

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *Денисенко Євген Олексійович*
(ПІБ)

академічної групи *121-18-1*
(шифр)

спеціальності *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

освітньої програми *Інженерія програмного забезпечення*
(назва освітньої програми)

на тему: *Розробка ігрового застосунку для платформи Android на основі Kotlin*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтингово ю	інституційн ою	
кваліфікаційної роботи	<i>Проф. Якунін А.О.</i>			
розділів:				
спеціальний	<i>Проф. Якунін А.О.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2022

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних
систем

_____ (повна назва)

_____ (підпис)

_____ (прізвище, ініціали)

« » _____ 2022 року

ЗАВДАННЯ

на кваліфікаційну роботу

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 121-18-1

(група)

Денисенко Є.О.

(прізвище та ініціали)

тема кваліфікаційної роботи

платформи Android на основі Kotlin

Розробка ігрового застосунку для

затверджена наказом ректора НТУ «ДП» від

№

Розділ	Зміст виконання	Термін виконання
<i>Спеціальний</i>	<i>На основі матеріалів проєктно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<i>13.05.2022 р.</i>
<i>Економічний</i>	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	<i>27.05.2022 р.</i>

Завдання видав

_____ (підпис)

Проф. Якунін А.О.

(посада, прізвище,
ініціали)

Завдання прийняв до виконання

_____ (підпис)

Денисенко Є.О.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

РЕФЕРАТ

Пояснювальна записка: 54 с., 19 рис., 3 дод., 10 джерело

Об'єкт розробки: ігровий застосунок для платформи Android на основі мови програмування Kotlin.

Мета кваліфікаційної роботи: створення застосунку який дасть змогу дізнатись якими були ігри в кінці XX століття, шляхом створення на платформі Android.

У вступі розглядається стан проблеми, а також підтвердження актуальності ігор часів кінця XX століття.

У першому розділі розглянуто існуючі технології, призначення розробки, визначена постановка завдання.

У другому розділі розглянуто технології, які доступні у наш час, обирається платформа для розробки, виконується проектування та розробка програми, наводиться опис алгоритмів та структури функціонування програми.

В економічному розділі розраховано вартість роботи по створенню додатку та визначено час на його розробку.

Метою практичного значення є розробка додатка, який за своєю тематикою та графічними складовими відповідає ігровим додаткам кінця XX століття.

Актуальність даного програмного продукту визначається попитом щодо ігор минулого століття, а також популярність ігрових застосунків з «піксельною графікою» .

Список ключових слів: **МОБІЛЬНИЙ ДОДАТОК, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ТЕХНОЛОГІЇ, ВІРТУАЛЬНА МАШИНА.**

ABSTRACT

Explanatory note: 54 p., 6 figs., 19 appx., 10 sources.

Object of development: a game application for the Android platform based on the Kotlin programming language.

The purpose of the qualifying work: to create an application that will allow you to find out what games were like in the late twentieth century, by creating on the Android platform.

The introduction examines the state of the problem, as well as confirmation of the relevance of the games of the late twentieth century.

The first section considers the existing technologies, the purpose of development, defines the task.

The second section discusses the technologies that are available today, selects a platform for development, performs design and development of the program, describes the algorithms and structure of the program.

In the economic section, the cost of work on creating an application is calculated and the time for its development is determined.

The purpose of practical significance is to develop an application that in its subject matter and graphic components corresponds to the game applications of the late twentieth century.

The relevance of this software product is determined by the demand for games of the last century, as well as the popularity of gaming applications with "pixel graphics".

List of keywords: MOBILE APPLICATION, SOFTWARE, TECHNOLOGIES.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ- програмне забезпечення

ОС – операційна система

IDE - Integrated Development Environment;

JVM – Java Virtual Machine

JRE – Java Runtime Environment

ВМ – віртуальна машина

ЦП – центральний процесор

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
РОЗДІЛ 1_АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ..	9
1.1. Загальні відомості з предметної галузі	9
1.2. Призначення розробки та галузь застосування.....	10
1.3. Підстави для розробки.....	11
1.4. Постановка завдання.....	12
1.5. Вимоги до програми або програмного виробу.....	12
1.5.1. Вимоги до функціональних характеристик.....	12
1.5.2. Вимоги до інформаційної безпеки	13
1.5.3. Вимоги до складу та параметрів технічних засобів	13
1.5.4. Вимоги до інформаційної та програмної сумісності.....	13
РОЗДІЛ 2_ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ .	14
2.1. Функціональне призначення програми.....	14
2.2. Опис застосованих математичних методів.....	14
2.4. Опис структури системи та алгоритмів її функціонування	18
Рис. 2.4 Інтерфейс IDE Android Studio.....	20
2.5. Обґрунтування та організація вхідних та вихідних даних програми	25
2.7. Опис розробленої системи	25
РОЗДІЛ 3_ЕКОНОМІЧНИЙ РОЗДІЛ.....	31
3.1. Розрахунок трудомісткості та вартості розробки програмного продукту ...	31
3.2. Розрахунок витрат на створення програми	35
ВИСНОВКИ.....	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	38

ДОДАТОК А **Ошибка! Закладка не определена.**

КОД ПРОГРАМИ..... **Ошибка! Закладка не определена.**

ДОДАТОК Б **Ошибка! Закладка не определена.**

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ**Ошибка! Закладка не определена.**

ДОДАТОК В **Ошибка! Закладка не определена.**

Перелік файлів на диску **Ошибка! Закладка не определена.**

ВСТУП

Мета цієї кваліфікаційної роботи безпосередньо пов'язані з напрямом підготовки та відповідає узагальненій тематиці кваліфікаційних робіт, типових задач діяльності, умінню та компетенціям, якими повинен володіти бакалавр 121 «Інженерія програмного забезпечення».

Починаючи з середини 1970-х років. У світі активно почала розвиватися ера «аркадних ігор». Один з перших електронних аркадних автоматів випустила компанія «Atari». Система представляла із себе гру в теніс. Пізніше, в 1975 була створена «ігрова приставка», яка мала змогу підключатися до телевізора.

У 1980-х , з'явилися відеоігри в новій якості, вони стали різнокольоровими. У 1991 на персональному комп'ютері з'явилась одна із перший ігор з тривимірною графікою «Catacomb 3-D». З появою цієї гри, у розробників з'явилась змога накладати різні текстури, таким чином однакові моделі могли бути візуально різними. До 2000 року, багато компаній займалися удосконаленням, оптимізацією та розповсюдженням їх по всьому світу.

В продовж десятиліття з'являлися як нові жанри ігор, так і нові прилади для комфортного управління. Починаючи з 2005 року, популярними жанрами стали: «екшн», «симулятор», «пригоди».

Починаючи з 2014 року розробники розділились на декілька категорій графічного дизайну ігор. Деякі компанії продовжили удосконалення технологій, з'явилися шоломи віртуальної реальності, для анімації персонажів почали використовувати технологію захоплення рухів акторів.

Інші вирішили повернутися до 2D графіки.

З появою смартфонів у 2007 році, з'явилась ще одна платформа в якій невеликі розробники могли показати свої продукти для широкої аудиторії.

На даний момент ігри на смартфонах складають значну частину від загальної кількості існуючих ігор на всіх видах пристроїв. Мобільні телефони мають перевагу від інших приладів, оскільки вони дешевше, компактніші, більш доступні.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

Комп'ютерна гра — це гра, в яку грають за допомогою комп'ютера. Це форма розваги у сучасному світі. Комп'ютерні ігри постійно розвиваються, покращуючи свої технічні характеристики і стаючи усе більш економічним чинником у багатьох країнах світу.

Термін «комп'ютерні ігри» в першу чергу відноситься до ігор, спеціально створених для класичних(наприклад, особистих або домашніх) комп'ютерів, тоді як «консольні ігри» відноситься до ігор, в які грають на спеціально розроблених «консолях»(які здебільшого являються також комп'ютери). Деякі консолі мають вбудовані монітори, а інші використовують телевізор для відображення зображення.

Термін «відеоігри» відноситься до усіх ігор, які використовують екран для відображення контенту.Перша відеогра була зроблена американцем Стівом Расселом, але вона не мала великого успіху. Потім, через десять років, американець Нельсон Бушнелл також зробив першу відеоігру трохи більше успішною на ринку. Тематика гри «Теніс».

З цього моменту подальший розвиток та випуск ігор, ігрових систем та додатків до них поділяють на покоління, які зумовлюються технічним прогресом та заміною старіших приладів на нові. Ігрова приставка (у випадку з кишеньковими системами правильніше ігрова консоль) - спеціалізоване електронний пристрій , розроблене і створене для відеоігор. Найбільш часто використовуваним пристроєм виведення є телевізор або , рідше , комп'ютерний монітор - тому такі пристрої і називають приставками , так як вони приставляються до незалежного пристрою відображення . Портативні (кишенькові) ігрові системи мають власне вбудоване пристрій відображення (ні

до чого не приставляються) , тому називати їх ігровими приставками дещо некоректно. Виділяють різні покоління ігрових систем, на даний момент їх вісім.

На початку 1990 року компанія NEC випустила TurboGrafx-CD - CD-привід для консолі TurboGrafx-16. Її спробували наслідувати інші компанії, але на ігрових приставках диски не знайшли поширення через локальність їх випуску та відсутність єдиних стандартів. 8 жовтня 1991 року угодою між Fujitsu, Microsoft, Philips і Tandy, було визначено набір стандартів для ПК, який включав в себе CD-привід, графічну і звукову карти. Це дозволило випускати CD-приводи з гарантією, що вони окупляться. До 1993 існування ігор на CD-ROM стало комерційно життєздатним. Першою дійсно успішною грою для ПК на CD стала 7th Guest, що продалася в кількості мільйона копій. Star Wars: Rebels Assault використовувала графіку, створену на комп'ютерах, що використовувалися для створення спецефектів у фільмах. У вересні 1993 вийшла Myst, що завдяки неквапній подачі сюжету і візуальній складовій стала найбільш продаваною відеогрою року на ПК, за перші тижні продавшись у кількості 4 млн копій.

Починаючи з 2007 року. З'являються перші смартфони. З цього періоду індустрія мобільних ігор почала активно розвиватися.

На даний момент, на ринку ігрових застосунків, переважають мобільні додатки

1.2. Призначення розробки та галузь застосування

Програмний додаток що створено для кваліфікаційної роботи, має назву «Розробка ігрового застосунку для платформи Android на основі Kotlin»

Основні терміни та ключові слова:

Java Virtual Machine – основна частина виконуючої системи Java, так названої JRE. VM Java виконує байт-код Java(набір інструкцій які виконує VM Java).

JVM являється ключовим компонентом платформи Java. Віртуальні машини Java доступні для багатьох апаратних і програмних платформ. Java може бути як з'єднуючим звеном і як самостійна платформа. Java є кросс- платформеною мовою програмування, один і той же код програми можна запустити і на інших платформах.

Kotlin – статично типізована, об'єктно орієнтована мова програмування, працююча поверх Java Virtual Machine. Перевагою Kotlin є його простота, більш швидка компіляція і краща підтримка мови IDE. Мова також сумісна з Java. За допомогою Kotlin ми можемо створювати додатки до ОС Android, що дозволяє додавати нові функції без переписування додатку цілком.

Розроблений продукт може використовуватися на різних платформах з різними версіями ОС Android.

Призначення розробки - показати людям якими були ігрові застосунки раніше. Також даний додаток допоможе людям провести час.

Середовищем розробки було обрано Android Studio. Це інтегроване середовище для розробки Android. Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування застосунків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, тощо)

1.3. Підстави для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу.

Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується з наказом ректора.

Таким чином підставами для розробки (виконанням кваліфікаційної роботи) є:

- освітня програма спеціальності 121 “Інженерія програмного забезпечення”;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету “Дніпровська політехніка” № 000-х від 00.00.2022р;
- завдання на кваліфікаційну роботу на тему «Розробка ігрового застосунку для платформи Android на основі Kotlin»

1.4. Постановка завдання

Метою проекту - розробити програмний додаток який дозволить у повній мірі показати якими були ігрові застосунки на початку ігрової індустрії.

Під час роботи необхідно розробити додаток яким буде зручно користуватися.

Кінцевий продукт повинен мати наступний функціонал:

- зображення противників і мапи
- керування персонажем
- підрахунок балів після завершення гри

Наведений програмний додаток буде створений на мові програмування високого рівня Kotlin, переваги завдяки яким була обрана ця мова це простота, кросс-платформерність.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Кінцевий продукт повинен мати та дотримуватися наступних функціональних вимог:

- поле для гри;
- рух танків;

- стрільба;
- знищення противників;
- підсумок балів;

1.5.2. Вимоги до інформаційної безпеки

Основні вимоги до інформаційної безпеки під час роботи з додатком:

- конфіденційність інформації;

Під час роботи з додатком, жодних дані не зберігаються.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для стабільної роботи з додатком вам потрібні відповідні технічні або подібні характеристики обладнання для операційної системи Android:

- операційна система(ОС) Android 4.2 і вище;
- ЦП: Exynos 4412;
- накопичувач: 8 Гб;
- оперативна пам'ять: 3 Гб;

1.5.4. Вимоги до інформаційної та програмної сумісності

Мовою програмування було обрано Kotlin. Ця програмна мова була використана для створення рухів, стрільби та створення мапи. Код повинен бути написаний за всіма стандартами та згідно з договором про стиль коду Kotlin.

Для редагування та розробки вихідних кодів на мові Kotlin було використано інтегроване середовище розробки (IDE) Android Studio яке написано мовою Java.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

Результатом виконання цієї кваліфікаційної роботи має бути програмний додаток, який представляє собою танкову гру.

Кінцевим результатом роботи програми має бути вікно в якому зображено мапу для пересування, головного героя, ворогів, після знищення головного героя, виводиться кількість балів.

Основне призначення додатку:

- створення продукту, який у повній мірі дозволяє цікаво провести час.
- розробити зручне керування

2.2. Опис застосованих математичних методів

Під час проектування та розробки даної інформаційної системи використовувалися лише прості арифметичні дії. Математичні методи не використовувалися.

2.3. Опис використаної архітектури та шаблонів проектування

Операційна система Android

Android - вільне програмне забезпечення. Саме відкритий код дозволяє модифікувати базову операційну систему, а також створювати її нові дистрибутиви. Справа в тому, що дана операційна система може бути встановлена практично на будь-який мобільний пристрій на відміну, наприклад, від iOS (операційна система компанії Apple, яка призначена для iPhone, iPad, а також інших пристроїв, виробництва цієї компанії).

Крім того, Android має розвинене співтовариство програмістів, що створюють спеціальні програми під цю операційну систему, які розширюють її функціональність. Як правило, практично всі програми для Android написані на мові програмування Java. Утиліти для цієї операційної системи можуть поширюватися безкоштовно, а деякі - за гроші

Легкість налаштування, безкоштовність, простота в роботі і дозволили Android стати однією з найбільш широко використовуваних мобільних платформ. З кожним роком до розробки Android і вдосконалення додатків приєднується все більша кількість людей, що додають нові функції або ж адаптують Android під пристрої, які спочатку для цієї операційної системи не були призначені.

Програмне забезпечення Android має кілька версій. Першою версією цієї операційної системи була 1.0, а на сьогоднішній день найактуальнішою є 10. Компанія Google дала назви різним версіям з кулінарного принципу (желе, сендвіч, морозиво та ін.).

Список Найпопулярніших версій Android:

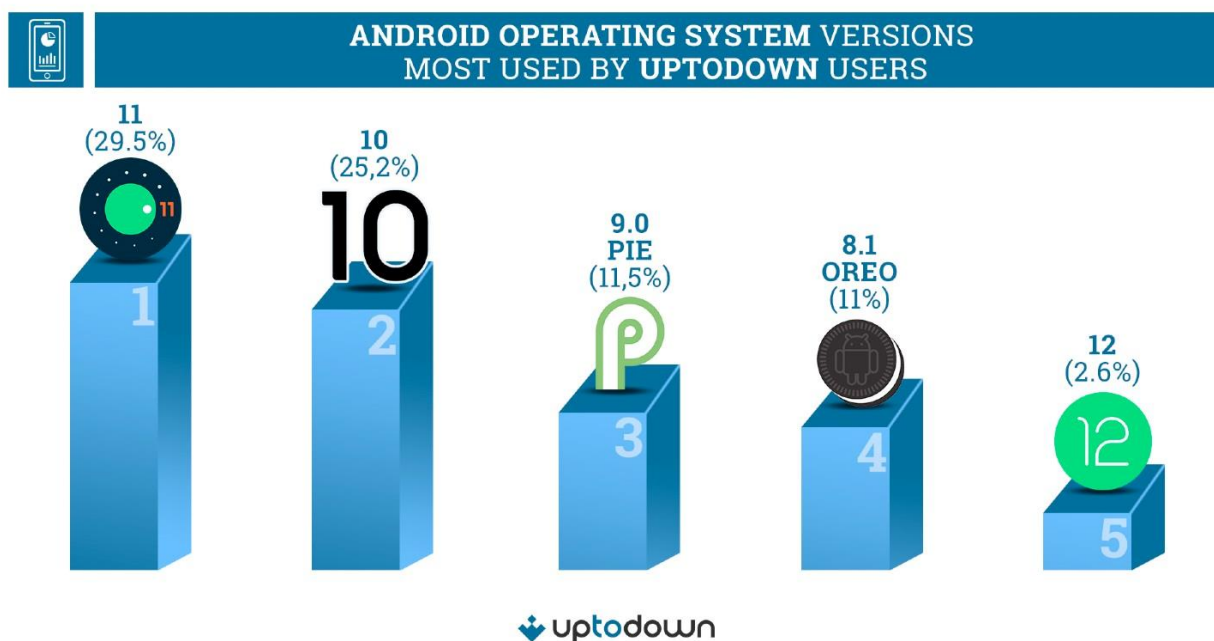


Рис 2.1 Результати запиту використання версії Android

Переваги ОС Android

Одним з безперечних достоїнств цієї операційної системи є її безкоштовність. Вона цілком успішно конкурує з іншими виробниками програмного забезпечення для мобільних девайсів (компанії Apple і Windows). Також Android дуже легко налаштувати на власний смак, а якщо недостатньо

стандартних налаштувань, можна підлаштувати систему під себе за допомогою зміни вихідного коду. Безперечною перевагою є і можливість встановлювати на Android утиліти сторонніх розробників, а також інсталювати цю систему на будь-який мобільний пристрій.

Під час проектування додатку, мовою для програмування було обрано Kotlin для реалізації усіх поставлених завдань.

Kotlin можна визначити як мову програмування з відкритим кодом, розроблену Jet-brains та підтримувану Google, через фонд Котлін і працює на віртуальній машині Java. Він має можливість поєднувати функціональні операції та об'єктно-орієнтоване програмування на платформі з відкритим кодом. Він офіційно оголошений мовою програмування для розробки Android в травні 2017 р. Котлін дає кодеру безпечно і надійно писати код.

Існує багато областей, в яких Котлін може зайняти важливу позицію. Саме його можна використовувати майже в кожній області, де використовується Java. Давайте тепер нам відомо кілька сценаріїв, де це можна використовувати.

- У науковій грі розробки.
- Аналіз даних
- Повторне використання коду для програм Android в IOS.
- Вбудовані системи.
- Веб-додатки, такі як Netflix.

При розробці та проектуванні даного ігрового додатку використовувалась система автоматичної збірки Gradle.

Ручна збірка додатку для запуску на пристрої потребує багато часу та ресурсів. Вона затягує процес розробки.

Щоб прискорити його, програмісти створюють скрипти, це скорочує час збірки, але ускладнює роботу над командним проектом.

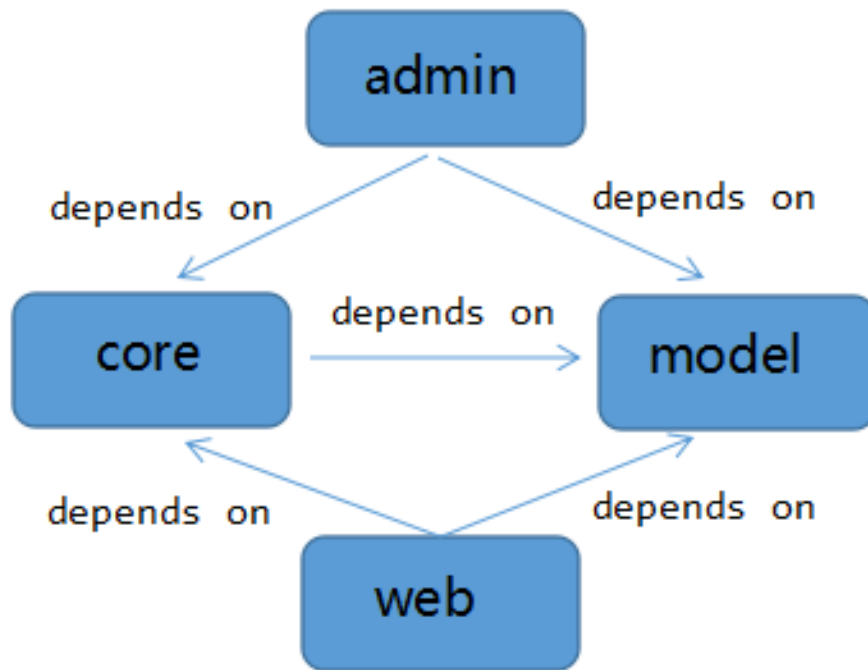


Рис 2.2 Зображення взаємозв'язку модулів

Kotlin - це функціональна об'єктно-орієнтована мова програмування зі статичним написанням, що дозволяє компілювати для віртуальної машини Java та JavaScript. Він розроблений командою програмістів з JetBrains, редактором IntelliJ IDEA, інтегрованого середовища розробки для Java і на якій базується Android Studio, офіційна IDE для розробки програм для Android.

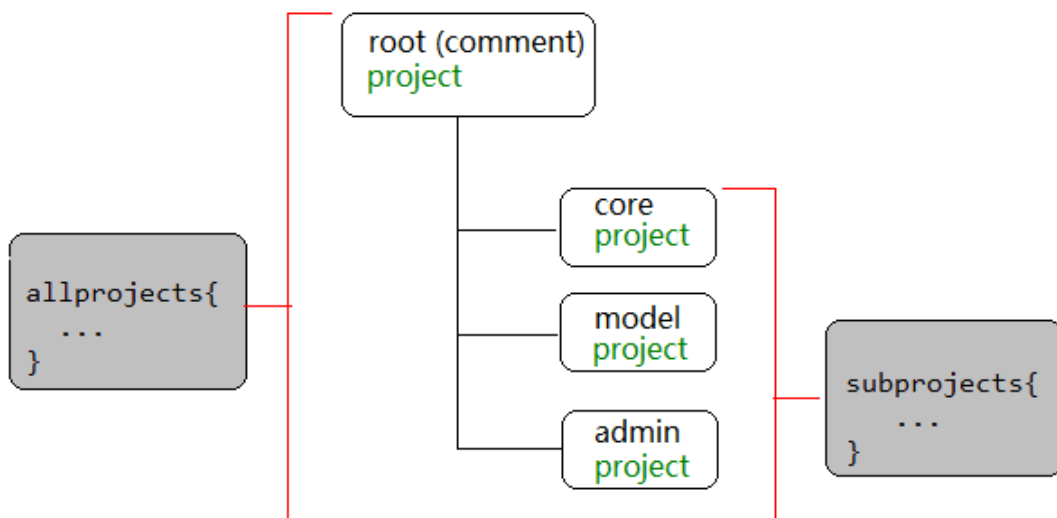


Рис 2.3 Зображення області конфігурації сценарії збірки

Підтримка EDI для Kotlin: Android Studio побудована на основі JetBrains IntelliJ IDEA, і команда JetBrains роками працює над забезпеченням безперебійної роботи Kotlin з IntelliJ IDEA

Мова Kotlin нова, але стрімко розвивається, мова програмування. Сумісність з Google і Java, лаконічність синтаксису і можливість реалізувати будь-які завдання робить її популярною серед розробників мобільних додатків. При створенні Kotlin використовувалися побажання і напрацювання розробників. Тому незабаром мова може стати основною для андроїд додатків.

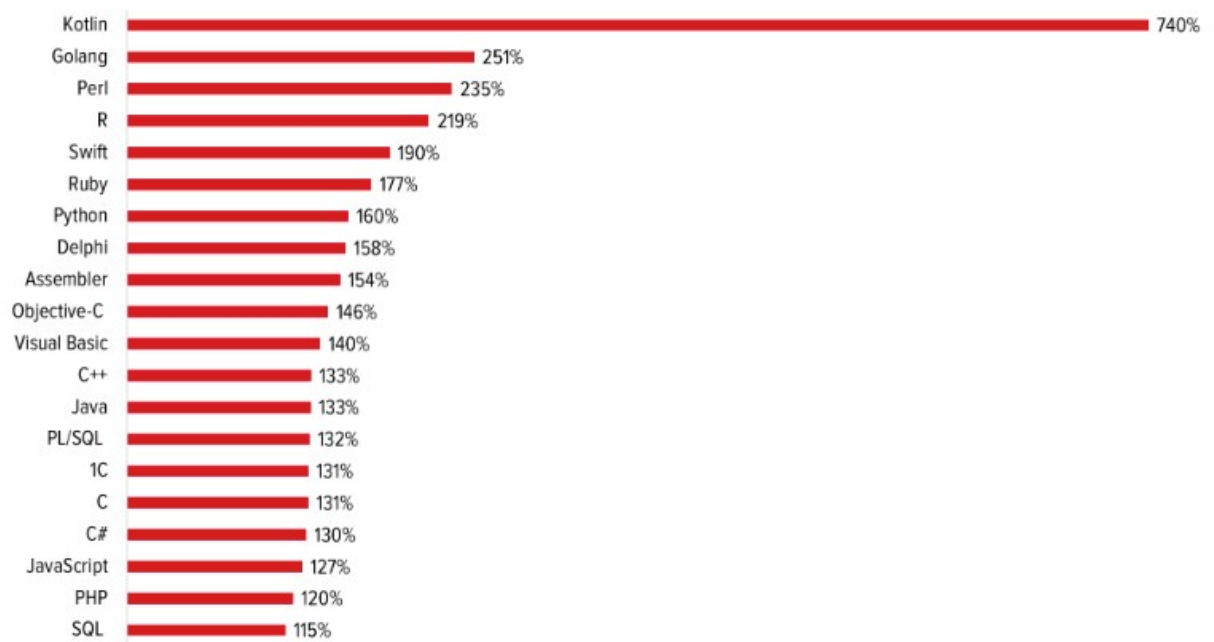


Рис 2.4 Результат запиту попит на мови програмування в період 2020-2021рр

2.4. Опис структури системи та алгоритмів її функціонування

- Мова програмування Kotlin.
- Android Studio

Android Studio – інтегроване середовище розробки виробництва Google, за допомогою якого розробникам стають доступні інструменти для створення програм на платформі Android OS. Android Studio можна встановити на Windows, Mac та Linux. Android Studio створене з урахуванням IntelliJ IDEA.

Android Studio містить інструменти для розробки рішень для смартфонів і планшетів, а також нові технологічні рішення для Android TV, Android Wear, Android Auto, Glass і додаткові контекстуальні модулі.

Особливості Android Studio

Нові функції відображаються з кожною новою версією Android Studio. На даний момент доступні такі функції:

- Розширений редактор макетів: WYSIWYG, здатність працювати з UI компонентами за допомогою Drag-and-Drop, функція попереднього перегляду макета на декількох конфігураціях екрана.
- Складання додатків, засноване на Gradle.
- Різні види збірок та генерація декількох .apk файлів
- Рефакторинг коду
- Статичний аналізатор коду (Lint), що дозволяє знаходити проблеми продуктивності, несумісності версій та інше.
- Вбудований ProGuard та утиліта для підписування додатків.
- Шаблони основних макетів та компонентів Android.
- Підтримка розробки додатків для Android Wear та Android TV.
- Вбудована підтримка Google Cloud Platform, яка включає інтеграцію з сервісами Google Cloud Messaging і App Engine.

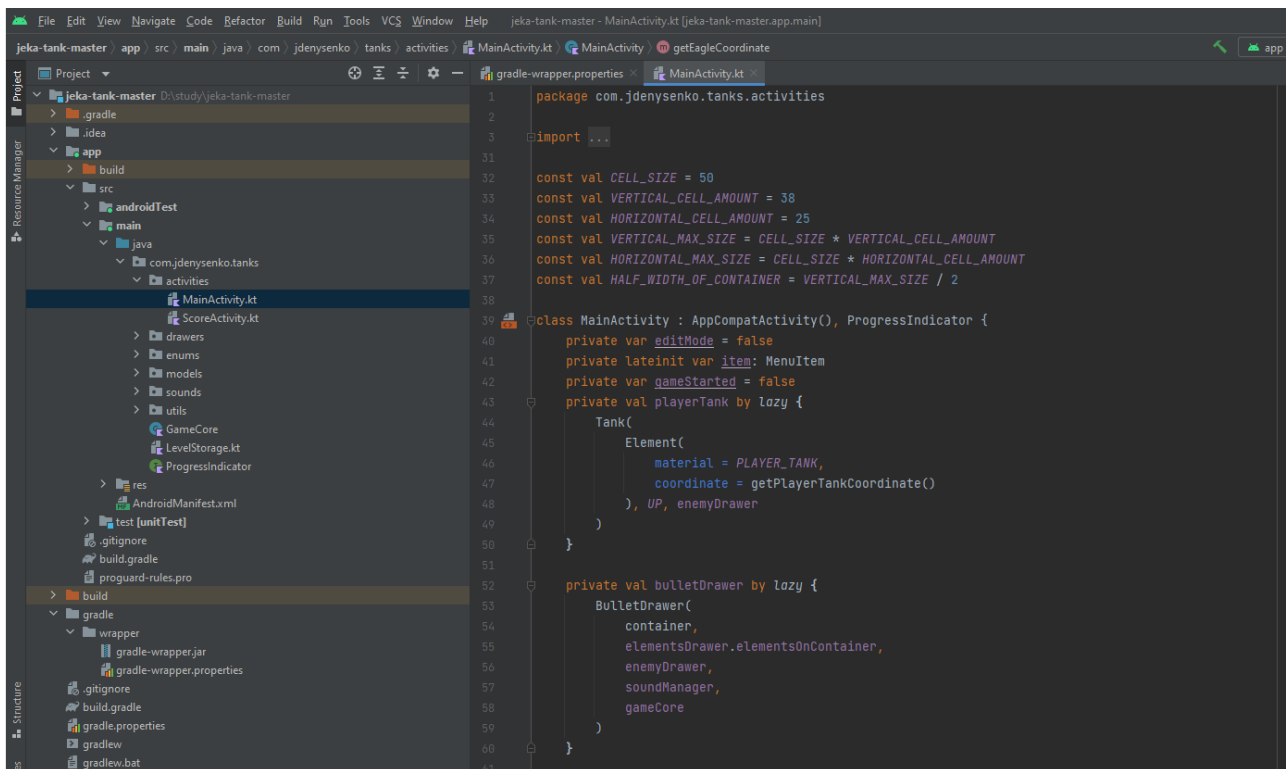


Рис. 2.5 Зображення інтерфейсу IDE Android Studio

Для того щоб була можливість редагувати мапу, та для подальшої розробки програми, була створена віртуальна система координат. В подальшому ця частина нам знадобиться для створення сітки, яка допоможе редагувати мапу, а також завдяки системі координат, з'явиться можливість для створення траєкторії кулі.

```

const val CELL_SIZE = 50
const val VERTICAL_CELL_AMOUNT = 38
const val HORIZONTAL_CELL_AMOUNT = 25
const val VERTICAL_MAX_SIZE = CELL_SIZE * VERTICAL_CELL_AMOUNT
const val HORIZONTAL_MAX_SIZE = CELL_SIZE * HORIZONTAL_CELL_AMOUNT
const val HALF_WIDTH_OF_CONTAINER = VERTICAL_MAX_SIZE / 2

```

Рис 2.6 Зображення віртуальної системи координат

У нас є фіксована кількість комірок в грі . Система координат комірок відрізняється від кількості пікселів на екрані. За умовами гри, танки рухаються по клітинках. Створюємо поле, для більш зручного редагування мапи, створимо сітку, яка буде зображуватися при редагуванні мапи.

```
private fun drawHorizontalLines() {
    var topMargin = 0
    while (topMargin <= container.layoutParams.height) {
        val horizontalLine = View(container.context)
        val layoutParams = FrameLayout.LayoutParams(FrameLayout.LayoutParams.MATCH_PARENT, height: 1)
        topMargin += CELL_SIZE
        layoutParams.topMargin = topMargin
        horizontalLine.layoutParams = layoutParams
        horizontalLine.setBackgroundColor(container.context.resources.getColor(android.R.color.white))
        allLines.add(horizontalLine)
        container.addView(horizontalLine)
    }
}
```

Рис 2.7 Зображення горизонтальних ліній

```
private fun drawVerticalLines() {
    var leftMargin = 0
    while (leftMargin <= container.layoutParams.width) {
        val verticalLine = View(container.context)
        val layoutParams = FrameLayout.LayoutParams(width: 1, FrameLayout.LayoutParams.MATCH_PARENT)
        leftMargin += CELL_SIZE
        layoutParams.leftMargin = leftMargin
        verticalLine.layoutParams = layoutParams
        verticalLine.setBackgroundColor(container.context.resources.getColor(android.R.color.white))
        allLines.add(verticalLine)
        container.addView(verticalLine)
    }
}
```

Рис 2.8 Зображення вертикальних ліній

Після запуску додатку у нас є можливість редагувати мапу, при виборі кнопки редагування, на екрані зображується сітка, з правого боку, зображуються матеріали, які ми можемо використовувати для створення мапи.

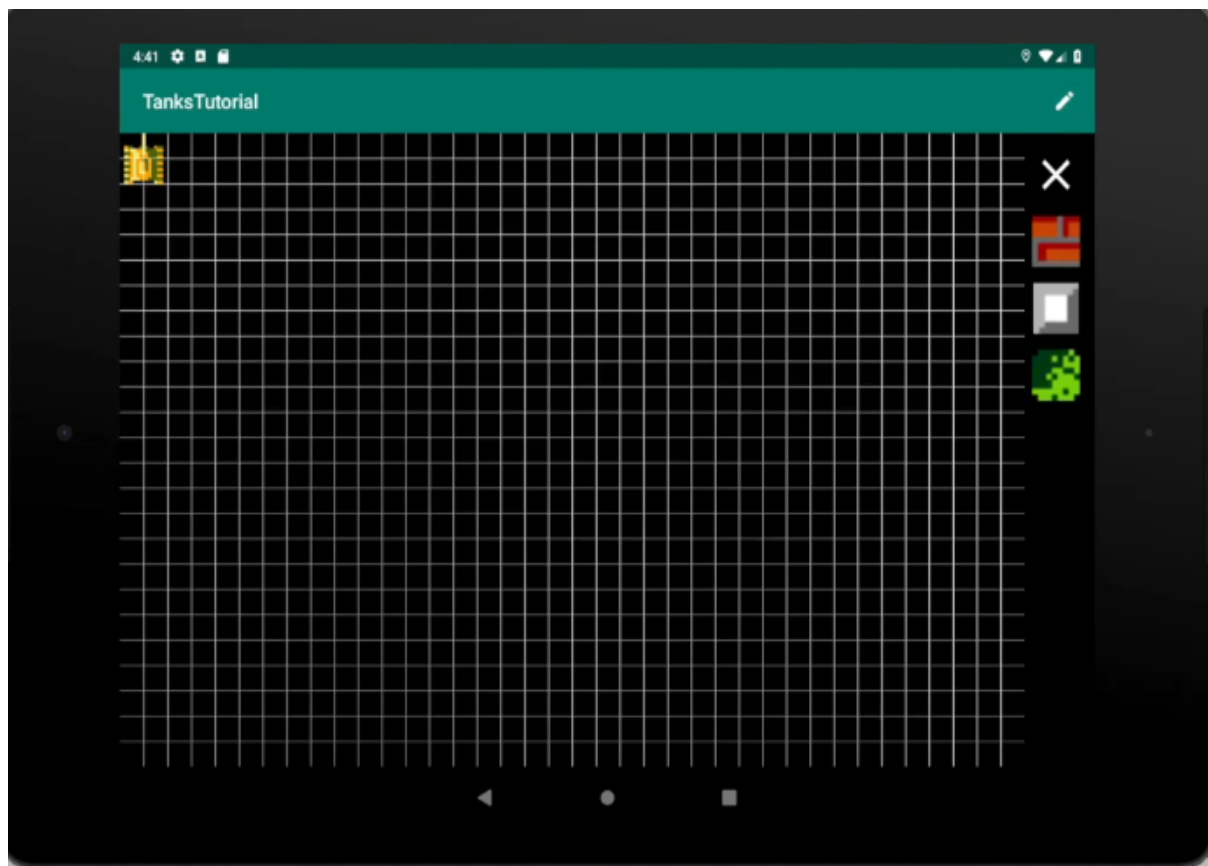


Рис 2.9 Екран з редагуванням мапи.

Танки генеруються з затримкою в 3 секунди з лімітом в 20 танків.

```
private const val MAX_ENEMY_AMOUNT = 20
```

Рис 2.10 Ліміт ворогів

Вороги також з’являються із спеціальних точках в даний проект містить лише 3 точки “Респауну”.

```
private fun getRespawnList(): List<Coordinate> {  
    val respawnList = mutableListOf<Coordinate>()  
    respawnList.add(Coordinate( top: 0, left: 0))  
    respawnList.add(Coordinate( top: 0, left: HALF_WIDTH_OF_CONTAINER - CELL_SIZE))  
    respawnList.add(Coordinate( top: 0, left: VERTICAL_MAX_SIZE - 2 * CELL_SIZE))  
    return respawnList  
}
```

Рис 2.11 Точки відродження танків

Генерація танків має затримку в 3 секунди

```
fun startEnemyCreation() {  
    if (gameStarted) {  
        return  
    }  
    gameStarted = true  
    Thread(Runnable {  
        while (enemyAmount < MAX_ENEMY_AMOUNT) {  
            if (!gameCore.isPlaying()) {  
                continue  
            }  
            drawEnemy()  
            enemyAmount++  
            Thread.sleep( millis: 3000 )  
        }  
    }).start()  
    moveEnemyTanks()  
}
```

Рис 2.12 Затримка для генерації ворогів

Танки рухаються прямо але можуть зробити поворот з частотою 1 раз в 400 мілісекунди.

```
private fun moveEnemyTanks() {
    Thread(Runnable {
        while (true) {
            if (!gameCore.isPlaying()) {
                continue
            }
            goThroughAllTanks()
            Thread.sleep( millis: 400)
        }
    }).start()
}
```

Рис. 2.13 Реалізація повороту ворожого танку.

Ворожі танки із мають 10% шанс для пострілу кулею.

```
private fun goThroughAllTanks() {
    if (tanks.isNotEmpty()) {
        soundManager.tankMove()
    } else {
        soundManager.tankStop()
    }
    tanks.toList().forEach { it: Tank
        it.move(it.direction, container, elements)
        if (checkIfChanceBiggerThanRandom( percentChance: 10)) {
            bulletDrawer.addNewBulletForTank(it)
        }
    }
}
```

Рис 2.14 Постріл ворожого танку.

Коли база знищена, або гг ми підраховуємо число балів яке ґсновується на кількості вбитих танків.

2.5. Обґрунтування та організація вхідних та вихідних даних програми

Вхідними даними розробленого додатку є натискання клавіш на клавіатурі для пересування . Користувач має можливість натискати клавіші для пересування та для стрільби.

Для роботи з додатком користувачу буде потрібен смартфон або планшет.

Вихідними даними програмного додатку є зображення ігрової мапи, зображення ворогів, зображення головного героя .

Натискання на клавіші дозволяють користувачеві пересуватися по мапі, а також стріляти у противників.

2.7. Опис розробленої системи

Під час розробки ігрового додатку були використанні наступні технічні характеристики персонального комп'ютера під операційною системою Windows:

- операційна система (ОС): Windows 10;
- центральний процесор (ЦП):AMD A10 7850k;
- відеоадаптер:NVIDIA GeForce GTX 960;
- відеопам'ять: 4 Гб;
- накопичувач: 520 Гб;
- оперативна пам'ять: 16 Гб.

Для тестування мобільного додатку використовувався емулятор вбудований в Android Studio

При запуску гри, у користувача є можливість відкрити редактор карти, при натисканні на олівець, перед користувачем зображується сітка, а праворуч, зображені елементи карти, які він може використати для створення мапи. Після вибору елемента, користувач натисканням на клітинку може розмістити об'єкт у потрібному місці.

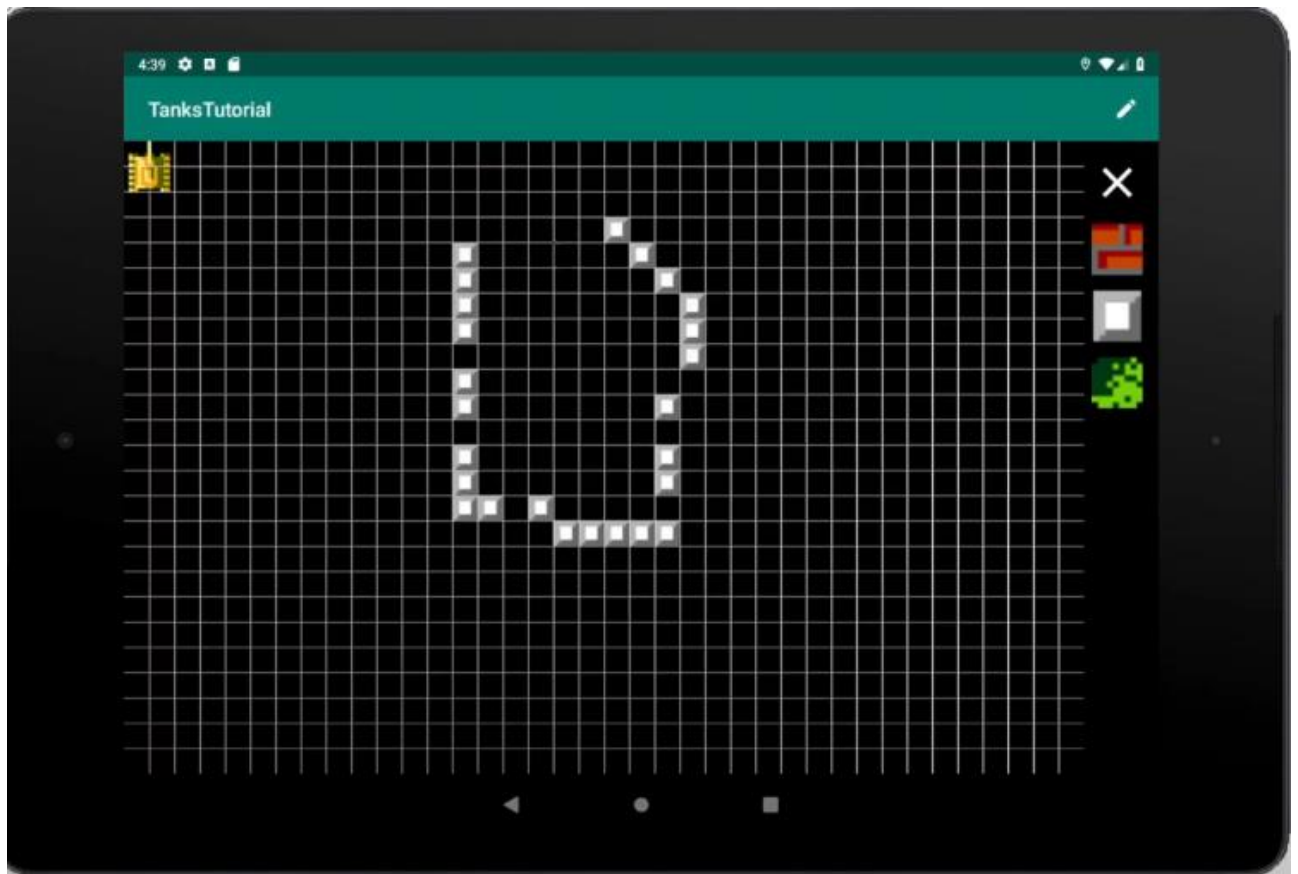


Рис 2.15 Зображення редактору мап

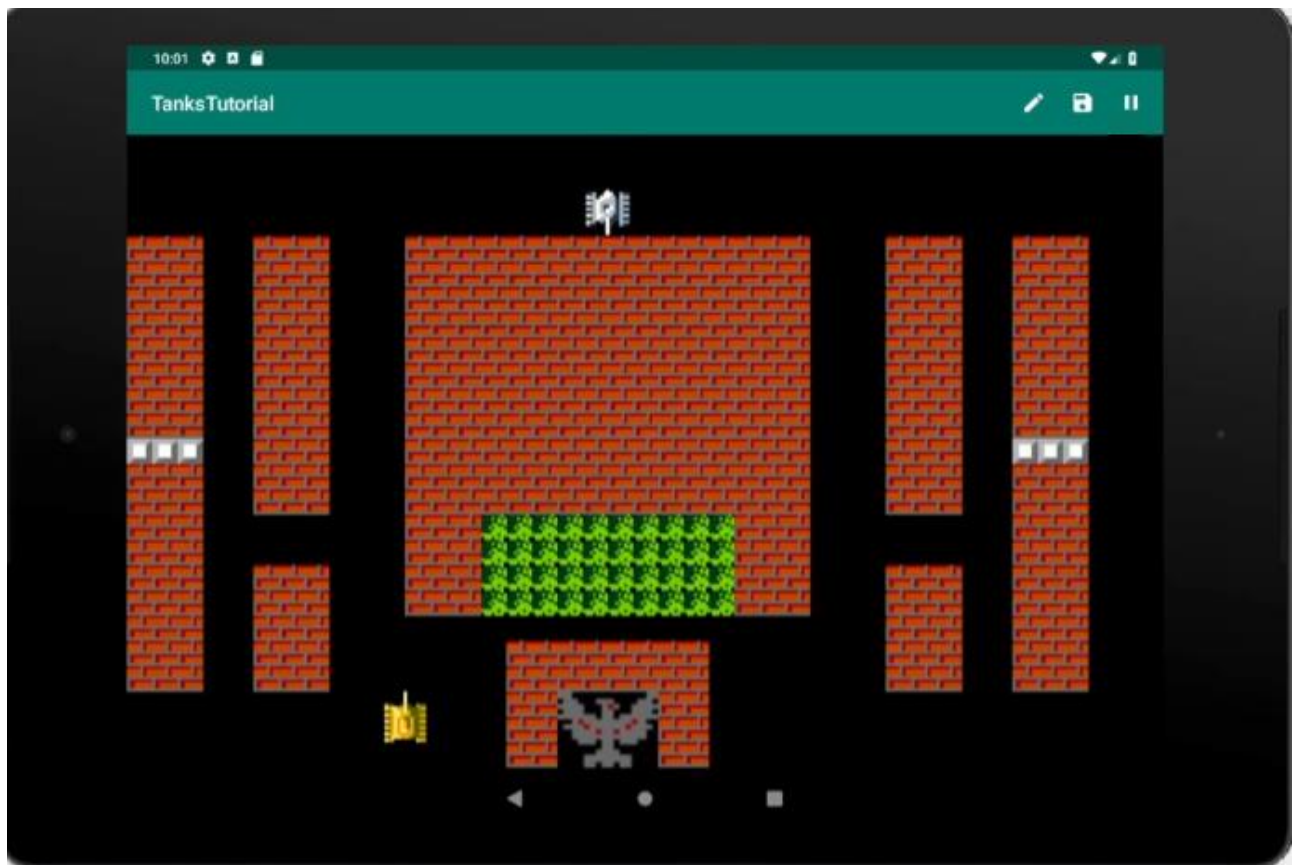


Рис. 2.16 Зображення початку гри

Пересування у грі виконується завдяки натискання на клавіші. При запуску гри, танк користувача зображується у нижній частині мапи. Поряд із ним, зображується і база головного героя у вигляді Орла. Коли користувач натискає клавішу “Стрілка вгору”, танк починає рухатись до верхньої частини мапи. При натисканні клавіші “Стрілка праворуч”, танк повертає на 90 градусів. При натисканні клавіші “Стрілка вниз”, танк повертається до нижньої частини мапи. При натисканні клавіші “Стрілка ліворуч”, танк повертає на 90 градусів у напрямку лівого краю мапи.

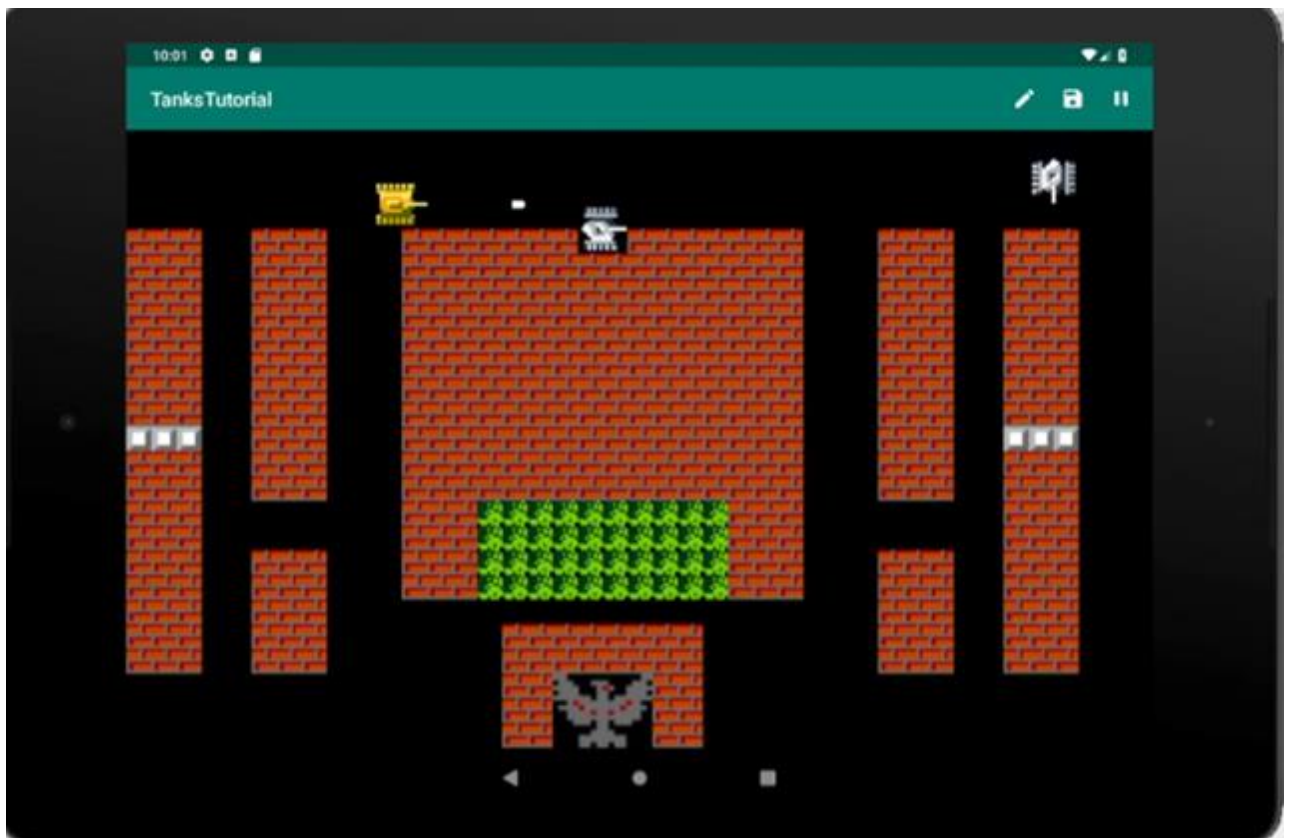


Рис. 2.17 Зображення стрільби головного героя

При натисканні на Space (пропуск), танк яким керує користувач випускає кулю. Куля потрапляючи в стіну, або у ворожий танк зникає, разом із нею зникає і об'єкт в який влучила куля. Куля може знищити ворожий танк, або цеглу, якщо куля влучає у кущі, куля пролітає їх наскрізь. Якщо куля потрапляє у білу цеглу, то куля зникає

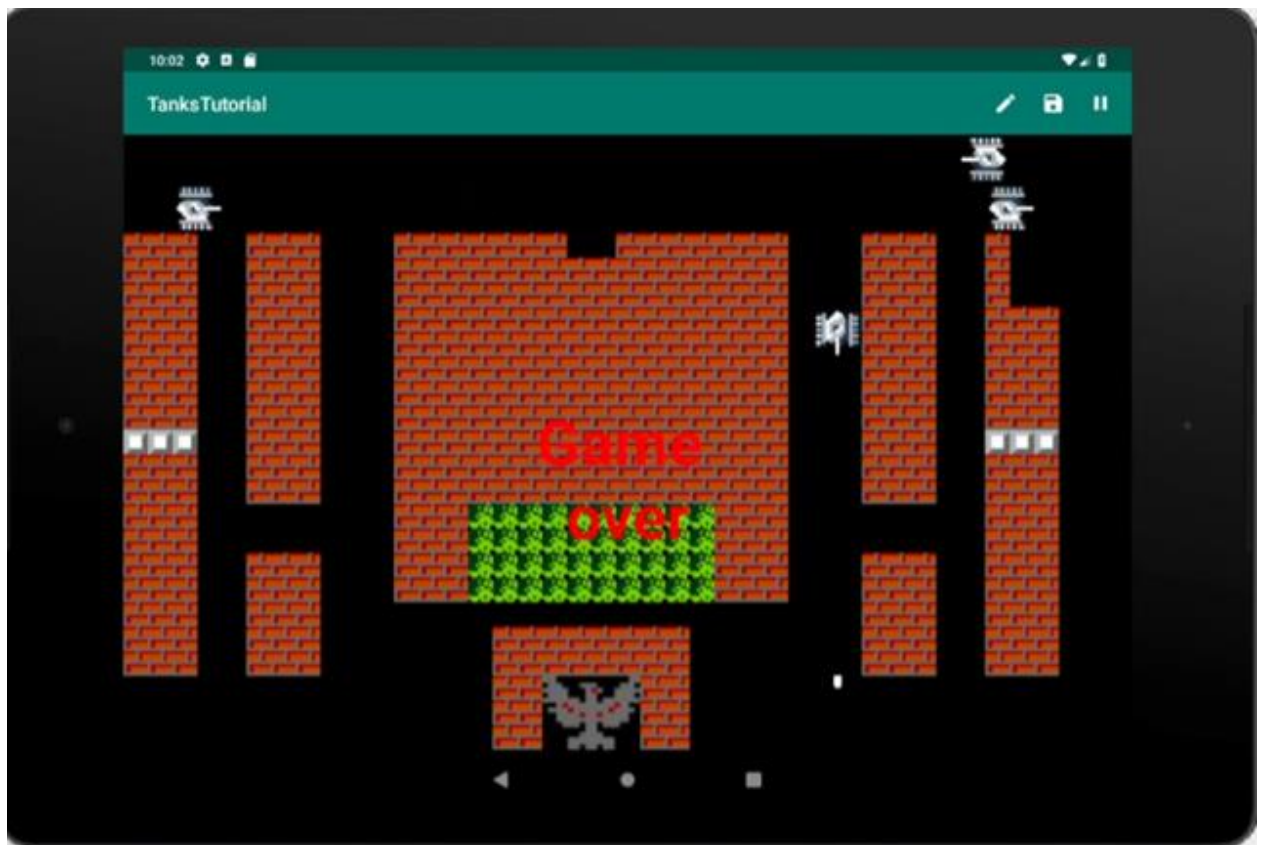


Рис. 2.18 Зображення знищення героя

Після влучання кулі в головного героя, або його базу на екрані зображується напис “Game over” інформуючи гравця про завершення гри.

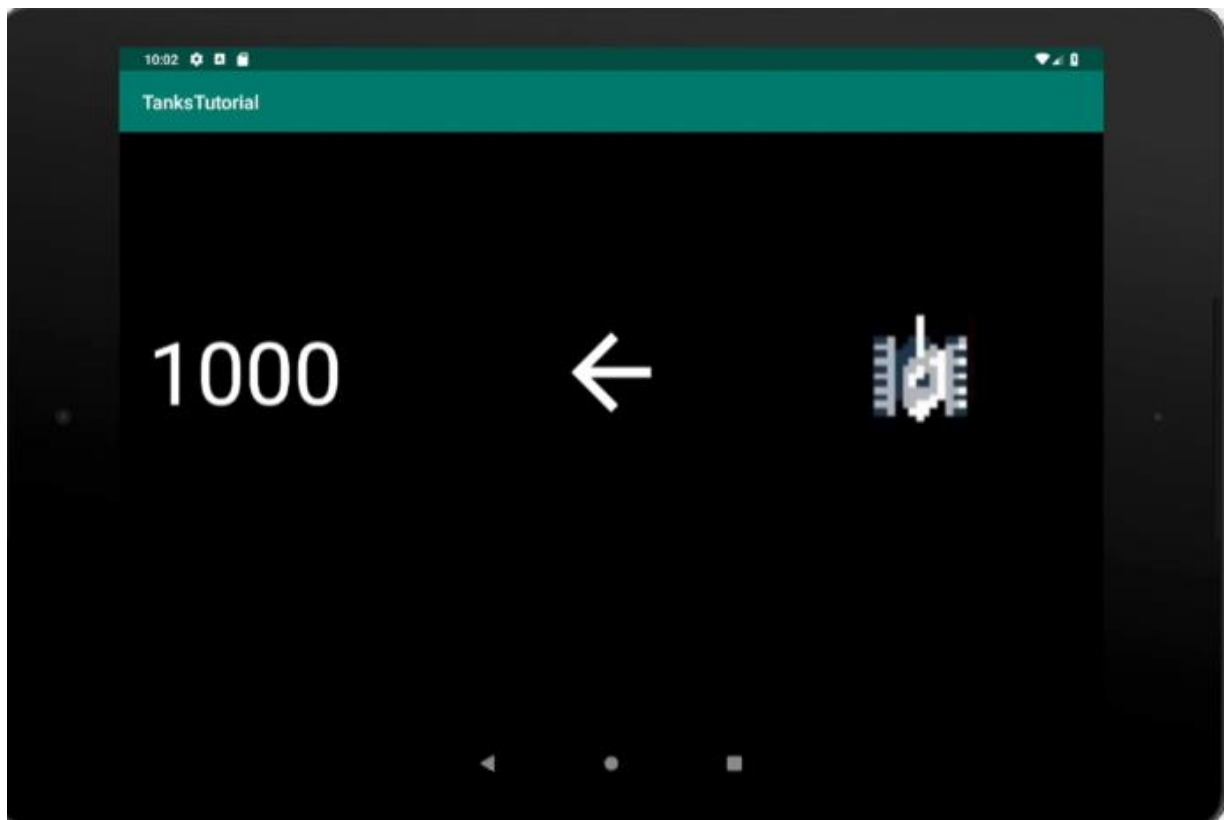


Рис. 2.19 Зображення результатів гри

Підрахунок балів. Після екрану з кінцем гри з'являється наступний екран с підрахунком балів. На екрані зображується ворожий танк, а також кількість балів. За кожний танк, гравцю нараховують 100 балів

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми - 976;
2. коефіцієнт складності програми - 1,4;
3. коефіцієнт корекції програми в ході її розробки - 0,08;
4. годинна заробітна плата розробника - 156 грн/год;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі - 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності - 0,8;
7. вартість машино-години ЕОМ - 16 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_\delta, \quad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

t_{oml} – витрати праці на налагодження програми на ЕОМ;

t_δ – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p)Q = q \cdot C \cdot (1 + p), \quad (3.2)$$

де q – передбачуване число операторів;

C – коефіцієнт складності програми;

p – коефіцієнт кореляції програми в ході її розробки.

$$Q=976 \cdot 1,4 \cdot (1+0,08)=1475,71$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \dots 85) \cdot k}, \text{ людино-годин,} \quad (3.3)$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності;

Витрати праці на розробку алгоритму рішення задачі формулою (3.3):

$$t_u = \frac{1475,71 \cdot 1,2}{78 \cdot 0,8} = 28,37, \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин,} \quad (3.4)$$

де Q - умовне число операторів програми;

k - коефіцієнт кваліфікації програміста.

Підставивши відповідні значення змінних у формулу (3.4):

$$t_a = \frac{1475,71}{25 \cdot 0,8} = 73,78 \text{ , людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k} \text{ , людино-годин,} \quad (3.5)$$

Відповідно до формули (3.5):

$$t_n = \frac{1475,71}{20 \cdot 0,8} = 92,23 \text{ , людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4 \dots 5) \cdot k} \text{ , людино-годин,} \quad (3.6)$$

Підставивши значення до формули (3.6):

$$t_{oml} = \frac{1475,71}{5 \cdot 1,3} = 227,03 \text{ , людино-годин.}$$

– за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml} \text{ , людино-годин,} \quad (3.7)$$

Підставивши значення до формули (3.7):

$$t_{oml}^k = 1,5 \cdot 227,03 = 340,54 \text{ , людино-годин.}$$

Витрати праці на підготовку документації:

$$t_d = t_{dp} + t_{do} \quad , \text{ ЛЮДИНО-ГОДИН,} \quad (3.8)$$

де t_{dp} - трудомісткість підготовки матеріалів і рукопису;

$$t_{dp} = \frac{Q}{(15 \dots 20) \cdot k} \quad , \text{ ЛЮДИНО-ГОДИН,} \quad (3.9)$$

Використавши формулу (3.9):

$$t_{dp} = \frac{1475,71}{18 \cdot 0,8} = 102,47 \quad , \text{ ЛЮДИНО-ГОДИН.}$$

t_{do} - трудомісткість редагування, печатки й оформлення документації.

$$t_{do} = 0,75 \cdot t_{dp} \quad , \text{ ЛЮДИНО-ГОДИН,} \quad (3.10)$$

Відповідно до формули (3.10):

$$t_{do} = 0,75 \cdot 102,47 = 76,85 \quad , \text{ ЛЮДИНО-ГОДИН.}$$

Повертаючись до формули (3.8):

$$t_d = 102,47 + 76,85 = 179,32 \quad , \text{ ЛЮДИНО-ГОДИН.}$$

Тоді виводимо трудомісткість розробки програмного забезпечення за формулою (3.1):

$$t=50+28,37+73,78+92,23+227,03+179,32=650,73, \text{ людино-годин.}$$

У результаті ми розрахували, що в загальній складності необхідно 650,73 людино-годин для розробки даного програмного забезпечення.

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми Зз/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{ПО} = Z_{ЗП} + Z_{МВ} \text{ , грн,} \quad (3.11)$$

де $Z_{ЗП}$ - заробітна плата виконавців, яка визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР} \text{ , грн,} \quad (3.12)$$

де t - загальна трудомісткість, людино-годин;

$C_{ПР}$ - середня годинна заробітна плата програміста, грн/година

Скориставшись формулою (3.12):

$$Z_{ЗП} = 650,73 \cdot 156 = 101513,88, \text{ грн,}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{МВ} = t_{омл} \cdot C_{МЧ} \text{ , грн,} \quad (3.13)$$

де $t_{омл}$ - трудомісткість налагодження програми на ЕОМ, год.

$C_{MЧ}$ - вартість машино-години ЕОМ, грн/год.

Підставивши значення до формули (3.13):

$$Z_{MB} = 227,03 \cdot 16 = 3632,48, \text{ грн,}$$

Тоді витрати на створення ПЗ знайдемо за формулою (3.11):

$$K_{ПО} = 101513,88 + 3632,48 = 105146,36, \text{ грн,}$$

Витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс,} \quad (3.14)$$

де B_k - число виконавців;

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p = 176$ годин).

Відповідно до формули (3.14):

$$T = \frac{650,73}{1 \cdot 176} = 3,69, \text{ міс,}$$

На розробку даного програмного забезпечення потрібно 656,64 людино-годин. Тобто, ймовірна тривалість розробки складатиме для одного розробника 3,69 місяці при стандартному 8-годинному робочому дні та 176-годинному робочому місяці. Очікувані витрати на створення програмного забезпечення при курсі гривні відносно до долара на початок червня який складає 29.25 гри тоді вартість розробки складатиме 105146,36 грн.

ВИСНОВКИ

Виконуючи дану кваліфікаційну роботу було розроблено ігровий додаток. Під час розробки додатку використовувались сучасні підходи для створення програм з можливістю подальшого удосконалення. За необхідністю застосунок може бути розширений новим функціоналом.

Розроблений додаток призначений для відпочинку. Додаток надає можливість керувати ігровим танком, і знищувати противників.

В результаті розробки створено додаток, який може бути використано на практиці, тому що може замінити стандартну клавіатура та надати нові можливості взаємодії з комп'ютером.

Виконуючи цю кваліфікаційну роботу були досягнуті наступні етапи:

- предметна область задачі була проаналізована;
- встановленні підстави та вимоги розробки інформаційної системи;
- визначена структура та обрані технології для створення додатку;
- розробка коду застосунку;

Також під час виконання кваліфікаційної роботи було визначено трудомісткість розробленого програмного продукту (656,64 людино-годин), підраховано вартості роботи по створенню програми (92086,24 грн) та підраховано час на його створення (3.7 міс).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація Kotlin <https://www.jetbrains.com/>.
2. Сайт з туторіалами <https://www.w3schools.com/java/default.asp>
3. «Head First. Kotlin», 2019
4. «Волшебство Kotlin», 2020
5. "Head First Java, Изучаем Java", Кеті Сера, Берт Бейтс
6. Історія відеоігор https://uk.wikipedia.org/wiki/Історія_відеоігор
7. Gradle <https://blog.skillfactory.ru/glossary/gradle/>
8. Аспекти тестування стор <https://smartbear.com/blog/architecture-and-infrastructure-of-mobile-testing/>
9. Загальна інформація про Kotlin <https://uk.education-wiki.com/1625670-what-is-kotlin>
10. Статистика зарплат програмістів, тестувальників і РМ в Україні | DOU <https://jobs.dou.ua/salaries/?period=2021-12&position=Junior%20SE&technology=Python&experience=0->

