

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:
Завідувач кафедри

Сергій СТИРЕНКО

_____ (підпис)

“ _ ” _____ 2021 р.

Дипломний проєкт

на здобуття ступеня бакалавра
за освітньо-професійною програмою “Комп’ютерні системи та мережі”
спеціальності 123 “Комп’ютерна інженерія”

на тему: Ігровий додаток "Шахи" для Android

Виконав (-ла): студент (-ка) 4 курсу, групи ІО-71
(шифр групи)

Кизима Ілля Леонідович

(прізвище, ім’я, по батькові)

_____ (підпис)

Керівник Долголенко Олександр Миколайович

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант (нормоконтроль) Сімоненко Валерій Павлович

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

(підпис)

Київ – 2021 р.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ“КИЇВСЬКИЙ
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальність 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій СТИРЕНКО

(підпис)

“ ” _____ 2021 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студента

Кизими Іллі Леонідовича

1. Тема проєкту Ігровий додаток "Шахи" для Android
керівник проєкту Долголенко Олександр Миколайович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 5 травня _____ 2021 року № 1139-с
2. Термін здачі студентом закінченого проєкту 20 травня 2021 р.
3. Вихідні дані до проєкту технічна документація.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)
Опис предметної області, дослідження методики створення систем автоматичного тестування на базі готових бібліотек, розроблення програмного забезпечення
5. Перелік графічного матеріалу (з точним позначенням обов’язкових креслень)
функціональна схема, принципова схема, структурна схема

6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ п/п	Найменування етапів дипломно-го проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми проєкту</i>	<i>10.12.2020-15.12.2020</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>15.12.2020-15.03.2021</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>15.03.2021-25.03.2021</i>	
4.	<i>Розробка структур окремих підсистем</i>	<i>25.03.2021-5.04.2021</i>	
5.	<i>Програмна реалізація системи</i>	<i>5.04.2021-15.04.2021</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>15.04.2021-20.05.2021</i>	
7.	<i>Захист програмного продукту</i>	<i>25.04.2021</i>	
8.	<i>Передзахист</i>	<i>23.05.2021</i>	
9.	<i>Захист</i>	<i>14.06.2021</i>	

Студент-дипломник _____
(підпис)

Керівник проєкту _____
(підпис)

Анотація

В бакалаврському дипломному проєкті реалізований ігровий додаток "Шахи" для операційної системи "Android", призначений для любителів гри в "Шахи"

Додаток дозволяє грати вдвох на одному пристрої в режимах "бліц" та "класичний", а також грати мережею WIFI Direct в класичному режимі.

Програмний продукт був створений з використанням мови програмування Kotlin та бази даних SQLite.

Annotation

The bachelor's degree project implements the game application "Chess" for the operating system "Android", designed for fans of playing "Chess"

The application can allow you to play on one device in "blitz" and "classic" mode, as well as play WIFI Direct in classic mode.

The software product was created using Kotlin language programming and SQLite database

ТЕХНІЧНЕ ЗАВДАННЯ

до дипломної роботи
освітньо-кваліфікаційного рівня бакалавр

на тему: «Ігровий додаток “Шахи” для Android»

Київ – 2021 року

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ.....	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ	2
3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1. Вимоги до продукту що розробляється.....	3
5.2. Вимоги до програмного забезпечення.....	3
5.3. Вимоги до апаратної частини.....	3
6. ЕТАПИ РОЗРОБКИ	4

					<i>ІАЛЦ.467200.002 ТЗ</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Ігровий додаток “Шахи” для Android Технічне завдання	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Разробив</i>	<i>Кизима І.Л.</i>						<i>1</i>	<i>3</i>
<i>Керівник</i>	<i>Додголенко О.М.</i>							
<i>Реценз.</i>								
<i>Н. Контр.</i>	<i>Сімоненко В.П.</i>							
<i>Затверд.</i>	<i>Стіренко В. Г.</i>					<i>НТУУ “КПІ” ФІОТ ІО-71</i>		

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

Це технічне завдання поширюється на розробку ігрового додатку “Шахи” для Android. Область застосування цього продукту: додаток дозволяє грати в шахи локальною мережею або на одному пристрої.

2. ПРИЧИНИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання бакалаврського дипломного проекту, затверджене кафедрою обчислювальної техніки Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є розробка ігрового додатку “Шахи” для Android.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література з технічних питань та статі в інтернеті за даною темою.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до розробленого продукту

- Інтуїтивно зрозумілий інтерфейс
- Широка доступність для цільової групи
- Можливість легкого розширення функціоналу
- Технічна коректність

5.2. Вимоги до програмного забезпечення

5.1. Вимоги до розробленого продукту

- Інтуїтивно зрозумілий інтерфейс
- Можливість легкого розширення функціоналу
- Самодостатність
- Технічна коректність і актуальність інформації

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Зм.	Лист	№ докум.	Підпис	Дата		

5.2. Вимоги до програмного забезпечення

Android 8.0 і вище

6. ЕТАПИ РОЗРОБКИ

	Дата
6.1. Вивчення необхідної літератури	21.02.2021
6.2. Складання і узгодження технічного завдання	08.03.2021
6.3. Написання вступної частини та огляд рішень	22.03.2021
6.4. Розробка архітектури системи	04.04.2021
6.5. Написання програмної частини	20.04.2021
6.6. Тестування та виправлення помилок	21.05.2021
6.7. Оформлення документації дипломного проекту	26.05.2021

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		3

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту

**на тему: «Система реєстрації та обліку студентів на лабораторних
заняттях»**

Київ – 2021 року

ЗМІСТ

ВСТУП.....	3
Розділ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ	4
1.1. Особливості додатків для гри в шахи.....	4
1.2. Сутність мобільних додатків.....	7
1.3. Аналіз існуючих шахових додатків.....	8
1.3.1. Шахи від Chees Prince.....	8
1.3.2. Шахи від AI Factory Limited.....	9
1.3.3. Шахи від Chess Friends.....	10
1.3.4. Шахи від Pottes Dorel.....	11
1.3.5. Шахи від Thaiyalnayagi Gandhi.....	12
1.4. Порівняння вище оглянутих додатків.....	12
1.5. Загальний опис функціоналу майбутнього додатку.....	13
1.5.1. Гра вдвох.....	13
1.5.2. Гра локальною мережею.....	14
1.5.3. Годинник.....	14
1.5.4. Статистика.....	14
Висновки до розділу 1.....	15
Розділ 2. ПРОЄКТУВАННЯ ІГРОВОГО ДОДАТКУ “ШАХИ” ДЛЯ ANDROID.....	16
2.1. Огляд ОС “Android”.....	16
1.1.1. Сервіси (Service).....	16
2.1.1. Приймач трансляції (Broadcast receiver).....	18
2.1.2. Провайдери контенту (Content providers).....	18
2.1.3. Намір (Intent).....	18
2.1.4. Активність та її життєвий цикл.....	19
2.1.5. Фрагмент та його життєвий цикл.....	20
2.1.6. Макети (Layout).....	21
2.1.7. Відображення (View).....	22
2.1.8. SQLite.....	23
2.2. Огляд архітектурних паттернів що використовуються у мобільних додатках.....	24

<i>ІАЛЦ.467200.003 ПЗ</i>								
					<i>Ігровий додаток “Шахи” для Android Зміст</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Разробив</i>		<i>Кизима І.Л.</i>					1	57
<i>Керівник</i>		<i>Долголенко О.М.</i>						
<i>Реценз.</i>								
<i>Н. Контр.</i>		<i>Сімоненко В.П.</i>			<i>НТУУ “КПІ” ФІОТ Ю-71</i>			
<i>Затв.</i>		<i>Стіренко В. Г.</i>						

2.2.1. Model-View-Controller (MVC).....	24
2.2.2. Model-View-Presenter (MVP).....	25
2.2.3. Model-View-ViewModel (MVVM).....	26
2.2.4. Model-View-Intent (MVI).....	27
2.3. Вибір технологій для розробки.....	28
2.3.1. Вибір мови програмування для розробки клієнтської частини.....	28
2.3.2. Вибір технології для гри локальною мережею.....	30
Висновки до розділу 2.....	33
Розділ 3. Розробка ігрового додатку “Шахи” для Android.....	34
3.1. Засоби розробки.....	34
3.2. Загальний опис структури Android проекту.....	34
3.2.1. Файл маніфесту (AndroidManifest.xml).....	35
3.2.2. Директорія ресурсів Android (res).....	36
3.2.3. Система збірки проектів Gradle.....	37
3.3. Опис додатку.....	41
3.3.1. Користувацький інтерфейс.....	41
3.3.2. Ігрова логіка.....	46
3.3.3. База даних.....	50
3.3.4. Передача даних мережею Wi-Fi direct.....	51
Висновки до розділу 3.....	53
Висновки.....	54
Джерела.....	55

ВСТУП

Кількість користувачів ОС “Android” безупинно зростає, а разом з нею і популярність мобільних застосунків. Так в третьому кварталі 2020 року кількість завантажень з “Google play” склала 28,3 мільярди, що на 31% більше ніж за аналогічний період 2019 року. І саме ігри є найпопулярнішою категорією – 13,49% від всіх завантажень.

Шахи – абстрактна стратегічна гра на спеціальній дошці, що має назву шахівниця й поділена на 64 світлі та темні клітини (поля), між 16 світлими (білими) і 16 темними (чорними) фігурами за встановленими для них правилами пересування[1].

Перші згадки про прототипи цієї гри датуються 3 – 4 тисячоліттям до нашої ери, а в 18-му столітті шахи набули особливої популярності, що не зменшується і донині. А в 20-му столітті почались перші спроби створення математичних алгоритмів та комп’ютерних програм для шахів. І вже в 1996 році комп’ютер Deep Blue фірми ІВМ вперше обіграв тодішнього чемпіона світу з шахів – Гаррі Каспарова.

					ІАЛЦ.467200.003 ПЗ	Арк.
						3
Зм.	Лист	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

В даному розділі будуть розглянуті особливості ігрових додатків для гри в шахи та вже існуючі мобільні застосунки.

1.1. Особливості додатків для гри в шахи

Шахи — високоінтелектуальна гра, що розвиває вміння швидко приймати рішення та продумувати свої кроки та можливі ходи суперника наперед. На сьогодні популярність гри в «Шахи» точно оцінити неможливо. В базі ФІДЕ (Міжнародна шахова федерація – фр. Fédération Internationale des Échecs) зареєстровано понад 360 тисяч активних гравців, що беруть участь в змаганнях, а “загальний список гравців” кваліфікаційної комісії Федерації шахів України містить близько дванадцяти тисяч учасників. При цьому кількість любителів що не приймають участь в змаганнях оцінити неможливо.

Перші спроби розробки алгоритму для гри в шахи належать Алану Тьюрингу. В 1951 році він написав алгоритм за яким машина могла би грати в шахи, але через відсутність доступу до комп’ютера алгоритм жодного разу не перевірявся в роботі, а людині потрібно було близько тридцяти хвилин для того щоб зробити один хід.

Наступного 1952 року в ядерній лабораторії Лос-Аламоса була розроблена програма для гри в шахи на шахівниці 6х6 та без слонів. Машина за 12 хвилин розраховувала 4 напівходи (один хід однієї сторони). Партія проти професійного гросмейстера тривала 10 годин та закінчилась поразкою комп’ютера, але машина змогла обіграти новачка за 23 ходи , що вже було досягненням.

Машина “Belle” (рис 1.1) стала першою яка досягла рівня шахового майстра. Це був перший комп’ютер що проектувався

					ІАЛЦ.467200.003 ПЗ	Арк.
						4
Зм.	Лист	№ докум.	Підпис	Дата		

тільки для гри в шахи. Її розробка остаточно була завершена 1983 року Джо Кондоном та Кеном Томпсоном в Bell Labs. Але ця машина виграла чемпіонат світу по комп'ютерним шахам ще 1980 року, а також п'ять разів отримувала перемогу в чемпіонаті АСМ (Association for Computing Machinery) по комп'ютерним шахам.

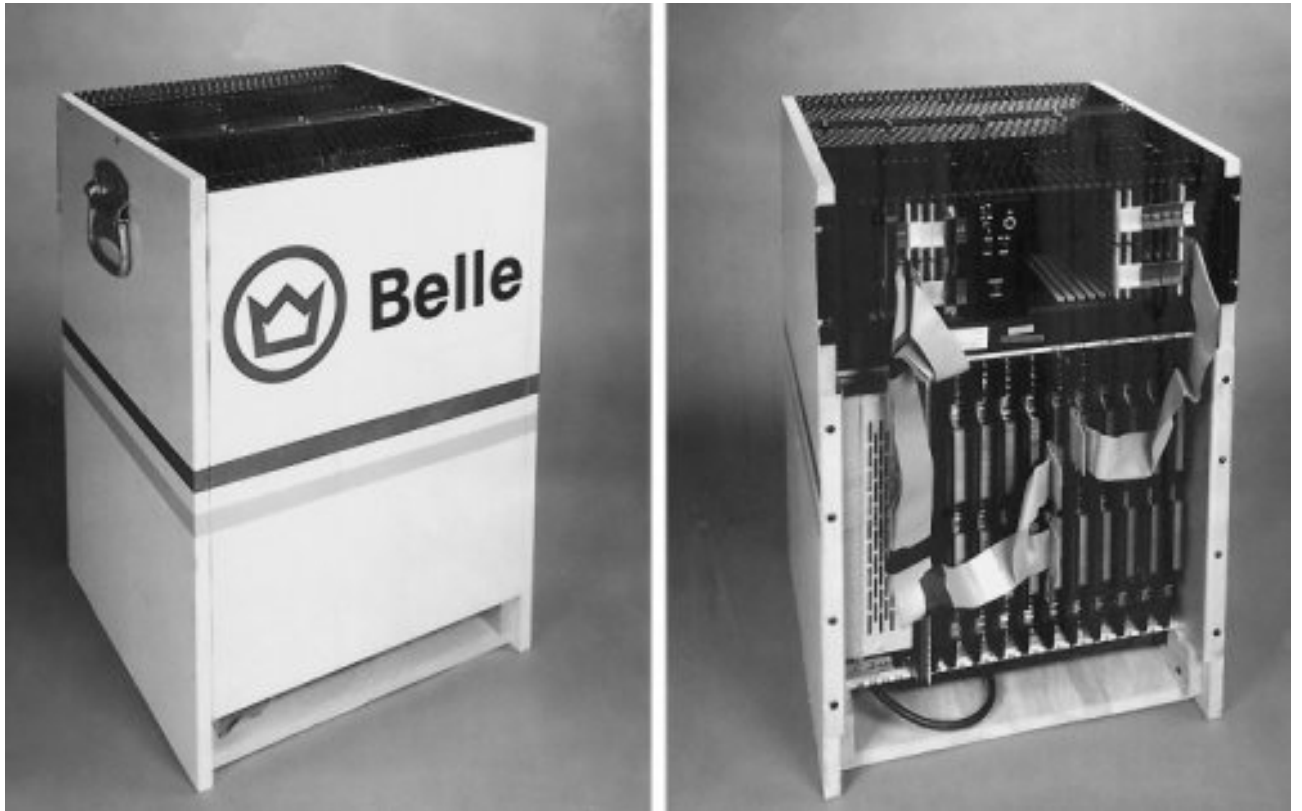


Рис 1.1 Машина для гри в шахи - Belle

Розробка програми Fritz завершилась 1991 року. 1994 року третя версія цієї програми обіграла тодішнього чемпіона світу шахів - Гаррі Каспарова в турнірній блиц-партії, щоправда в другому матчі машина програла чотири рази та двічі зіграла в нічию.

В лютому 1996 машина Deep Blue (рис 1.2) від компанії ІВМ програла Гаррі Каспарову з рахунком 4-2, але комп'ютер виграв першу партію та став першою машиною що обіграла діючого чемпіона світу з шахів в турнірних умовах. Машина являла собою суперкомп'ютерний кластер, що складається з двох стійок в кожній з яких розміщено 30 вузлів на базі робочих станцій RS/6000. На

кожному вузлі був встановлений процесор та дві плати. Два вузла використовували 135-МГц процесори, інші - 120-МГц. На кожному вузлі було встановлено 2ГБ оперативної пам'яті та 4 ГБ дискової. Кожен спеціалізований шаховий процесор робив на частоті 24 МГц, та перебирав від 2 до 2,5 шахових позицій в секунду. Програмне забезпечення було написане мовою С та працювало під управлінням операційної системи АІХ 4.2.



Рис 1.2 Шаховий суперкомп'ютер - Deep Blue

1.2. Сутність мобільних додатків

Мобільний додаток – програмне забезпечення розроблене для роботи на мобільних пристроях. Під час розробки мобільного застосунку необхідно враховувати конструктивні та програмні особливості мобільного пристрою такі як: процесор слабший ніж на звичайному персональному комп'ютері, робота від акумулятора, постійна можливість перервати виконання програми у разі надходження дзвінка, сповіщення, переходу до іншої програми тощо.

Перші мобільні додатки були завчасно встановлені та їх неможливо було видалити. Першою мобільною грою став тетріс в 1994 році на пристрої Nagenuk MT-2000. А вже 1999 року з'явилась Nokia 7110s з технологією WAP, що дозволяла завантажувати додаткові данні в тому числі й різні додатки. Також це дозволило створювати застосунки розробникам не пов'язаним з виробниками мобільних пристроїв.

З появою першого цифрового магазину “App Store”, на iPhone з операційною системою IOS, 2008 року у розробників з'явилась можливість викладати свої застосунки та отримувати 70% прибутку від них. Того ж року з'явився аналог для ОС Android — Android Market.

Android (укр. Андроїд) — операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux[2].

iOS — мобільна операційна система для смартфонів, електронних планшетів, носяться програвачів і деяких інших пристроїв, що розробляється і випускається американською компанією Apple[3].

					ІАЛЦ.467200.003 ПЗ	Арк.
						7
Зм.	Лист	№ докум.	Підпис	Дата		

74,43% всіх проданих мобільних пристроїв протягом перших трьох кварталів 2020 року саме на операційній системі “Android”, а IOS зайняла всього 24,99% ринку.

1.3. Аналіз існуючих шахових додатків

На даний момент існує багато мобільних додатків для гри в шахи, кожен з них має свої переваги та недоліки. Нижче будуть розглянуті найпопулярніші мобільні додатки для гри в шахи.

1.3.1. Шахи від Chees Prince

Додаток є повністю безкоштовним. Має зручний та інтуїтивно зрозумілий користувацький інтерфейс. В додатку присутні 10 рівнів складності, а також 100 шахових задач. Реалізована можливість гри вдвох на одному пристрої. Наявне підсвічування можливих ходів обраною фігурою (Рис. 1.3) З мінусів можна відмітити відсутність гри мережею, неможливість зберігати партію для продовження під час наступного запуску додатку та відсутність режиму бліц. Також наявна реклама.



Рис. 1.3. Підсвічування можливих ходів в реалізації від Chees Prince

1.3.2. Шахи від AI Factory Limited

В даній реалізації також присутній вибір складності (тут їх 12). Можливі підказки у вигляді підсвічування всіх полів на які можна “піти” обраною фігурою та підсвічування фігури рекомендованої для ходу та тих що знаходяться “під боєм”. Це спрощує гру для початківців-любителів, але гальмує подальше навчання. При створенні гри використаний шаховий рушій Treebeard. Також присутні ведення статистики перемог та поразок; налаштування гри, а саме фора білої сторони та час гри та додатковий час(Рис. 1.4); синхронізація з Google Games; гра вдвох на одному пристрої; можливість збереження та огляду партії. Але можливість грати мережею все ще знаходиться на стадії бета тестування.



Рис. 1.4. Налаштування гри в реалізації від AI Factory Limited

1.3.3. Шахи від Chess Friends

Сервіс має одну з найбільших спільнот, приблизно 1 600 000 зареєстрованих гравців. Реалізовані можливості гри з комп'ютером та гри онлайн в різних режимах: класичний, бліц, швидкий. В грі наявні підсвічування полів на які можна походити обраною фігурою. Також можливий огляд партії після гри. Присутня система рейтингів та рівнів гри. Кидається в вічі застарілий дизайн. Також серед недоліків є неможливість зберегти незавершену гру. Приклад гри наведений нижче(Рис. 1.5).

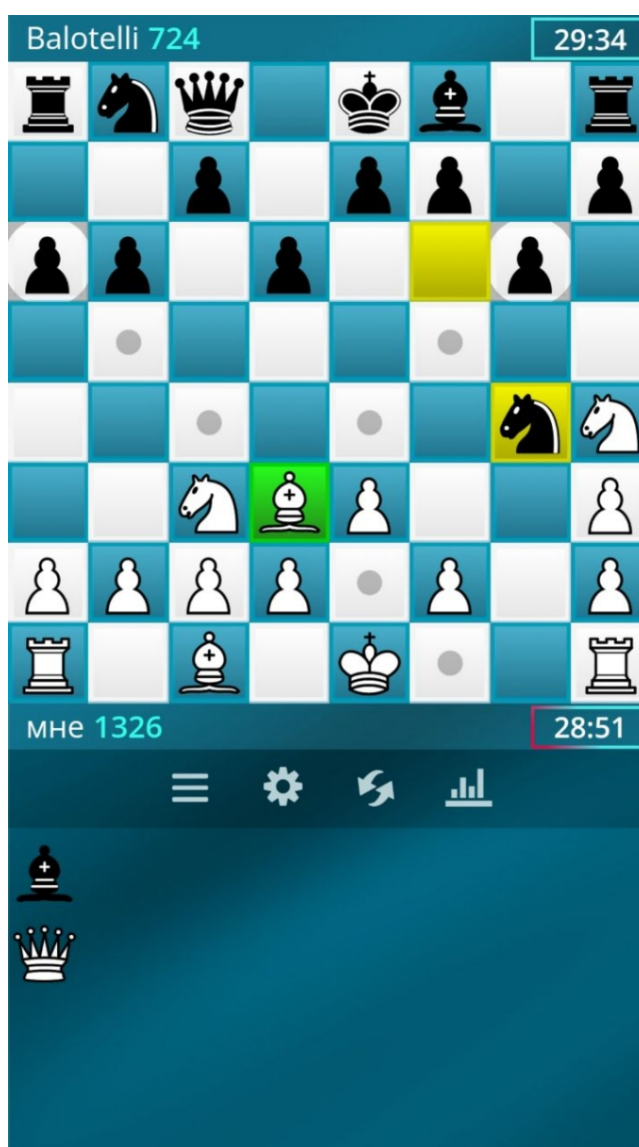


Рис. 1.5. Приклад гри в реалізації від Chess Friends

1.3.4. Шахи від Pottes Dorel

В додатку є різні режими гри такі як: гравець проти комп'ютера, гравець проти гравця(на одному пристрої), головоломки. Незавершена гра автоматично зберігається та завантажується з початком наступного запуску. З недоліків можна виділити відсутність різних видів шахів(бліц, швидкісний), огляд партії після завершення гри, та також відсутність можливості гри мережею. Приклад гри наведено на рисунку 1.6



Рис. 1.6. Приклад гри в реалізації від Pottes Dorel

1.3.5. Шахи від Thaiyalnayagi Gandhi

В цьому додатку присутні режими лише класичної гри вдвох на одному пристрої та по Bluetooth. Сама реалізація гри по Bluetooth не досконала: застосунок часто закривається, є недоліки в реалізації правил наприклад короля можна “з’їсти”, а рокіровку зробити через биті поля. Приклад наведено на рисунку 1.7.

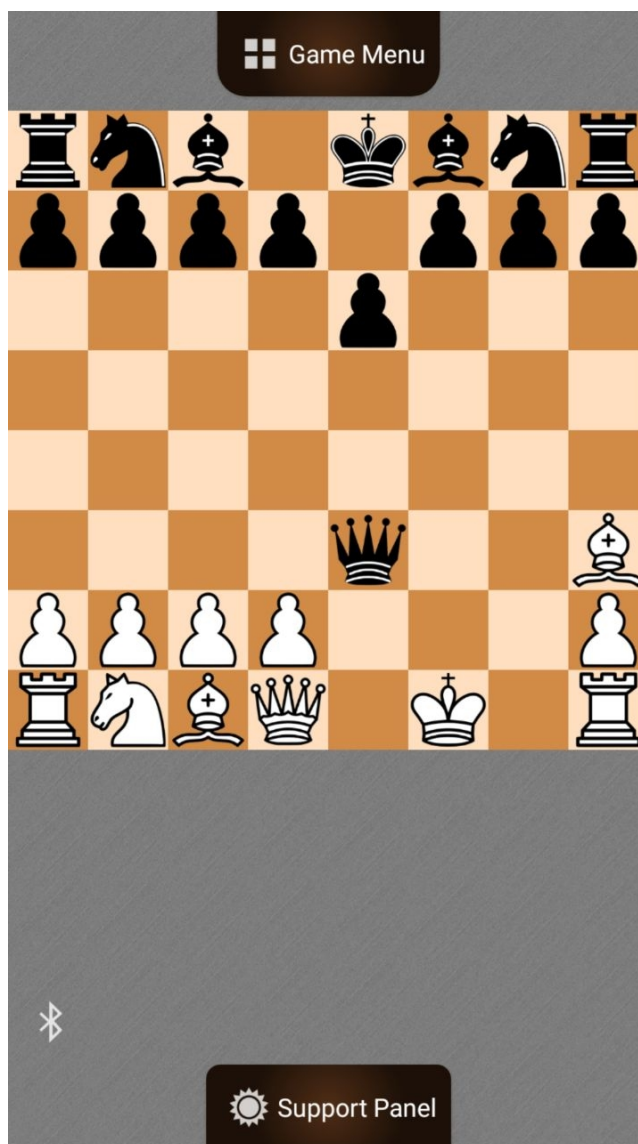


Рисунок 1.7 Приклад гри в реалізації від Thaiyalnayagi Gandhi

1.4. Порівняння вище оглянутих додатків

В таблиці 1.1 представлено порівняння вище оглянутих додатків

Таблиця 1.1 - порівняння ігрових додатків для гри в шахи

Розробник	Chees Prince	AI Factory Limited	Chess Friends	Pottes Dorel	Thaiyalnayagi Gandhi
Гра мережею Інтернет	-	+	+	-	-
Гра локальною мережею	-	-	-	-	+
Реалізація гри в бліц режимі	-	+	+	-	-
Можливість огляду партій	-	+	+	-	-

Як бачимо гра локальною мережею реалізована лише в одному з оглянутих додатків який має свої недоліки: помилки під час виконання програми та відсутність можливості огляду зіграних партій. А гра мережею Інтернет реалізована в двох доволі популярних застосунках, , в яких цей режим не має явних недоліків.

1.5. Загальний опис функціоналу майбутнього додатку

1.5.1. Гра вдвох

Два гравці грають на одному пристрої в різних режимах, наприклад класичний або бліц. Також пропонується реалізувати режим редагування дошки, в якому не діятимуть правила, таким чином можна буде самостійно розставити фігури для моделювання, обговорення або розв'язування шахових задач, або щоб встановити “фору”, тобто одна із сторін може грати меншою чи більшою кількістю фігур.

1.5.2. Гра локальною мережею

Серед додатків оглянутих вище є багато таких що реалізують гру з використанням мережі Internet, при цьому реалізація додатків для гри локальною мережею залишає бажати кращого, тому пропонується реалізувати в розроблюваному застосунку гру з використанням локальної мережі. Грати вдвох на одному пристрої може бути не зручно, а доступ до всесвітньої мережі є не завжди та не всюди. Отже ця функція буде корисною для компаній які багато подорожують або часто виїжджають на територію де відсутній Інтернет.

1.5.3. Годинник

Дана функція буде реалізована як додаток для гри в бліц з використанням матеріальних шахів. Перед запуском годинника необхідно ввести час гри та додатковий час. Обидва гравці бачитимуть скільки часу залишилось.

1.5.4. Статистика

Для можливості огляду зіграних партій в додаток буде додана статистика, в якій відобразатиметься нотація зіграних ігор.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		14

ВИСНОВКИ ДО РОЗДІЛУ 1

У цьому розділі були розглянуті особливості ігрових додатків для гри в “Шахи” на ОС “Android”. Також був здійснений огляд застосунків для цієї гри. Проаналізувавши існуючі реалізації можна зробити висновок, що на сьогоднішній день існує багато різноманітних додатків, але в кожному з них є свої недоліки. В більшості з них добре реалізована гра мережею Інтернет, або гра в двох на одному пристрої, але не існує хорошої реалізації для гри локальною мережею без доступу до Інтернету.

					ІАЛЦ.467200.003 ПЗ	Арк.
						15
Зм.	Лист	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІГРОВОГО ДОДАТКУ

“ШАХИ” ДЛЯ ANDROID

2.1. Огляд ОС “Android”

Виконуваний код Андроїд це байт-код для віртуальної машини Dalvik (DVM). Dalvik – віртуальна машина заснована на регістрах, розроблена як частина ОС “Android” та є середовищем виконання компонентів ОС та користувацьких додатків. DVM виконує байт-код власного формату, відмінного від Віртуальної машини Java, хоча ПЗ створюється саме на Java. Для кожного додатку або компоненту ОС DVM створює окремий захищений процес. Додаток Android може містити чотири основні компоненти: сервіси (Service), активності (Activity), приймачі трансляції (Broadcast Receiver) та провайдери контенту (Content provider).

1.1.1. Сервіси (Service)

Сервіс - компонент програми який виконує тривалі операції у фоновому режимі.[7] Служба може продовжувати працювати навіть після переходу до іншого додатку. Також є можливість прив'язати компонент до сервісу для того щоб взаємодіяти з ним. Сервіси поділяються на три типи:

- Передньоплановий (Foreground): відтворює операцію яка помітна користувачеві. Обов'язково має відобразитися у сповіщенні. Як приклад можна навести музичний додаток, що відтворює аудіофайл та відображає у сповіщенні його, або крокомір, щ працює в фоні та відображає кількість кроків.
- Фоновий (Background): виконується без сповіщення, наприклад ущільнення сховища програми.
- Зв'язний (Bound): компонент додатку прив'язаний до служби.

					ІАЛЦ.467200.003 ПЗ	Арк.
						16
Зм.	Лист	№ докум.	Підпис	Дата		

Служба описується в підкласі Service, в перевизначених методах:

- onCreate() - викликається якщо сервіс створено операційною системою
- onStartCommand() - викликається системою якщо запуск служби вимагає інший компонент.
- onBind() - викликається якщо інший компонент хоче прив'язатися до зв'язної служби.
- onBind() - викликається якщо служба завершує зв'язування з іншими компонентами.
- onBind() - викликається якщо нові клієнти підключаються до послуги
- onDestroy() - викликається перед тим як служба знищується

Схема життєвого циклу зв'язного та незв'язного сервісу наведена на рисунку 2.1.

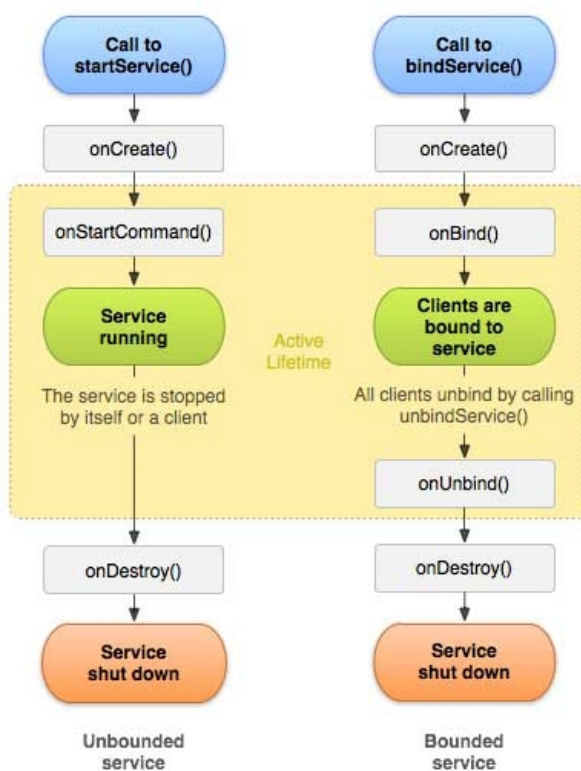


Рис 2.1 - Життєвий цикл сервісу

2.1.1. Приймач трансляції (Broadcast receiver)

Приймач трансляції - компонент сповіщає додаток про події які відбулись за його межами, тим самим надаючи можливість застосунку реагувати на загальносистемні повідомлення. Найчастіше надає API іншого процесу. Таким чином система знає які процеси використовуються іншими компонентами та не знищує їх.

2.1.2. Провайдери контенту (Content providers)

Провайдер контенту управляє набором даних додатків які зберігаються в файловій системі пристрою, базі даних SQLite, веб-сервері або будь-якому іншому постійному сховищі даних до якого має доступ додаток. Завдяки такому зберіганню даних інші застосунки можуть отримати до них доступ. Файли в Android іменуються за схемою URI - рядок символів за яким ідентифікується ресурс. Для системи провайдер контенту - точка входу в додаток для публікації імених елементів даних. Таким чином створюється можливість використовувати компоненти інших програм. Наприклад надсилання повідомлення може здійснюватися за допомогою будь-якого із встановлених месенджерів, відповідно зникає потреба в реалізації власного. Для цього непотрібно посилатися на код іншого додатку, а просто запустити активність месенджера в якій відбувається надсилання повідомлень.

2.1.3. Намір (Intent)

Намір - це абстрактний опис операції, яку потрібно виконати, різновид повідомлень який дозволяє пов'язувати різні об'єкти. Він може використовуватися для запуску активності (Activity), зв'язки з фоновим сервісом (Service), його можна надіслати до приймача трансляції (BroadcastReceiver).

Намір надає можливість виконувати прив'язку виконання між кодом у різних програмах. Найбільш значне його використання - у запуску активності, де це можна вважати взаємодією між активностями. Це пасивна структура даних, що містить абстрактний опис дії, яку потрібно виконати.

2.1.4. Активність та її життєвий цикл

Активність (Activity) – окремий екран в Андроїд. Кожен клас активності успадковується від базового класу Activity. Далі опишемо життєвий цикл активності:

1. При першій ініціалізації викликається метод onCreate().
2. Коли активність готова та відображається користувачеві викликається метод onStart(),
3. Активність переходить на передній план (в фокус) що описано в методі onResume().
4. Коли фокус втрачається викликається метод onPause().
5. Метод onStop() викликається коли активність більше не видно(наприклад коли відображається інша активність).
6. Перед завершенням активності викликається метод onDestroy().

Графічне представлення життєвого циклу активності наведено на рис. 2.2.

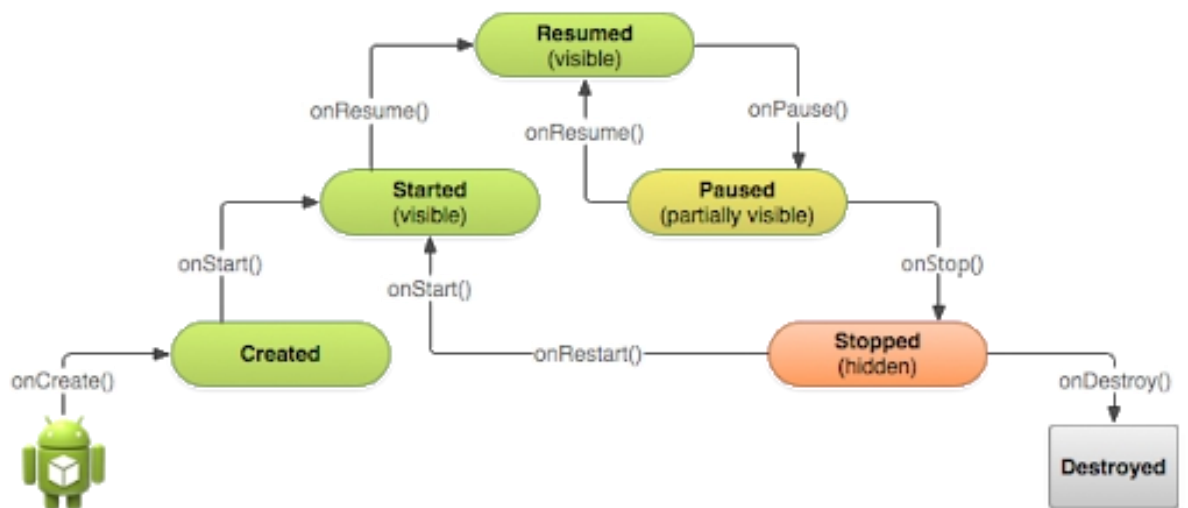


Рис. 2.2 Життєвий цикл Android активності

2.1.5. Фрагмент та його життєвий цикл

Для кращого вигляду додатку на великих екранах була розроблена концепція фрагментів. Фрагмент – частина візуального інтерфейсу додатку що може повторно багаторазово використовуватися. Фрагмент існує лише в контексті активності. Одна активність може містити в собі декілька фрагментів. Життєвий цикл фрагмента складається з наступних етапів:

1. Коли фрагмент прикріплюється до певної активності, у фрагмента викликається метод `onAttach()`.
2. В методі `onCreate()` описано створення фрагменту.
3. В методі `onCreateView()` створюється візуальний інтерфейс фрагменту.
4. Метод `onViewCreated()` викликається після створення візуального інтерфейсу.
5. Коли фрагмент стає активним викликається метод `onStart()`.
6. Коли фрагменту передається фокус викликається `onResume()`.
7. Після виконання методу `onPause()` фрагмент стає неактивним, але залишається видимим.

8. В методі `onDestroyView()` знищується візуальна частина фрагменту,
9. а в `onDestroy()` – остаточно знищується увесь фрагмент.
10. Метод `onDetach()` викликається коли фрагмент відкріплюється від активності.

Схематичне зображення життєвого циклу фрагменту наведено на рис. 2.3

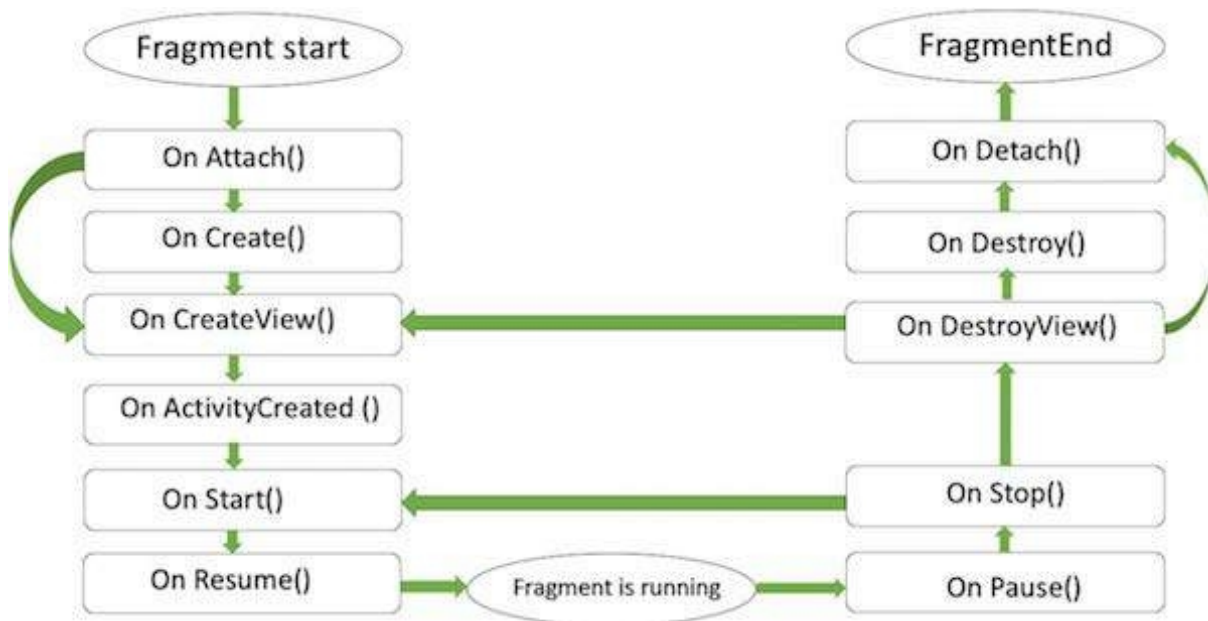


Рис. 2.3 Життєвий цикл Android фрагменту

2.1.6. Макети (Layout)

В програмному забезпеченні для ОС андроїд макет це опис візуальної частини додатку в форматі XML та може містити в собі інші зображення. Всі макети є субкласами класу `android.View.ViewGroup`. Макет може описувати як фрагменти так і активності. Макети поділяються на:

- Лінійні (`LinearLayout`), – група зображень, що вирівнює всі дочірні елементи в одному напрямку.
- Відносний (`RelativeLayout`) – група зображень, що відображає дочірні зображення у відносних позиціях.

- Табличний (TableLayout) – зображення що містить в собі інші зображення згруповані в рядки та стовпці.
- З абсолютним плануванням (AbsoluteLayout) – зображення в якому вказується точне положення його дочірніх елементів.
- З кадровою структурою (FrameLayout) – зображення що містить в собі декілька дочірніх зображень, розташованих шарами (один поверх іншого).
- Списковий (ListView) – група зображень, яка відображає список елементів що прокручуються.
- У вигляді сітки (GridView) – група елементів, що відображає елементи в двомірній сітці з можливістю прокрутки.

2.1.7. Відображення (View)

Всі відображення є субкласами класу android.View.View. Це можуть бути зображення, надписи, кнопки або групи зображень (макети). Кнопка (Button) та відображення для вводу тексту (EditText) є субкласами android.View.View.TextView, в свою чергу від класу Button наслідуються RadioButton(список з можливістю відмітити лише один з пунктів) та CheckBox (список з можливістю відмітити декілька пунктів). При цьому ImageButton наслідуються від ImageView. Детальна ієрархія відображень наведена на Рис. 2.4.

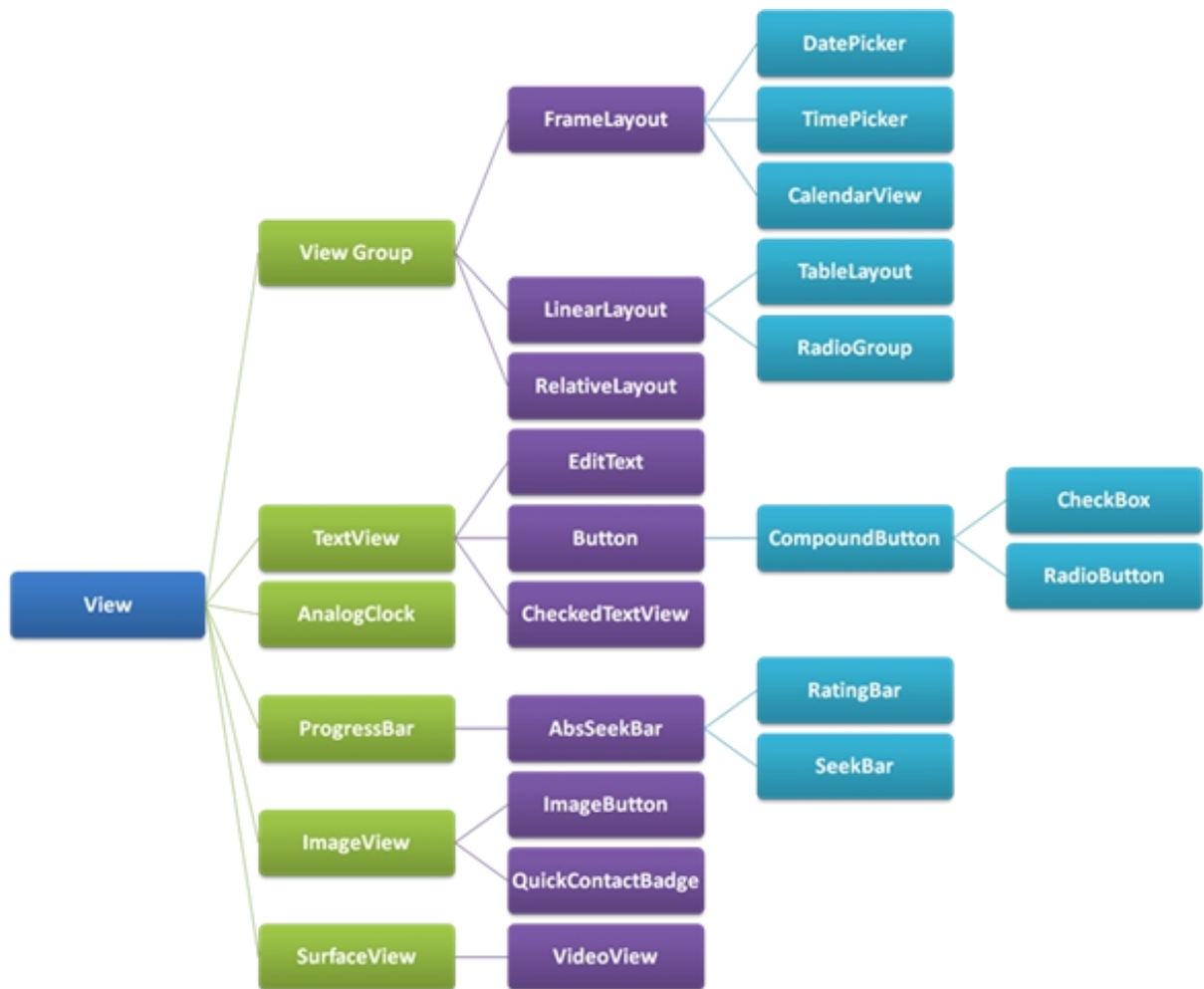


Рис. 2.4. Ієрархія візуальних відображень в ОС Андроїд.

2.1.8. SQLite

SQLite - це бібліотека на мові C, яка реалізує невеликий , швидкий , автономний , високонадійний , повнофункціональний механізм баз даних SQL[6]. Ця база даних має багато переваг, серед яких:

- Нульова конфігурація - базу даних не потрібно встановлювати, налаштовувати, призначати дозволи користувачам, або створювати новий екземпляр.
- Код навіть повного налаштування займає до 600 КБ, а без додаткових функцій ще менше
- Простий API
- Висока швидкість роботи

- Має відкритий сирцевий код, який є детально задокументований
- Існує єдиний файл вихідного коду, який легко додати в новий проєкт
- Відсутні зовнішні залежності
- Кроссплатформеність

В Android існує допоміжний клас для управління створенням бази даних та управлінням версіями - SQLiteOpenHelper. Для його створення потрібно перевизначити наступні методи: onCreate(SQLiteDatabase), onUpgrade(SQLiteDatabase, int, int) та , за потреби onOpen (SQLiteDatabase).Цей клас відповідальний за відкриття бази даних, її створення якщо цього не було зроблено раніше, наприклад під час першого запуску застосунку та її оновлення.

SQLite розуміє більшу частину стандартної мови SQL, але деякі функції опущені.


2.2. Огляд архітектурних паттернів що використовуються у мобільних додатках

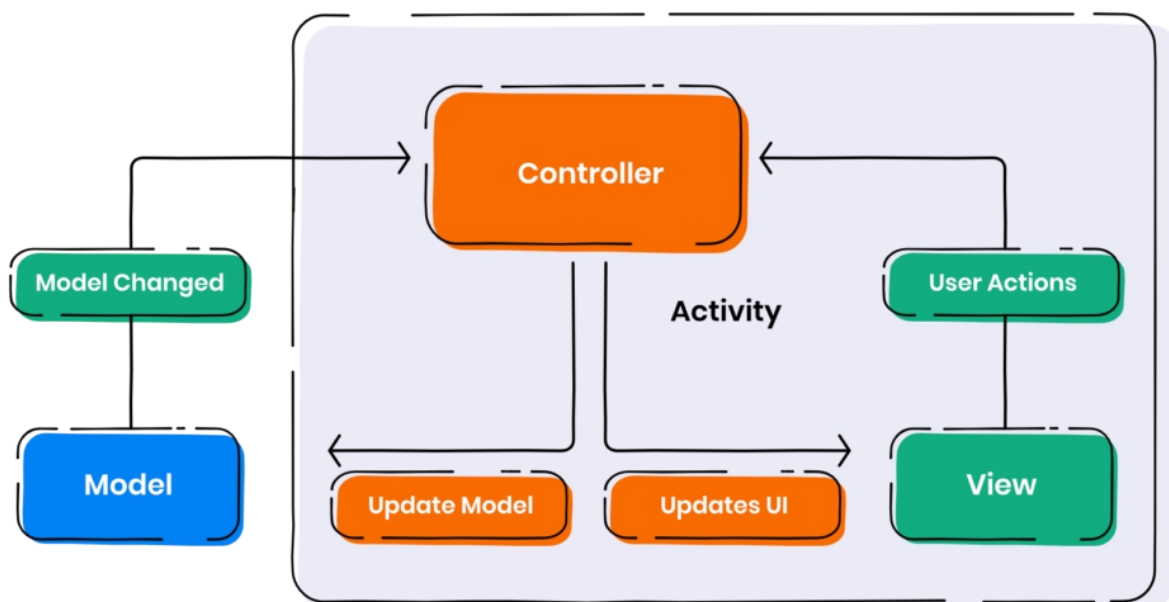
Архітектура програмного забезпечення(англ. softwarearchitecture) – це структура програми або обчислювальної системи, яка включає програмні компоненти, видимі зовні властивості цих компонентів, а також відносини між ними.[4]

2.2.1. Model-View-Controller (MVC)

Один з наstarіших та найпоширеніших паттернів, його концепція вперше була описана 1978 року Трюгве Реенскаугом. Його суть полягає в суворому розділенні між моделлю (model), видом (view) та контролером (controller). Модель містить данні та методи роботи з ними. Вид відповідає за відображення отриманих

даних. Контролер обробляє дії користувачів та змінює модель відповідним чином. В активній моделі при зміні моделі вид отримує повідомлення та змінюється, відобразивши нову інформацію. В пасивній вид оновлюється через контроллер. Також контроллер може управляти декількома активностями та змінювати їх між собою. Однак в “Android” активіті є і видом і контролером, що порушує принцип єдиної відповідальності, на якому базується даний паттерн. Діаграма класів даного шаблону наведена на рис 2.5

 | Diagram 1 - MVC (Model - View - Controller)



Scalable Path 2020

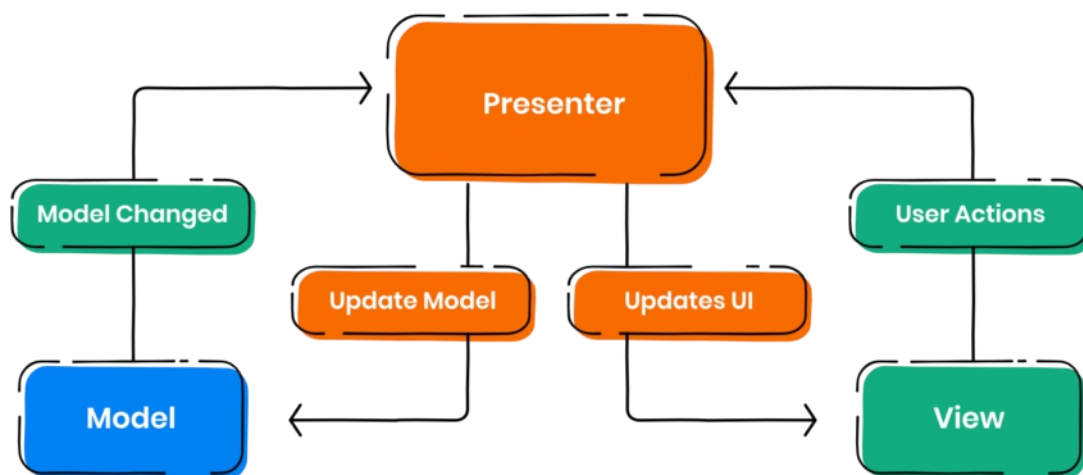
Рис. 2.5. Діаграма класів MVC

2.2.2. Model-View-Presenter (MVP)

Даний паттерн є похідним від MVC. Модель так само містить в собі логіку роботи з даними та самі дані. Його відмінність полягає в тому що пред'явник (Presenter) на відміну від контролера містить логіку реагування на події, оновлює модель та змінює вид, а сама

модель не має взаємодіє з відображеннями. Діаграма класів даного шаблону наведена на рис. 2.6

Diagram 2 – MVP (Model – View – Presenter)

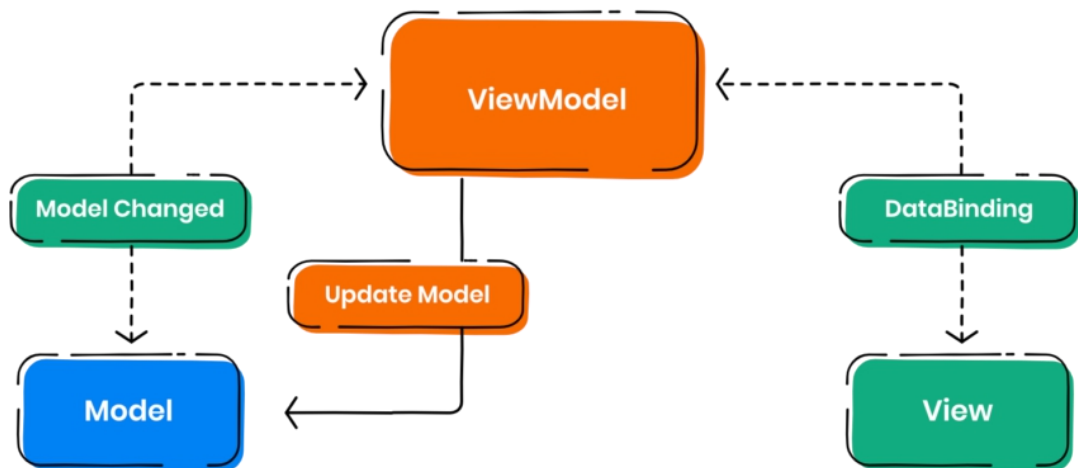


Scalable Path 2020

Рис. 2.6. Діаграма класів MVP

2.2.3. Model-View-ViewModel (MVVM)

В моделі (Model) описана логіка роботи з даними. Відображення (View) являє собою графічний інтерфейс. Модель виду (ViewModel) пов'язує модель та вид через механізми прив'язки даних, тобто якщо в моделі змінюються дані то це призводить до автоматичної зміни відображення. Для реалізації даного паттерну в android SDK існує бібліотека прив'язки даних, яка автоматично поєднує дані з виду моделі у вид. Діаграма класів даного шаблону наведена на рис. 2.7

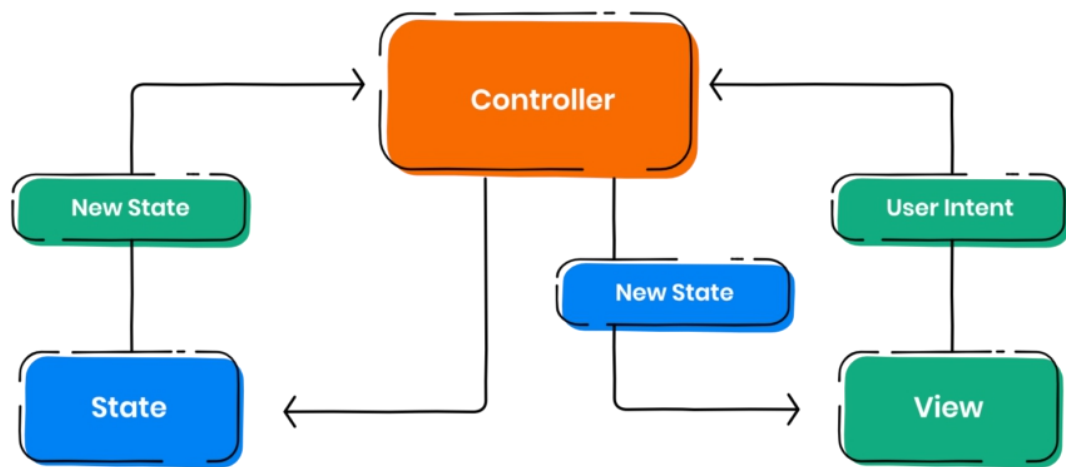


Scalable Path 2020

Рис. 2.7. Діаграма класів MVVM

2.2.4. Model-View-Intent (MVI)

Функція наміру (Intent) бере вхідні дані від користувача (тобто події користувацького інтерфейсу, такі як події кліків) і переводить їх у “щось”, яке буде передано як параметр функції `model ()`. Конструктор моделі (Model) бере вихідні дані з наміру як вхідні дані для маніпулювання моделлю. Результатом роботи цієї функції є нова модель (зі зміненим станом. Таким чином не потрібно оновлювати вже існуючу модель. Метод `view` бере модель, повернуту з конструктора моделі, і надає її як вхід до функції `view ()`. Таким чином відображення оновлюється. Діаграма класів даного шаблону наведена на рис. 2.8.



Scalable Path 2020

Рис. 2.8. Діаграма класів MVI

2.3. Вибір технологій для розробки

2.3.1. Вибір мови програмування для розробки клієнтської частини

Є доволі багато мов програмування на яких можлива розробка для ОС Андроїд, але офіційними для розробки є лише Java та Kotlin, тому вибір стоїть лише між ними.

Java – об’єктно-орієнтована мова програмування з явною, суворою, статичною типізацією та C-подібним синтаксисом. Початковий код зберігається в файлах з розширенням .java. Компілятор (javac) приймає початковий код і компілює його в файли з розширенням .class, котрий виконується на віртуальній машині Java (JVM) (Рис 2.9). Але у випадку запуску програми на операційній системі Android замість JVM використовується віртуальна машина Dalvik, що було описано вище.

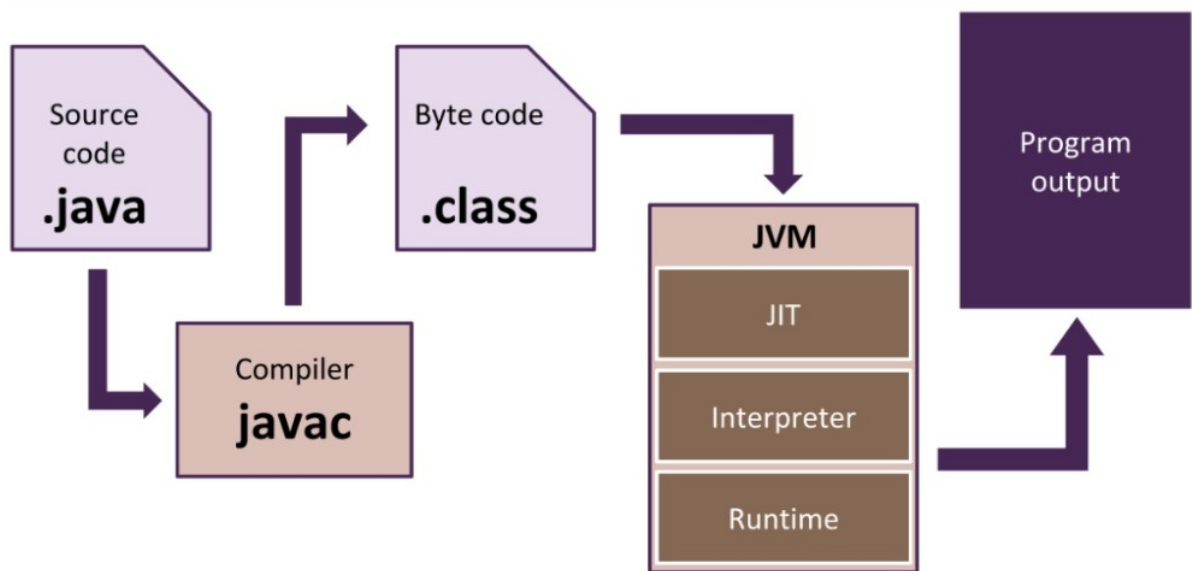


Рис 2.9 Процес компіляції сирцевого коду Java

Kotlin – об’єктно-орієнтована мова програмування з суворою типізацією, що працює поверх JVM та є сумісним з Java. Початковий код зберігається в файлах з розширенням .kt. Компілятор Kotlinc приймає на вхід початкові файли написані на Kotlin та на Java, це необхідно для сумісності двох мов, та дає на виході файли .class. І вже вони виконуються на JVM (Рис 2.10) або у випадку з Android – DVM.

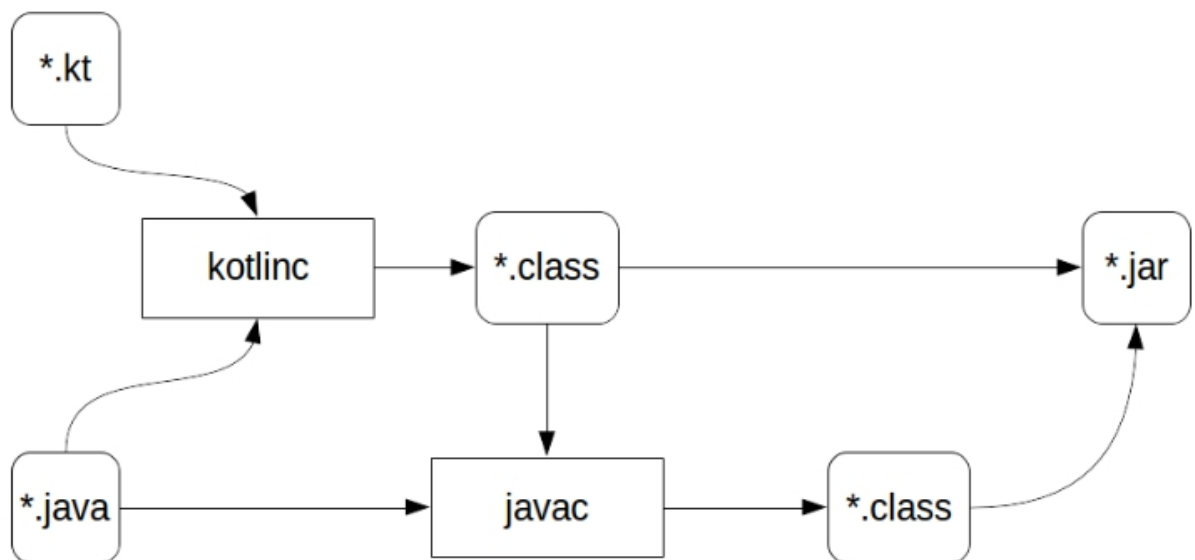


Рис. 2.10 Компіляція сирцевого коду Kotlin

Як бачимо обидві мови схожі між собою та відрізняються здебільшого різним синтаксисом. Kotlin має простіший та більш сучасний синтаксис, що значно пришвидшує написання коду та збільшує його читабельність, але синтаксис Java суворіший тому зменшується вірогідність помилок під час створення програмного забезпечення. При цьому для вивчення та використання Kotlin необхідні знання в області розробки мовою програмування Java.

Враховуючи вище перелічені фактори було прийнято рішення для подальшої розробки використовувати мову Kotlin

2.3.2. Вибір технології для гри локальною мережею

На сьогоднішній день існує безліч бездротових технологій, але найпопулярнішими є Bluetooth (Блютуз) WI-FI та (Вай-Фай).

Bluetooth – це стандарт бездротової технології короткого радіусу дії та використовується для обміну даними між фіксованими та мобільними пристроями або аксесуарами до них, працює на частоті від 2,402 ГГц до 2,48 ГГц. IEEE стандартизував дану технологію як IEEE 802.15.1, але більше не підтримує стандарт. На сьогоднішній день Bluetooth управляється Bluetooth Special Interest Group до якої входять різні компанії в галузях обчислювальної техніки, телекомунікацій, мереж, побутової техніки. Платформа Android включає підтримку мережевого стека Bluetooth, завдячуючи цьому пристрої з підтримкою Bluetooth можуть обмінюватися даними бездротово між собою. Програмний фреймворк забезпечує доступ до функціональних можливостей Bluetooth з використанням Bluetooth API. Ці API дозволяють програмам підключатися до інших пристроїв Bluetooth, дозволяючи бездротові функції point-to-point і multipoint.

					ІАЛЦ.467200.003 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

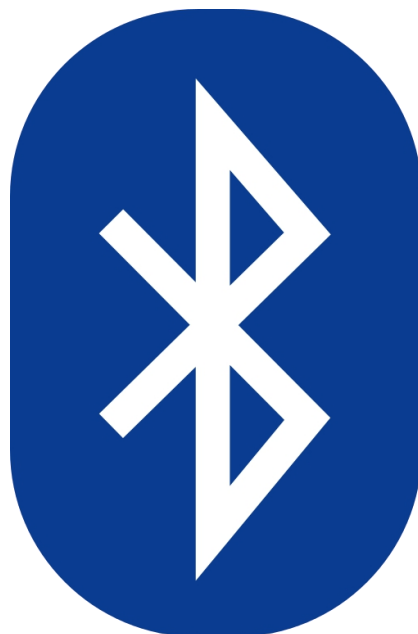


Рис 2.11 Логотип Bluetooth

Використовуючи API Bluetooth, програма може виконувати наступні дії:

- Проводити пошук інших пристроїв Bluetooth.
- Запитувати локальний адаптер Bluetooth щодо спарених пристроїв.
- Під'єднуйтесь до інших пристроїв за допомогою пошуку служб.
- Передавати дані на інші пристрої та з них.
- Керувати кількома з'єднаннями.

Wi-Fi – загальноприйнята назва стандартів передачі даних IEEE 802.11. для комунікації в бездротовій локальній мережі; а також торгова марка. Wi-Fi Direct – набір програмних технологій завдячуючи яким можна встановити з'єднання між двома або більше пристроїв з підтримкою Wi-Fi напряму, не потребуючи підключення до сторонньої мережі або точки доступу. В загальному Bluetooth та Wi-Fi Direct схожі між собою, однак Wi-Fi Direct має більший радіус дії, вищу швидкість передачі даних, але є більш енергоємним.

Було прийнято рішення використовувати технологію Wi-Fi Direct для створення локальної мережі, оскільки її вище описані позитивні сторони переважають над недоліками.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

ВИСНОВКИ ДО РОЗДІЛУ 2

У даному розділі були оглянуті принципи побудови застосунків для Android та архітектурні паттерни програмного забезпечення, що використовуються для створення мобільних додатків. Були проаналізовані та вибрані технології, які необхідні для створення додатку.

					ІАЛЦ.467200.003 ПЗ	Арк.
						33
Зм.	Лист	№ докум.	Підпис	Дата		

РОЗДІЛ 3. РОЗРОБКА ІГРОВОГО ДОДАТКУ “ШАХИ” ДЛЯ ANDROID

3.1. Засоби розробки

Під час розробки ігрового додатку “Шахи” для Android було використано Android studio.

Android studio – середовище розробки програмного забезпечення для операційної системи Android, створене на ядрі IntelliJ IDEA. Перевагами над іншими засобами розробки є графічний редактор макетів, відносно швидкий емулятор пристроїв з даною операційною системою. З недоліків можна виділити високу ресурсозатратність.

Android SDK - засіб розробки програмного забезпечення для операційної системи Android, дозволяє запускати, тестувати та відлагоджувати код в режимі сумісності з різними пристроями.

3.2. Загальний опис структури Android проєкту

Сирцевий код додатків Android з чіткою структурою директорій, що полегшує читання коду, отже і його підтримку та подальше нарощення функціональності. В застосунках для даної операційної системи структура модулів строго регламентована.

Модуль являє собою набір вихідних файлів і параметрів збірки, які дозволяють розділити проєкт на дискретні одиниці функціональності. Проєкт може мати один або декілька модулів, вони можуть залежати між один від одного.

Android проєкт за замовчуванням містить в собі наступні директорії:

- Manifests (маніфести) – містить в собі файл-маніфест додатку (AndroidManifest.xml)
- Java – ця директорія містить в собі всі файли сирцевого коду

- res – тут містяться всі не кодові джерела, які розподілені між вкладеними папками.
 - res/drawable – містить всі малюнки, що використовуються в додатку, вони можуть бути різних типів
 - res/layout – містить в собі всі макети, описані мовою XML, що використовуються для опису користувацького інтерфейсу.
 - res/mipmap – тут містяться файли для відображення іконки на домашньому екрані.
 - res/values – директорія що містить XML файли, в яких зберігаються рядки котрі використовуються в додатку, описи кольорів, розмірів, стилей, тем.
- Gradle Scripts – зберігає в собі сценарії для автоматизованої системи збірки “Gradle”.

3.2.1. Файл маніфесту (AndroidManifest.xml)

Цей файл є обов’язковим в кожному проєкті, та описує інформацію про додаток, інструменти збірки Android, операційну систему та Google Play (магазин мобільних застосунків для операційної системи Android).

В цьому файлі обов’язково мають бути задекларовані:

- Ім’я пакета програми що зазвичай відповідає простору імен коду розробника, інструменти збірки використовують це для визначення розташування об’єктів коду, Збірник проєкту замінює це значення ідентифікатором з файлів Gradle, в подальшому це значення використовується як унікальний ідентифікатор додатку в ОС то в Google Play.
- Компоненти застосунку серед яких: всі активності, сервіси, послуги, приймачі запитів та провайдери контенту. Кожен компонент обов’язково має визначати ім’я свого класу, також можливо додатково описати конфігурацію пристрою, з яким він

може працювати та фільтри намірів (Intent), що описують коли компонент запускається.

- Дозволи, необхідні програмі для доступу до захищених частин системи або інших додатків. Тут також декларуються які необхідно мати іншим застосункам для доступу до вмісту даного додатку.
- Апаратні та програмні функції, які вимагає програма, від чого залежить які пристрої матимуть можливість встановити додаток з Google Play.

3.2.2. Директорія ресурсів Android (res)

Ресурси – додаткові файли та статичний вміст які використовує програмний код. Це можуть бути растрові зображення, текстові рядки, коди кольорів і так далі. Варто відділяти ресурси від власне коду, оскільки так значно спрощується їх підтримка. Операційна система Android дає можливість в одному додатку надавати ресурси для різних конфігурацій пристроїв наприклад: різні макети для смартфонів, планшетів, годинників і т.д; різні файли ресурсів рядків для пристроїв з різними мовами; малюнки з різною роздільною здатністю для відповідних екранів і так далі. Для кожної конфігурації існує своя папка.

В першу чергу ресурси сортуються за типами, для кожного з яких підтримуються різні підкаталоги, які детальніше описані в таблиці 3.1

Таблиця 3.1

animator/	XML-файли, які описують анімацію властивостей
anim/	XML-файли, які описують анімацію руху

Продовження таблиці 3.1

color/	XML-файли що визначають список кольорів
drawable/	Растрові файли різних типів (.png, .9.png, .jpg, .gif)
ipmap/	Файли іконок для екранів р різних розширенням
layout/	XML-файли що визначають користувацький інтерфейс
menu/	XML-файли що визначають меню додатку
raw/	Довільні файли для збереження в необробленому вигляді
values/	XML-файли що містять в собі прості значення такі як рядки, цілі числа, кольори і т.д.
xml/	Довільні XML-файли які можна прочитати викликавши під час виконання коду
font/	XML-файли що визначають шрифти

3.2.3. Система збірки проєктів Gradle

Gradle – це інструмент автоматизації збірки з відкритим сирцевим кодом, який розроблений для того, щоб бути досить гнучким для створення майже будь-якого типу програмного забезпечення.[5]

Gradle виконує лише ті завдання для яких змінились вхідні та вихідні дані. Можливе використання кешу попередніх збірок, що дозволить повторне використання результатів виконання попередніх завдань, або навіть запусків з іншої машини, але лише зі спільним кешем. Таким чином Gradle є одним з найпродуктивніших збірників проєктів.

Gradle працює на JVM (Java virtual machine – віртуальна машина Java), що полегшує його запуск на різних платформах, також завдяки цьому в логіці збірки можливо використовувати стандартні API Java (API Application Programming Interface – програмний інтерфейс додатку).

Процес побудови типового модуля програми для Android, виконує наступні кроки(рис. 3.1):

1. Компілятори перетворюють сирцевий код у файли DEX (Dalvik Executable), які включають байт-код, що працює на пристроях Android, а все інше – у скомпільовані ресурси.
2. Пакувальник APK поєднує файли DEX та зібрані ресурси в єдиний файл .apk. Однак перед тим, як додаток можна встановити та розгорнути на пристрої Android, APK потрібно підписати.
3. Пакувальник APK підписує ваш APK, використовуючи ключі (keystore) для випускної версії(реліз) або налагоджувальної (дебаг):
 1. Якщо ви створюєте налагоджувальну версію свого додатка, тобто додаток, який ви призначені лише для тестування та профілювання, пакувальник підписує ваш додаток ключами налагодження. Android Studio автоматично налаштовує нові проекти за допомогою ключів налагодження.
 2. Якщо ви створюєте випускну версію свого додатка, яку ви збираєтеся випустити зовні, пакувальник підписує вашу програму сховищем ключів випуску.
4. Перед тим, як створити ваш остаточний файл .apk, пакувальник використовує інструмент zipalign для оптимізації вашого додатка для використання меншого об'єму пам'яті під час роботи на пристрої.

В кінці процесу збірки у вас є APK-файл для налагодження або APK-файл для випуску, який ви можете використовувати для розгортання, тестування або випуску для зовнішніх користувачів.

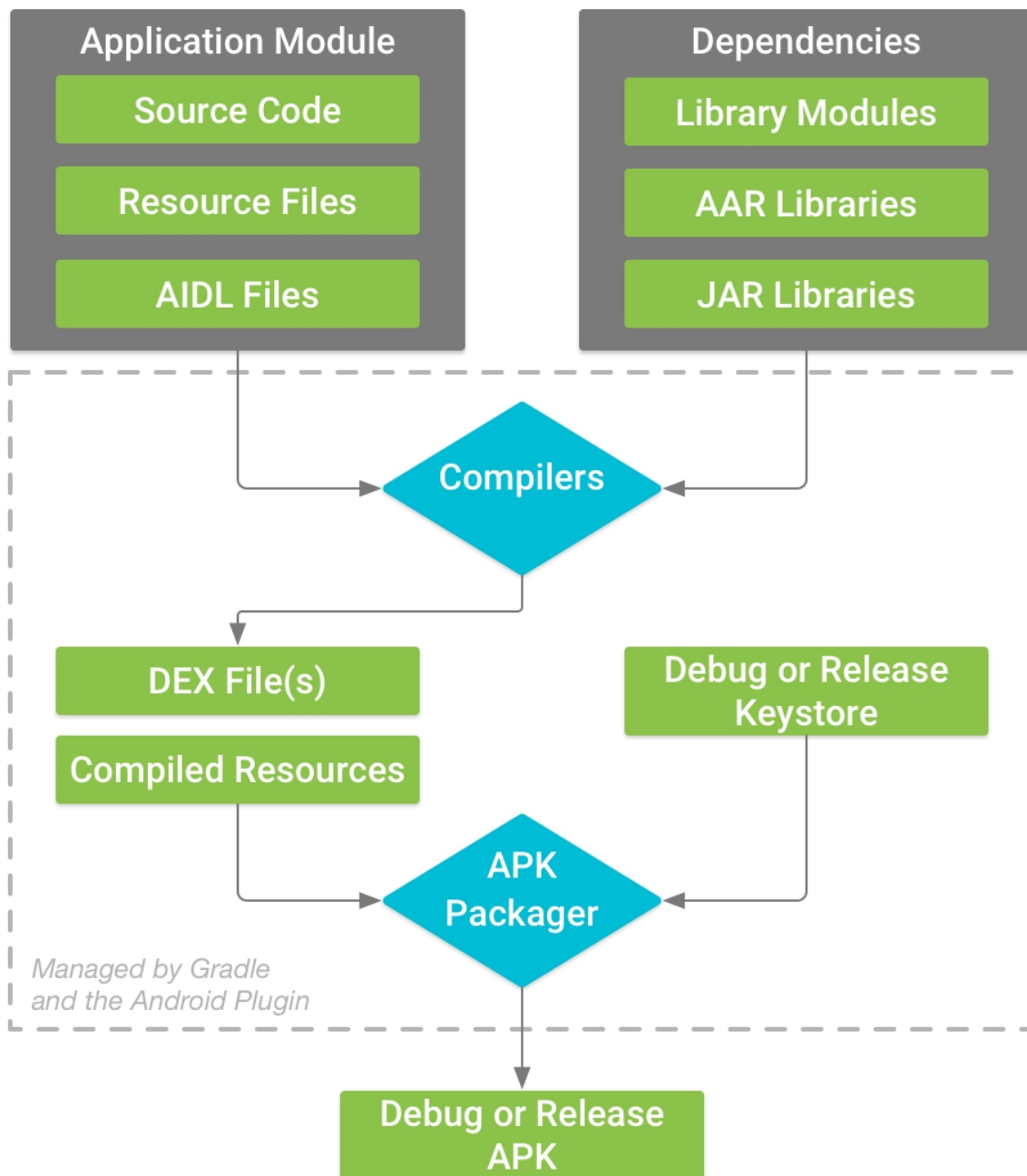


Рис 3.1 Процес побудови типового модуля програми для Android збірником Gradle.

Для спрощення розробки та подальшого розвитку проєкту файли сирцевого коду були розділені між двома пакетами, а саме UI

та Game_logic. Структура інших, некодових файлів не відрізняється від стандартної структури проєкту Android, яка була детально описана в другому розділі. Структура проєкту зображена на рисунку 3.2. Пакет UI містить в собі код в якому описується відображення активностей та фрагментів. В пакеті Game_logic міститься ігрова логіка, а саме перевірка чи за правилами гравець намагається здійснити хід. Оскільки в проєкті використовується лише одна база даних, клас в якому описаний її конструктор (DataBaseHelper) міститься в базовому пакеті сирцевого коду “com.example.chess”.

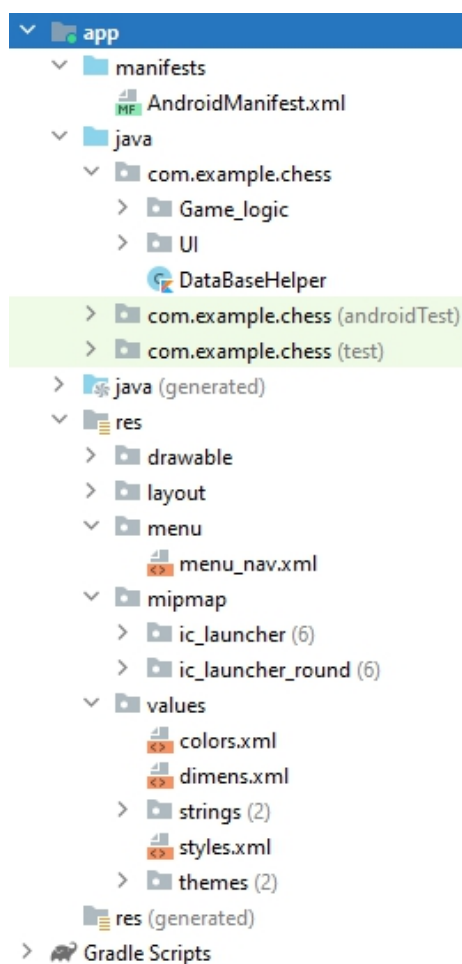


Рис. 3.2 - Структура файлів проєкту

3.3. Опис додатку

3.3.1. Користувацький інтерфейс

Додаток містить в собі одну активність та чотири фрагменти, кожен з яких відповідає за свою частину інтерфейсу, перехід між фрагментами здійснюється за допомогою меню(Рис. 3.2).

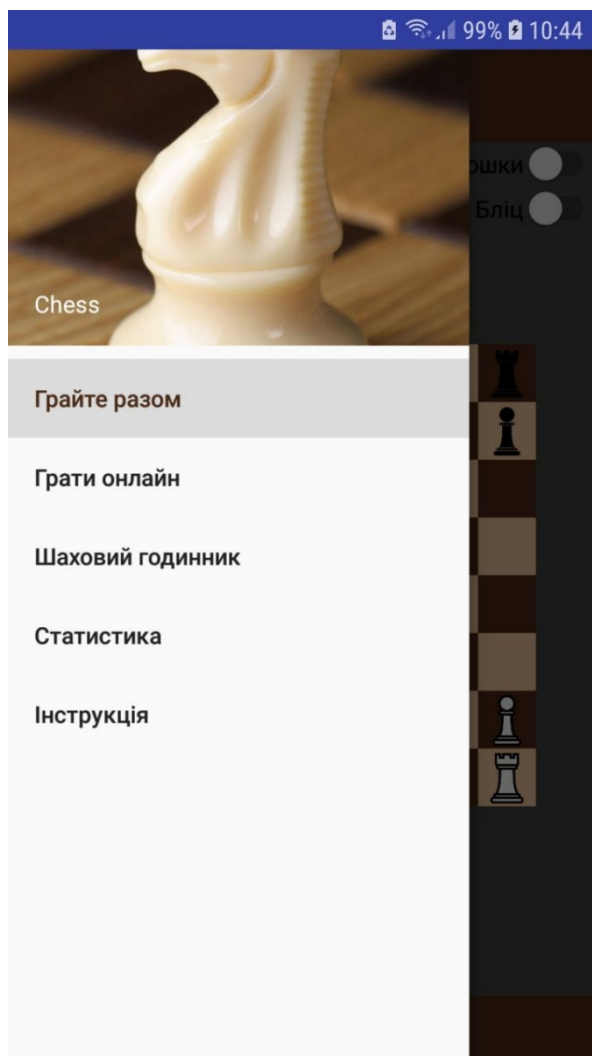


Рис. 3.3 Приклад відображення меню

Фрагмент гри вдвох (Рис. 3.3) містить в собі два перемикачі (Switch), власне ігрову дошку, поля для вводу параметрів гри в бліц, та кнопку “Розпочати гру”, та в разі гри в бліц відображення залишеного часу кожного з гравців. Один з цих перемикачів вмикає та вимикає можливість довільного ходу (незалежного від правил гри)

та можливість брати додаткові фігури обом користувачам. Це дозволяє використовувати фору наприклад брати додаткові фігури або забирати, також цю можливість можна використати для відміни ходу. Інший перемикач відповідальний за вибір режиму (бліц або класична гра). За замовчуванням він вимкнений, після його вмикання поля для вводу часу стають видимими та користувач може задати параметри гри. Після початку гри цей перемикач стає неклікабельним а поля для вводу невидимими.



Рис. 3.4. Приклад гри в бліц вдвох на одному пристрої з використанням фори

В логіці даного фрагменту відбувається зчитування координат поля на яке натиснув користувач: якщо жодне з полів не виділено - виділяємо його, якщо виділено - перевіряємо чи не збігається дане

поле з натисненим, якщо ні - передаємо початкові та кінцеві координати в метод класу Board, в якому відбувається перевірка чи за правилами здійснено хід, якщо так - переміщуємо фігуру.

Гра локальною мережею реалізована лише за класичними правилами, тобто без редагування дошки та без обмежень часу, відповідно зникає потреба в перемикачах режимів та вводу часу. Але з'являється нотація - запис вже зроблених ходів. Приклад гри локальною мережею зображений на рисунку 3.3



Рис. 3.5 Гра локальною мережею

В цьому фрагменті переміщення фігур відбувається так само як і в попередньому, за винятком лише того що після перевірки початкові та кінцеві координати передаються в клас

BroadcastReceiver, який відповідає за з'єднання між двома пристроями мережею WIFI Direct, та буде описаний нижче .

Для гри в бліц з використанням матеріальних шахів в додатку присутній годинник. Фрагмент годинника містить в собі поля для налаштування параметрів гри, кнопку натискання на яку означатиме початок відліку або перехід ходу до суперника отже і початок або продовження відліку часу другого гравця та призупинку відліку першого. Даний фрагмент зображено на рисунку 3.4.



Рис. 3.6 Режим шахового годинника

Для кращого розуміння інтерфейсу користувачем в додатку була детально розписана інструкція.

Також в додатку присутня статистика. Як видно з рис 3.6 в цьому фрагменті відображається список нотацій зіграних партій. Після вибору партії на екрані з'являються кнопки “наступний хід”

та ”попередній хід”, перемикач “альтернативна гра”, якщо він вимкнений можливе лише відображення нотації на шахівниці. Якщо увімкнений можливий довільний хід, або декілька ходів.

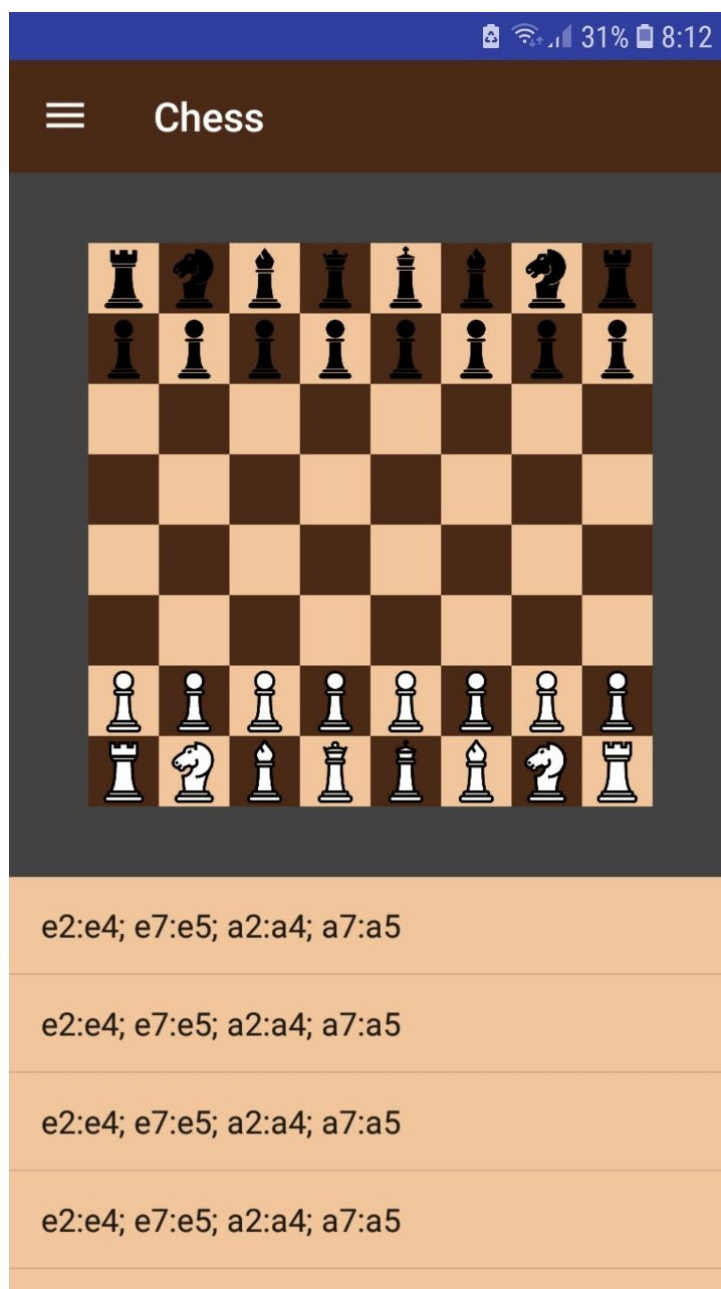


Рис. 3.7 Приклад фрагменту статистики

За замовчуванням в застосунку встановлена українська мова, але також присутня англійська. Додаток буде англomовним якщо встановлена мова пристрою – англійська. У всіх інших випадках – українська. Інтернаціоналізація реалізовувалась стандартними засобами Android: для кожної мови міститься свій файл набору

рядків доступ до яких можемо мати з інших частин проєкту за іменем рядка, файл з потрібною мовою операційна система обирає самостійно в залежності від мови пристрою.

3.3.2. Ігрова логіка

Перш ніж описувати реалізацію ігрової логіки необхідно описати правила гри в шахи. Гравці здійснюють хід по черзі, леше одною фігурою за хід, за винятком рокіровки. Починає гру біла сторона. Якщо фігура переміщується на поле зайняте фігурою суперника - фігура суперника знімається з дошки (хід має назву “взяття”). Кожна фігура пересувається за власними правилами:

- Пішак - пересувається лише по вертикалі, лише вперед, може взяти будь-яку фігуру противника лише по діагоналі. Перший хід може здійснити на дві клітинки, решту ходів лише на одну. Якщо пішак перемістився на два поля вперед та став поруч на одній горизонталі з пішаком суперника - лише наступного ходу суперник може забрати пішака походивши по діагоналі (так званий особливий хід “взяття на проході”) Приклад наведено на рисунку 3.6.
- Тура - ходить лише по горизонталі та вертикалі
- Кінь - пересувається за один хід на одну клітинку по горизонталі чи вертикалі потім по діагоналі в будь-якому напрямку. Єдина фігура здатна перестрибувати інші фігури.
- Слон - ходить по діагоналях.
- Король - пересувається в будь-якому напрямку на одну клітинку яка не знаходиться під боєм фігури суперника, за винятком рокіровки. Рокіровка - особливий хід короля що виконується наступним чином: король пересувається через одне поле в напрямку до тури, а тура ставиться на це поле. Рокіровка виконується лише якщо король та тура не робили до цього хід,

між ними не стоять інші фігури, король не знаходиться під шахом та не пересувається через біте поле.

- Ферзь пересувається по горизонталі, вертикалі або діагоналі.

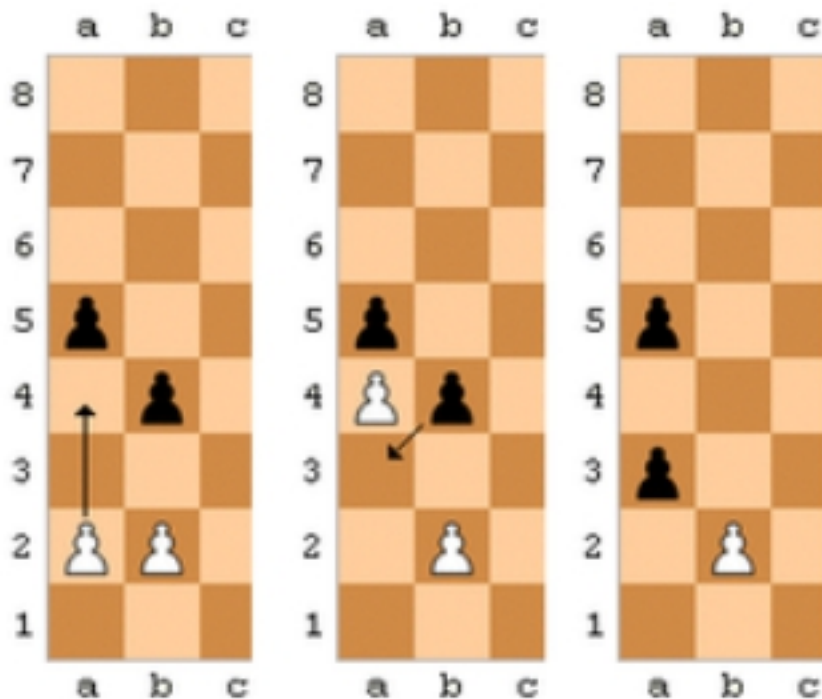


Рис. 3.8 Приклад взяття на проході

З користувацького інтерфейсу створюється екземпляр класу Board.

В класі Board з пакету Game_logic міститься список списків boardPiece, кожна комірка якого містить в собі об'єкти класів що реалізують інтерфейс Piece (шахова фігура) або null. Для кожної фігури реалізований свій клас, що реалізує інтерфейс Piece, це необхідно тому що кожна фігура має свої особливості здійснення ходу. З користувацького інтерфейсу є доступ лише до класу Board. Цей клас містить методи moveRule та moveWithoutRule кожен з яких відповідає за рух фігури за правилами та без правил відповідно. Методи цього класу наведені в таблиці 3.1

Метод `moveWithoutRule` видаляє фігуру зі старої позиції та переносить її на нову в списку `boardPiece`, за умови якщо стара позиція не була пустою, а нова не зайнятою.

Метод `moveRule` викликає метод `isMove` екземпляру фігури передаючи нові координати. Якщо він повертає `true` - відбувається переміщення фігури в списку `boardPiece`, та метод повертає `true` користувацький інтерфейс, де зміна позиції відображається.

Також клас містить наступні поля: `isCheck`, `isCheckmate`, `isStalemate`, `kingBlackPos`, `kingWhitePos` опис яких наведено в таблиці 3.2

Таблиця 3.2

Назва поля	Опис
<code>check</code>	Містить фігуру, яка оголосила шах, якщо шах не оголошено - приймає значення <code>null</code>
<code>isCheckmate</code>	Приймає значення <code>true</code> , якщо мат, інакше <code>false</code>
<code>isStalemate</code>	Приймає значення <code>true</code> , якщо пат, інакше <code>false</code>
<code>kingBlackPos</code>	Зберігає позицію чорного короля
<code>kingWhitePos</code>	Зберігає позицію білого короля
<code>passant</code>	Зберігає пішака (об'єкт <code>Pawn</code>), якого можна взяти на проході
<code>way</code>	Зберігає в собі список координат походивши на які фігура прикриє короля від шаху
<code>note</code>	В цю змінну записується нотація гри
<code>type</code>	Тип гри бліц або класичний
<code>time</code>	Якщо гра в бліц містить строкове представлення параметрів

Як вже було сказано класи в яких описані властивості фігур реалізують інтерфейс Piece, але окрім цього спадкують клас Board. Це необхідно для того щоб кожна фігура знала де знаходяться інші.

Фігура яка щойно здійснила хід перевіряє чи не знаходиться король суперника під її боєм (перевірка чи може фігура піти на певне поле, яке вказав користувач або на якому знаходиться король суперник буде наведена нижче), якщо так, то в змінну check записуємо цю фігуру, а в змінну way - всі поля на одне з яких потрібно піти щоб “закрити” короля від бою та свою позицію. Фігура яка ходить наступною не може піти на поле, координати якого не записані в way. Король при кожному своєму ході перевіряє чи не знаходиться поле на яке він збирається піти під боєм хоча б однієї з фігур суперника, якщо так - він туди піти не може, якщо йому був оголошений шах - змінній check присвоюється значення null (король вийшов з під нього). Блок схема описаного вище алгоритму наведена в додатку Б.

Зараз опишемо як відбувається перевірка чи за правилами здійснюється хід для інших фігури.

– Пішак: логіка ходу пішака реалізована в класі Pawn, який містить в собі змінну isFirst, яка набуває значення true якщо хід цього пішака перший, значить можна походити на 2 поля вперед. Якщо перший хід здійснено на два поля вперед - записуємо пішака в змінну passant класу Board. Якщо пішак доходить до кінця дошки - він замінюється іншою фігурою на вибір гравця.

– Тура: описано в класі Rock Якщо змінюється обидві координати - хід не за правилами, якщо тільки одна - перевіряємо чи не перетинає тура зайняту клітинку.

– Кінь - описаний в класі Knight. Модуль різниці по одній з координат має дорівнювати одиниці, по іншій двійці, потім перевіряємо чи не зайняте поле своєю фігурою.

– Слон - описаний в класі Bishop, модулі різниць початкових та кінцевих координат мають бути рівними (рух по діагоналі), також перевірка чи не перетинається зайняте поле під час руху

– Ферзь - описаний в класі Queen. Перевіряємо чи здійснюється рух по горизонталі чи по вертикалі, аналогічно турі, якщо ні - чи здійснюється рух по діагоналі аналогічно слону. Так само перевіряємо чи не перетинається зайняте поле.

Після здійснення ходу дописуємо його в note, по завершенню гри додаємо нотацію до бази даних.

3.3.3. База даних

Нотації всіх зіграних ігр записуються до бази даних. Вона містить в собі наступні поля: `_id`, `note`, `winner`, `type`, `time`. Опис кожного з полів наведено в таблиці 3.3

Таблиця 3.3

Назва поля	Тип даних	Опис
<code>_id</code>	INTEGER PRIMARY KEY AUTOINCREMENT	Ідентифікатор запису в базі даних, який є унікальним та генерується автоматично
<code>NOTE</code>	TEXT	Нотація гри
<code>WINNER</code>	TEXT	Колір сторони, що виграла
<code>TYPE</code>	TEXT	Гра мережею, чи на одному пристрої
<code>TIME</code>	TEXT	Параметри блицу, якщо класична гра поле залишається пустим

Для роботи з базою даних було створено клас DataBaseHelper, що містить в собі наступні методи:

- onCreate() створює базу даних в момент її першого виклику. Спочатку на пристрої створюється пуста база даних, після чого викликається цей метод з параметром SQLiteDatabase.
- onUpgrade() отримує три параметри - базу даних SQLite, номер версії бази даних користувача, та нову версію бази даних.
- updateMyDatabase() - приватний метод класу DataBaseHelper, який викликається з методів onCreate() та onUpgrade(), таким чином ми уникаємо повторення коду.
- insertGame() - приватний метод, який відповідає за вставку нового запис в базу даних.

3.3.4. Передача даних мережею Wi-Fi direct

Для використання WIFI direct (P2P) були додані в файл маніфесту дозволи на мережеві сокети, доступ до мережі WIFI, дозвіл на зміну стану з'єднання WIFI та доступ до точного розташування. Далі IntentFilter (фільтр намірів) трансляції був встановлений на прослуховування ввімкнення Wi-Fi P2P, зміну списку доступних пристроїв, зміну стану з'єднання та зміну конфігурації пристрою. Після чого викликаємо метод initialize() екземпляру WifiP2pManager, що повертає об'єкт WifiP2pManager.Channel, який використовується для підключення до Wi-Fi P2P. Далі створюємо новий клас BroadcastReceiver, який використовується для прослуховування мережі, в методі onReceive() цього класу прописуємо обробник кожної зміни станів перерахованих вище. А в методах onResume() та onPause() описуємо IntentFilter та broadcast receiver (приймач мовлення). Щоб розпочати пошук пристроїв

поблизу з Wi-Fi P2P, потрібно викликати метод `discoverPeers()`, передавши йому `WifiP2pManager.Channel` (був отриманий вище) та реалізацію методів, які система викликатиме при успішному або невдалому підключенні. Щоб отримати список однолітків потрібно реалізувати інтерфейс `WifiP2pManager.PeerListListener`, який надає інформацію про однорангові мережі. Щоб підключитись до пристрою створюємо об'єкт `WifiP2pConfig` і копіюємо в нього дані з `WifiP2pDevice` які хочем передати, та викликаємо метод `connect()`. Щоб встановити з'єднання з пристроєм що не підтримує Wi-Fi direct потрібно явно запитати пароль, викликавши метод `requestGroupInfo()`.

					ІАЛЦ.467200.003 ПЗ	Арк.
						52
Зм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 3

В даному розділі були оглянуті засоби розробки що використовувались для реалізації ігрового додатку “Шахи” для операційної системи Android, здійснена його розробка з використанням мови програмування Kotlin, мови розмітки XML та Android SDK. Також була описана архітектура додатку.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		53

ВИСНОВКИ

Результатом даної бакалаврської дипломної роботи є реалізація ігрового додатку “Шахи” для Android.

В ході виконання даної роботи були розглянуті аналоги додатків для гри в “Шахи”, але вони виявилися не досконалими та не реалізували деякі режими гри.

Були проаналізовані доступні та актуальні засоби для розробки системи. Було вирішено спроектувати додаток мовою програмування Kotlin.

В ході виконання дипломної роботи була досягнута поставлена мета та вирішена поставлена задача.

					ІАЛЦ.467200.003 ПЗ	Арк.
						54
Зм.	Лист	№ докум.	Підпис	Дата		

ДЖЕРЕЛА

1. Великий тлумачний словник сучасної української мови., 2005 – Київ: Перун, 205. – 1728 с.
2. Андроїд (ОС) - ВУЕ [Електронний ресурс] – Режим доступу до ресурсу: [https://vue.gov.ua/%D0%90%D0%BD%D0%B4%D1%80%D0%BE%D1%97%D0%B4_\(%D0%9E%D0%A1\)](https://vue.gov.ua/%D0%90%D0%BD%D0%B4%D1%80%D0%BE%D1%97%D0%B4_(%D0%9E%D0%A1)).
3. Wikipedia [Веб-сайт] - Режим доступу :<https://uk.wikipedia.org/wiki/IOS>
4. Поняття архітектури програмного забезпечення - Вікі ЦДПУ [Електронний ресурс] – Режим доступу до ресурсу: [https://wiki.cuspu.edu.ua/index.php/%D0%9F%D0%BE%D0%BD%D1%8F%D1%82%D1%82%D1%8F_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F#:~:text=%D0%90%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%20\(%D0%B0%D0%BD%D0%B3%D0%BB.,%D1%82%D0%B0%D0%BA%D0%BE%D0%B6%20%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F..](https://wiki.cuspu.edu.ua/index.php/%D0%9F%D0%BE%D0%BD%D1%8F%D1%82%D1%82%D1%8F_%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F#:~:text=%D0%90%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%20(%D0%B0%D0%BD%D0%B3%D0%BB.,%D1%82%D0%B0%D0%BA%D0%BE%D0%B6%20%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%20%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F..)

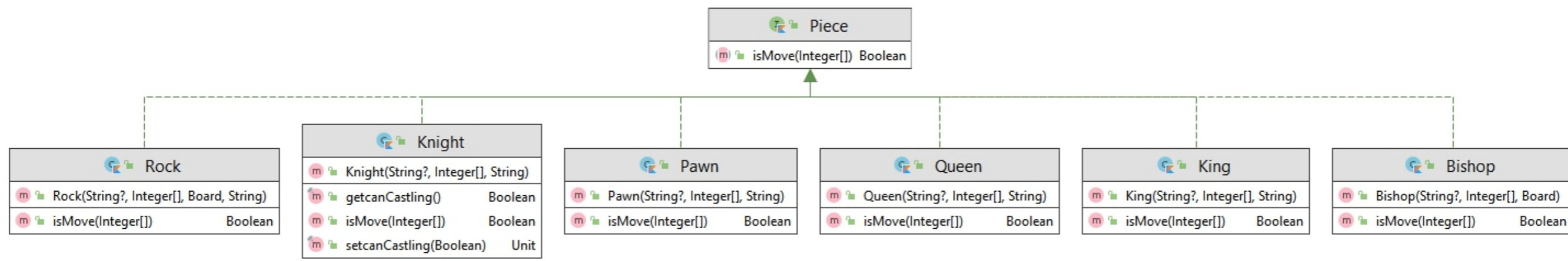
					ІАЛЦ.467200.003 ПЗ	Арк.
						55
Зм.	Лист	№ докум.	Підпис	Дата		

5. What is Gradle? [Електронний ресурс] – Режим доступу до ресурсу:
https://docs.gradle.org/current/userguide/what_is_gradle.html.
6. SQLite Home Page [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.sqlite.org/index.html>.
7. Services overview | Android Developers [Електронний ресурс] – Режим доступу до ресурсу:
<https://developer.android.com/guide/components/services?hl=ru>.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

Додаток А
ФУНКЦІОНАЛЬНА СХЕМА
до дипломного проєкту
на тему: «Ігровий додаток «Шахи» для Android»

Київ – 2021 року



DataBaseHelper

- DataBaseHelper(NonExistentClass)
- onCreate(NonExistentClass) Unit
- onUpgrade(NonExistentClass, Int, Int) Unit
- updateMyDatabase(NonExistentClass, Int, Int) Unit

WatchFragment

- WatchFragment()
- onCreate(NonExistentClass) Unit
- onCreateView(NonExistentClass, NonExistentClass, NonExistentClass) NonExistentClass
- onStart() Unit
- runTimer() Unit

PvpFragment

- PvpFragment()
- onCreate(NonExistentClass) Unit
- onCreateView(NonExistentClass, NonExistentClass, NonExistentClass) NonExistentClass
- onStart() Unit

Companion

- insertGame(NonExistentClass, String, String) Unit

StatisticFragment

- StatisticFragment()
- onCreate(NonExistentClass) Unit
- onCreateView(NonExistentClass, NonExistentClass, NonExistentClass) NonExistentClass
- onStart() Unit

OnlineFragment

- OnlineFragment()
- onCreate(NonExistentClass) Unit
- onCreateView(NonExistentClass, NonExistentClass, NonExistentClass) NonExistentClass
- onStart() Unit

MainActivity

- MainActivity()
- onBackPressed() Unit
- onCreate(NonExistentClass) Unit
- onNavigationItemSelected(NonExistentClass) Boolean

InstructionFragment

- InstructionFragment()
- onCreate(NonExistentClass) Unit
- onCreateView(NonExistentClass, NonExistentClass, NonExistentClass) NonExistentClass

Board

- Board()
- moveRule(Integer[][]) Boolean
- moveWithoutRule(Integer[][][]) Unit

WiFiDirectBroadcastReceiver

- WiFiDirectBroadcastReceiver(NonExistentClass, NonExistentClass, MainActivity)
- onReceive(NonExistentClass, NonExistentClass) Unit

HelpActivity

- HelpActivity()
- onCreate(NonExistentClass) Unit

Square

- Square()

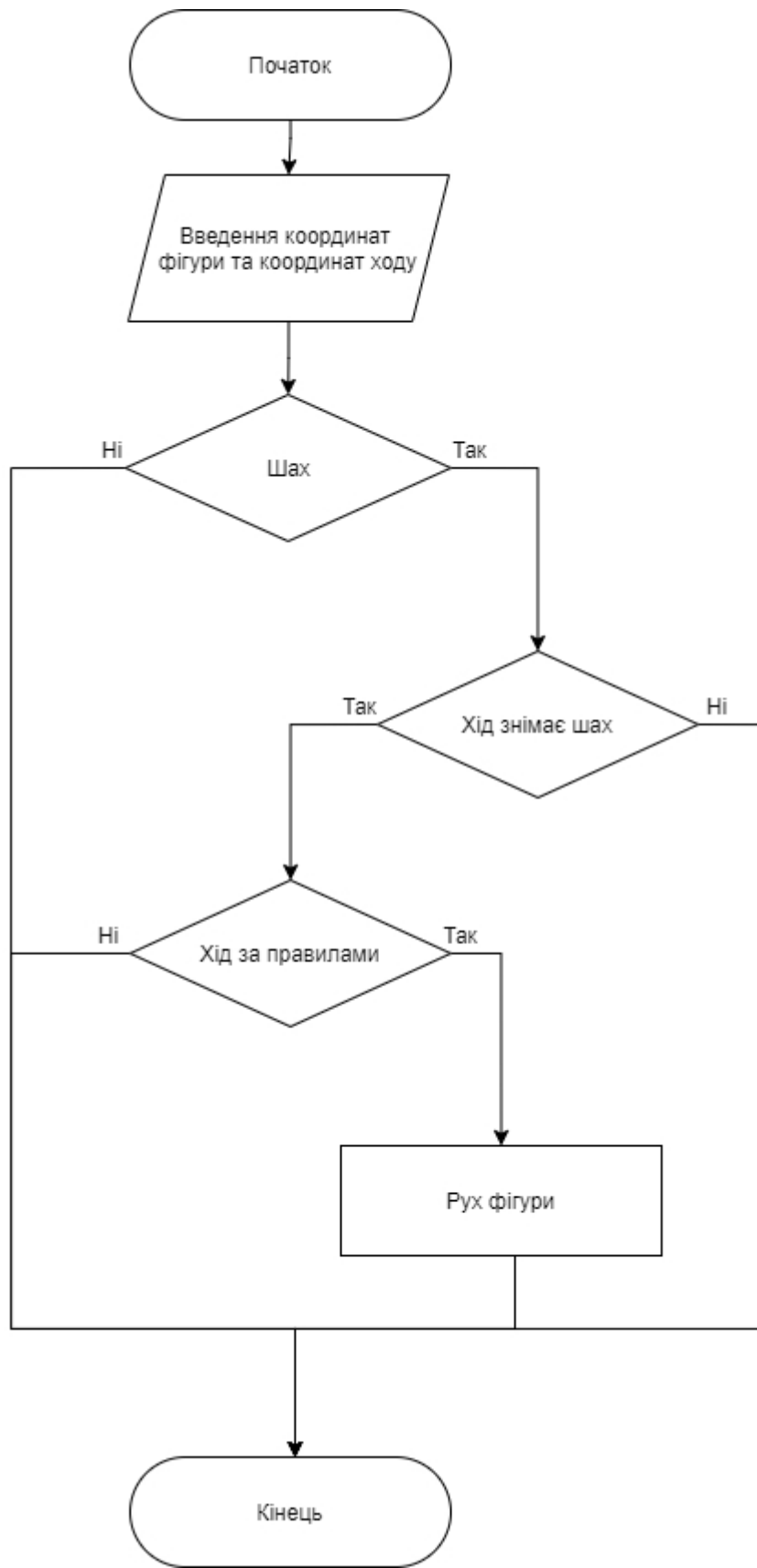
Powered by yf-ies

Змн.	Арк.	№ докум.	Підпис	Дат.
Розроб.		Кизима І.Л.		
Перевір.		Долголенко О.М.		
Т. Контр.				
Н. Контр.		Сімоненко В. П.		
Затверд.		Стіренко С. Г.		

ІАЛЦ.467200.004 Д1			
	Лім.	Маса	Масш.
Дипломна робота		Аркуш	Аркушів
НТУУ «КПІ», ФІОТ, ІО-71			

Додаток Б
ПРИНЦИПОВА СХЕМА
до дипломного проєкту
на тему: «Ігровий додаток «Шахи» для Android»

Київ – 2021 року



					<i>ІАЛЦ.467800.004 Д1</i>					
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Ігровий додаток "Шахи" для Android</i> <i>Принципова схема</i>					
<i>Розробив</i>	Кизима І.Л.							<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевірів</i>	Долголенко О.М.								1	1
<i>Реценз.</i>								<i>«КПІ імені Ігоря Сікорського»</i> <i>ФІОТ, гр. ІО-71</i>		
<i>Н. Контр.</i>	Сімоненко В.П.									
<i>Затв.</i>	Стіренко С.Г.									

Додаток В
СТРУКТУРНА СХЕМА
до дипломного проєкту
на тему: «Ігровий додаток «Шахи» для Android»

Київ – 2021 року

