyle="bold" />
```

```
<LinearLayout
```

```
    android:id="@+id/first_lay"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_below="@id/text_1"
    android:layout_marginTop="10dp"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    android:weightSum="2">
```

```
<LinearLayout
```

```
    android:id="@+id/math_quest"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="20dp"
    android:layout_marginEnd="20dp"
```

```
    android:background="@drawable/round_white">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif"
        android:text="Math"
        android:textColor="#083c0c"
        android:textSize="20sp"
        android:textStyle="bold" />
</LinearLayout>
<LinearLayout
    android:id="@+id/physic_quest"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="20dp"
    android:layout_marginStart="5dp"
    android:background="@drawable/round_white">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif"
        android:text="Physic"
        android:textColor="#083c0c"
        android:textSize="20sp"
        android:textStyle="bold" />
</LinearLayout>
</LinearLayout>
```

```
<LinearLayout
```

```
    android:id="@+id/second_lay"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:layout_below="@id/first_lay"  
    android:layout_marginTop="30dp"  
    android:layout_marginStart="50dp"  
    android:layout_marginEnd="50dp"  
    android:weightSum="2">
```

```
<LinearLayout
```

```
    android:id="@+id/chem_quest"  
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:layout_weight="2"  
    android:orientation="vertical"  
    android:gravity="center"  
    android:padding="20dp"  
    android:background="@drawable/round_white">
```

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:fontFamily="sans-serif"  
    android:text="Chemistry"  
    android:textColor="#083c0c"  
    android:textSize="20sp"  
    android:textStyle="bold" />
```

```
</LinearLayout>
```



```
</LinearLayout>
```

```
<androidx.appcompat.widget.AppCompatButton
```

```
    android:id="@+id/start_button"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/second_lay"  
    android:layout_marginStart="20dp"  
    android:layout_marginTop="60dp"  
    android:layout_marginEnd="20dp"  
    android:background="@drawable/start_but"  
    android:fontFamily="sans-serif"  
    android:text="LET`S GO"  
    android:textColor="#FFF0F5"  
    android:textSize="20sp"  
    android:textStyle="bold" />
```

```
<TextView
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="10dp"  
    android:layout_marginTop="50dp"  
    android:layout_marginEnd="50dp"  
    android:fontFamily="sans-serif"  
    android:gravity="left"  
    android:text="Developed by Anna Nitsenko"  
    android:textColor="@color/black"  
    android:textSize="17sp"  
    android:textStyle="italic" />
```

```
</RelativeLayout>
```