

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Курсова робота
за спеціальністю 121 Програмна інженерія
на тему:
**РОЗРОБКА ANDROID-ДОДАТКУ
ДЛЯ ДИСТРИБУЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Виконав: студент 3-го курсу
Юрій ГРИЩЕНКО

(підпис)

Науковий керівник:
кандидат фізико-математичних наук, доцент
Олександр ГАЛКІН

(підпис)

Засвідчую, що в цій курсовій роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент

(підпис)

КИЇВ-2021

РЕФЕРАТ

Обсяг роботи 28 сторінок, 7 ілюстрацій, 12 джерел посилань.

ЦИФРОВА ДИСТРИБУЦІЯ, ДОСТАВКА КОНТЕНТУ, ДИСТРИБУЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ДОДАТОК ANDROID, АНАЛІЗ ВЕБ-ДОДАТКІВ

Об'єктом роботи є процес цифрової дистрибуції програмного забезпечення, зокрема комп'ютерних ігор, або доставка контенту. Предметом роботи є програмний засіб для завантаження та оновлення програмного забезпечення для пристроїв з операційною системою Android.

Метою роботи є створення програмного засобу для завантаження та оновлення програмного забезпечення для пристроїв операційною системою Android, зокрема комп'ютерних ігор.

Методи розроблення: розробка програмного продукту на основі еволюційної моделі та відгуків користувачів. Інструменти розроблення: безкоштовне, вільно поширюване інтегроване середовище розробки Android Studio, мова програмування Kotlin, що працює на JVM.

Результати роботи: виконано загальний огляд існуючих засобів дистрибуції програмного забезпечення для смартфонів, зокрема пристроїв з операційною системою Android, проаналізовано їх переваги та недоліки, розроблено програмний продукт Mitch - мобільний додаток, який дозволяє встановлювати та оновлювати програмне забезпечення з сайту itch.io.

Програмний продукт Mitch може застосовуватися в повсякденному житті в цілях розваги, як альтернатива Google Play.

ЗМІСТ

РЕФЕРАТ	1
ВСТУП.....	4
1 ІСНУЮЧІ ЗАСОБИ ОНЛАЙН-ДИСТРИБУЦІЇ	7
1.1 Поняття цифрової дистрибуції	7
1.2 Порівняння існуючих сервісів онлайн-дистрибуції	8
2 ДОСЛІДЖЕННЯ РОБОТИ СЕРВІСУ ITCH.IO	10
2.1 Початок дослідження.....	10
2.2 Приклад дослідження роботи веб-додатку itch.io.....	11
3 РОЗРОБКА ДОДАТКУ ANDROID	15
3.1 Засоби розроблення	15
3.2 Графічний інтерфейс користувача.....	16
3.3 Оновлення ігор.....	18
ВИСНОВКИ	19
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	21

ПРЕДМЕТНИЙ ПОКАЖЧИК

Кирилиця

Д

дистрибуція 6

Н

нативний 16

С

сервіс 9

Латиниця

С

Curl 10, 12

D

DRM 6

W

WebView 10, 14

ВСТУП

Оцінка сучасного стану об'єкта розробки. У порівнянні з іншими видами персональних комп'ютерів, смартфони є одними з найбільш популярних. Станом на 2021 рік, приблизно 6 мільярдів людей мають смартфон, планшет, ноутбук, або настільний персональний комп'ютер.[1] Серед цих категорій, смартфони мають найбільшу популярність. Саме ці чотири види комп'ютерів було обрано не випадково: вони мають різний форм-фактор, але з іншого боку володіють приблизно однаковою потужністю та можливістю виконувати програми, які забажає встановити їх власник.

Більшість користувачів персональних комп'ютерів не є програмістами, і не можуть власноруч писати програмне забезпечення. Тому ці пристрої зазвичай постачаються з певною операційною системою, що має можливість виконувати певні основні дії, а більш спеціалізовані програми встановлюються окремо, за потребою користувача.

Звідси робимо висновок, що можливість дистрибуції програмного забезпечення є незамінною властивістю персональних комп'ютерів. Постає питання: як це зробити швидким, ефективним та зручним у користуванні.

Актуальність роботи та підстави для її виконання. Більшість користувачів надають перевагу саме онлайн-дистрибуції програмного забезпечення, та і взагалі, будь-якого контенту.[2] Це має певні переваги над іншими методами, такими як CD, DVD чи дискети, а саме:

- швидкість та зручність доставки.
- доступ до більш різноманітного асортименту товарів, які з деяких причин складно розповсюджувати у місці провищення користувача.
- зниження ціни дистрибуції контенту як для видавця, так і для користувача.

Зокрема, смартфони призначені для зручного користування в режимі онлайн, а не для передавання даних на фізичних носіях. Тому актуальним є створення засобів саме онлайн-розповсюдження програмного забезпечення, а власне такого, який не можна отримати іншими засобами. Саме для цього і було обрано за основу платформу itch.io, яка містить певний контент, який необов'язково розповсюджується, наприклад, через сервіс Google Play.

Мета і завдання роботи. Метою курсової роботи є створення програмного засобу для завантаження та оновлення програмного забезпечення для пристроїв операційною системою Android, зокрема комп'ютерних ігор. Для досягнення цієї мети поставлено такі завдання:

- Дослідити існуючі засоби онлайн-дистрибуції програмного забезпечення, зокрема

для системи Android.

- Дослідити різні основи, або “бекенди”, що надають доступ до певних програм.
- Для обраного бекенду розробити власний засіб для завантаження та оновлення ігор.

Об’єкт, методи й засоби розроблення. Об’єктом роботи є власне процес онлайн-дистрибуції програмного забезпечення, зокрема ігор.

Розобці програмного засобу передував аналіз роботи обраного бекенду за допомогою певних інструментів, таких як Інструментів розробника в Firefox, та командної утиліти Curl.

В якості інструменту створення власне програмного засобу було обрано Android Studio — інтегроване середовище розробки (IDE) мовами програмування Java та Kotlin, яке є безкоштовним, вільно поширюваним, і рекомендованим власне розробниками операційної системи Android.[3]

Було обрано саме мову програмування Kotlin, оскільки вона компілюється у Java bytecode і може виконуватися на JVM, але при цьому надає додаткові можливості, що пришвидшують написання коду, роблять його коротшим та простішим у розумінні. Google, розробники Android та Android Studio, також рекомендують використовувати Kotlin замість Java для розробки нових додатків.[4]

Можливі сфери застосування. Кінцевий продукт може використовуватись за призначенням, як додаток для завантаження та оновлення програмного забезпечення, і для більш загальної взаємодії з порталом itch.io. Продукт надає доступ до додатків у сфері розваг, здебільшого мобільних та комп’ютерних ігор.

1 ІСНУЮЧІ ЗАСОБИ ОНЛАЙН-ДИСТРИБУЦІЇ

1.1 Поняття цифрової дистрибуції

За визначенням, цифрова дистрибуція (або “онлайн-дистрибуція” чи “доставка контенту”) — це метод поширення цифрового медіа контенту, такого як аудіо, фільми, відеоігри тощо, без використання фізичних носіїв (паперу, дисків, касет). Як правило, він полягає в завантаженні легального медіа на пристрій користувача через мережу Інтернет.[5]

Інколи цей термін може вживатись у більш широкому розумінні, як будь-яка доставка цифрових даних, як і через Інтернет, так і з використанням фізичних носіїв, але в цій роботі будемо термін “цифрова дистрибуція” саме синонімом “онлайн-дистрибуція”.

Зазвичай, засоби поширення програмного забезпечення мають певні технології керування цифровими правами (англ. Digital rights management, або DRM). Вони використовуються для обмеження дій, які може виконати користувач над певними даними, а саме — дозволити відтворювати медіа, при цьому не дозволяючи нелегальне розповсюдження медіа іншим користувачам. Деякі розповсюджувачі можуть обмежувати кількість пристроїв, на які можна встановити певні медіа (наприклад, обмежитися 5 пристроями, за умови, що покупець є власником усіх цих пристроїв; аби обійти це обмеження, доведеться купляти продукт знову). Проте, це не є обов’язковим для засобів онлайн-дистрибуції — існують сайти, які дозволяють завантажувати медіа у формі звичайних файлів (у форматі MP3, FLAC, APK, тощо), при цьому обмеження на їх розповсюдження є суто легальним, без імплементації технологій DRM.

Порівнюючи засоби онлайн-дистрибуції, будемо враховувати, чи наявне в них керування цифровими правами. В цілях цієї роботи за основу візьмемо дистрибутора, який не імплементує ці технології.

Засоби доставки зазвичай передбачають за собою певну плату за контент, або ж монетизацію. Можна класифікувати методи цифрової дистрибуції за типом монетизації[5]:

“Заплати та грай” (англ. Pay to play) — користувач платить за контент, потім його завантажує і відтворює чи запускає. Прикладом такого методу є платні ігри в сервісі Google Play або Steam.

“Спробуй, перш ніж купляти” (англ. Try before you buy) — надається можливість безкоштовно деякий час грати в гру, або подивитися один епізод серіалу, тощо. Щоб отримати повний доступ до медіа, необхідна оплата. Прикладом є ігри з демо-версіями в Steam.

Рекламна — контент є безкоштовним; видавець заробляє гроші з показів реклами, яку користувач змушений дивитися чи слухати під час відтворення медіа. Прикладом є більшість безкоштовних ігор в Google Play, або ж відео в YouTube.

Передплата — користувач періодично платить певну суму грошей, за що йому надається

можливість відтворювати будь-який контент з певного набору, наданого видавцем. До цього типу можна віднести, наприклад, сервіс YouTube Premium.

Бачимо, що один і той самий сервіс може підтримувати декілька режимів дистрибуції, наприклад, Microsoft Store пропонує як і платне програмне забезпечення, так і безкоштовне, що може містити рекламу.

Оплата відбувається за допомогою банківської карти, або платіжних систем Інтернету, такими як PayPal.

1.2 Порівняння існуючих сервісів онлайн-дистрибуції

В цілях нашої роботи необхідно порівняти існуючі засоби цифрової дистрибуції, зокрема дистрибуції програмного забезпечення. Це робиться з двох причин:

По-перше, варто проаналізувати існуючі рішення, щоб зрозуміти їх переваги та недоліки, та проаргументувати створення власного рішення.

По-друге, ми зможемо обрати один із цих сервісів за основу, і зробити навколо нього певний фронт-енд для зручнішого використання. Такий підхід найбільш доцільний, оскільки у нас немає ресурсів, щоб власне легально видавати певний контент.

Розглянемо найбільш вживані сервіси, які підтримують операційну систему Android:

Google Play — стандартний засіб дистрибуції програмного забезпечення для більшості пристроїв, що працюють на системі Android. Має приблизно 2 мільярди користувачів, і більш ніж 3 мільйони додатків.[6] Розробник зобов'язаний заплатити 25 доларів США, аби публікувати свої додатки.

Amazon Appstore — аналог Google Play для пристроїв Fire OS (системи від Amazon, оснований на Android). Розповсюджує приблизно 500,000 додатків, серед них майже всі також доступні на Google Play.[7] Оплата для розробників — 90 доларів США за рік.

Схожими аналогами є Huawei AppGallery, Samsung Galaxy Store від Huawei та Samsung відповідно.

F-Droid — каталог програмних застосунків для Android, що містить тільки вільне/open source програмне забезпечення. Він не потребує реєстрації та позначає в описі кожного застосунку використання реклами, відстеження дій користувача, тощо, з посиланням на репозиторії з сирцевим кодом застосунків.

Itch.io — сайт, що розповсюджує ігри, зокрема інді-ігри (з англ. Indie від “independent” — незалежний), тобто ігри, створені незалежно від фінансової підтримки великих видавців. Оскільки цей сайт орієнтований для індивідуальних розробників, видавництво ігор на ньому є безкоштовним, і наразі сайт вже має близько 200000 ігор. Розповсюджує

ігри для систем Windows, macOS, Linux, Android, тощо.

Google Play, Amazon Appstore та їх аналоги мають велику кількість користувачів та додатків, проте мають права використання, які можуть заборонити видавництво певних додатків. Тим часом, F-Droid та itch.io є більш відкритими, і якщо враховувати, що видавництво додатків на цих платформах є безкоштовним, то стає зрозумілим, чому там є певний контент, якого немає на Google Play.

Itch.io має меншу кількість видавців та розробників у порівнянні з Google Play, проте відрізняється типом монетизації: на itch.io майже відсутня рекламна монетизація. Більшість проектів зовсім безкоштовні (розроблені чисто заради хобі), а інші розповсюджуються за стандартним принципом “Pay to play”.

Також перевагою F-Droid та itch.io в цілях нашої роботи є їх вільний сирцевий код та відсутність digital rights management, що може спростити їх дослідження та роботу з ними.

Недоліком itch.io є той факт, що він не має власне додатку для встановлення програм для Android. Такий клієнт офіційно існує лише для Windows, macOS та Linux — система Android підтримується, але встановити такі ігри можна лише вручну, завантаживши з сайту .apk файл, і потім вручну перевіряти наявність оновлень.

Саме тому за основу проекту Mitch взято сервіс itch.io — це перший власне Android-додаток, що надає доступ до цього сервісу, схожий на Google Play.

Також постає питання, як саме розповсюджувати сам додаток Mitch. На жаль, правила використання Google Play не дозволяють розповсюджувати інші засоби онлайн-дистрибуції, оскільки вони вважаються як конкуренти. Тому Mitch буде розповсюджуватися лише на F-Droid та власне на сайті itch.io.

2 ДОСЛІДЖЕННЯ РОБОТИ СЕРВІСУ ITCH.IO

2.1 Початок дослідження

Перше, що було зроблено перед початком розробки фронтенду Mitch, це отримання дозволу від власників сервісу itch.io. Оскільки ним володіє невелика компанія, отримання дозволу виявилось нескладним.[8] Це варто було зробити з двох причин:

По-перше, було невідомо, чи є у команди itch.io плани на створення власного офіційного Android-додатку; в такому разі проєкт Mitch виявився би набагато менш корисним.

По-друге, можливо, створення неофіційних фронт-ендів певних чином порушує Умови використання itch.io чи певні інші правила.

На щастя, виявилось, що розробники itch.io поки не планують створювати власний додаток, і вони з радістю дозволять іншим людям зробити неофіційний фронтенд, за умови, що код буде відкритим.

Також один із розробників звернув увагу на те, що вже існує клієнт для персональних комп'ютерів з вільним кодом, і що варто розглянути його код, аби зрозуміти, як саме працювати з їх API.[8]

Дійсно, найкращим варіантом було б розглянути вже існуючий код для авторизації, завантаження, встановлення, оновлення файлів з itch.io, тощо. В такому разі є два варіанти: можна або напряму запускати існуючий код всередині нашого додатку, або ж можна переписати код під систему Android.

Перший варіант виявився надто складним, оскільки існуючий код, яким користується офіційний клієнт для Windows, macOS та Linux, написаний на мові Go.[9] Єдиними офіційно підтриманими мовами для операційної системи Android є Java та Kotlin.

Існує проєкт Go Mobile, який дозволяє запускати Go на мобільних пристроях, проте у нього є недоліки:[10]

Проєкт є експериментальним.

Він передбачає або створення додатку повністю на Go, або ж створення додатку на Java/Kotlin з викликом деяких функції Go. Створення додатку повністю на Go значно ускладнює розробку користувацького інтерфейсу, з іншого боку виклик лише декількох функцій Go в нашому випадку виконати досить складно, оскільки обсяг коду дуже значний.

Другий варіант — власноруч переписати код на мову Kotlin — є досить складним з тієї ж причини. Кількість строк коду в їх проекті Go перевищує 10,000, до того ж не завжди є документація.

Насправді існує і третій варіант: замість того, щоб брати за основу їх клієнт, написаний на Go, можна дослідити роботу їх веб-інтерфейсу. Цей підхід має свої переваги:

Досліджувати роботу додатків, основаних на Web-технологіях, дуже просто, завдяки таким інструментам, як Firefox Developer Tools, або Curl.

Веб-інтерфейс можна вбудувати в існуючий Android-додаток за допомогою елементів WebView, що спростить розробку, оскільки не доведеться самому писати більшу частину користувацького інтерфейсу.

Тому було обрано третій варіант.

Розглянемо приклад дослідження поведінки Web-інтерфейсу під час завантаження гри.

2.2 Приклад дослідження роботи веб-додатку itch.io

Спочатку розглянемо варіант Pay to play, в якості прикладу використаємо гру Super Hexagon, що коштує 3 долари США.

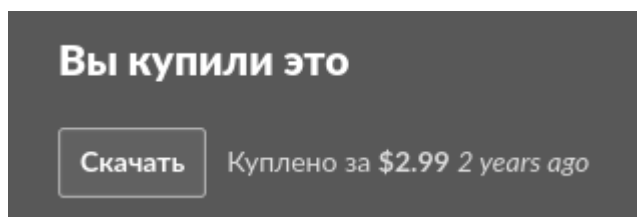


Рисунок 1 — Елемент веб-сторінки з

посиланням для завантаження купленої гри

Цей випадок виявився найпростішим. Після того, як користувач купив гру, можна зайти на основну сторінку гри (в цьому випадку <https://terrycavanagh.itch.io/super-hexagon>). На сторінці бачимо кнопку, що дозволяє завантажити гру (рис. 1). Це посилання має форму [https://terrycavanagh.itch.io/super-hexagon/download/\[ідентифікаційний код транзакції\]](https://terrycavanagh.itch.io/super-hexagon/download/[ідентифікаційний код транзакції]). Перейшовши за цим посиланням, бачимо сторінку, зображену на рисунку 2.

Спасибо за покупку Super Hexagon от Terry Cavanagh.

Вы заплатили за это \$2.99 2019-05-25 21:24:40, указав email goldenappl@protonmail.com. У этой страницы есть уникальная ссылка с вашим платежом. Вы всегда сможете скачать файлы последних версий с этой страницы. Вам также должно прийти email со ссылкой на эту страницу. Если вы вдруг потеряете ссылку, вы можете запросить отправить её повторно на email через нашу [форму обратной связи](#).

Эта покупка закреплена за аккаунтом [gardenapple](#).

Скачать	Super Hexagon [Windows] 24 MB	Jul 01, 2015
Скачать	Super Hexagon [Mac] 23 MB	Jul 01, 2015
Скачать	Super Hexagon [Linux] 28 MB	Jul 01, 2015
Скачать	Super Hexagon [Android] 26 MB	Jul 01, 2015

[Скачивание не началось?](#)

Рисунок 2 — Власне сторінка, що дозволяє завантажити гру

Тепер бачимо 4 посилання, що дозволяють завантажити конкретну версію гри, в цьому випадку це посилання на версії ігор для Windows, macOS, Linux та Android.

Поруч із кожною кнопкою наявна деяка інформація, зокрема дата видання, розмір файлу, та платформи, для яких призначений конкретний файл. Всі ці елементи мають власні HTML-класи та атрибути, що дають можливість автоматично обробити цю інформацію.

Розглянемо кнопки з написом “Скачать”: на цей раз вони не вказують на конкретну адресу, а виконують певний код, написаний на JavaScript. Щоб проаналізувати, що саме робить цей код, скористаємося інструментами розробника, які вбудовані в сучасні браузері (в нашому випадку Mozilla Firefox).

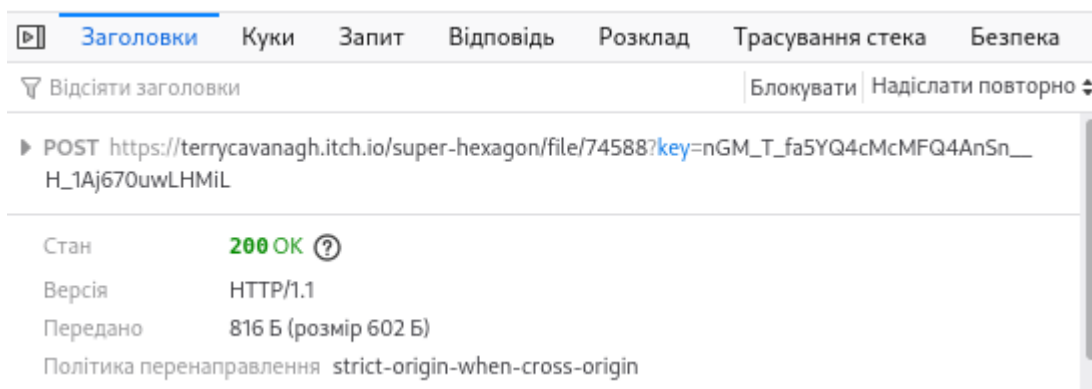


Рисунок 3

—Заголовок запиту, виконаного в JavaScript сайту

Ці інструменти дозволяють прослідкувати HTTP запити та відповіді, в тому числі запити, виконані мовою JavaScript. Бачимо POST запит (на рис. 3) без HTTP тіла, де адреса запиту має формат `https://terrycavanagh.itch.io/super-hexagon/file/[ідентифікаційний код файлу]?key=[ідентифікаційний код транзакції]`. Код транзакції співпадає з тим, що знаходиться у URL сторінки завантаження, отже він нам відомий.

Код файлу можна знайти напряму в HTML-документі, бачимо, що з кноп-

ками “Скачать” пов’язаний атрибут `data-upload_id=` “[ідентифікаційний код файлу]”, що співпадає з кодом у URL для POST-запиту. Ці кнопки також мають свій власний клас `download_btn` (англ. “download button” — кнопка завантаження), це дозволить нам чітко дізнатися коди для всіх доступних файлів.

Розглянемо відповідь на наш запит, яку надіслав сервіс.

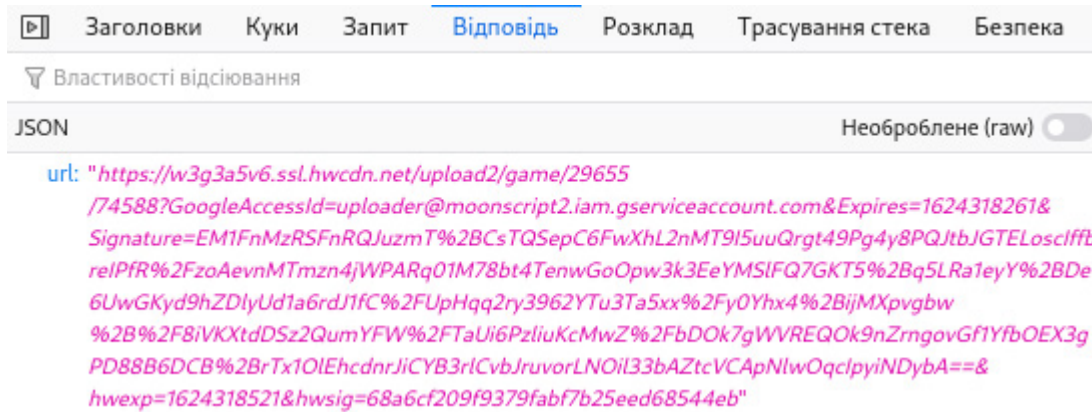


Рисунок 4 —

Відповідь на запит з рисунку 3

Бачимо, що сервер надсилає у відповідь дуже довгий URL, і вже перейшовши на нього, браузер починає власне завантажувати потрібний файл. Судячи з наявних параметрів `Expires` та `Signature`, це посилання буде дійсним лише протягом певного часу, і також воно певним чином прив’язано до конкретного акаунту.

Дійсно, якщо перейти за цим посиланням на декілька годин пізніше, до доступу до файлу не надається. Також, якщо цей запит надіслати за допомогою інструменту `curl`, побачимо помилку `401 Unauthorized`, що свідчить про те, що необхідно певним чином автентифікуватися.

На щастя, можна легко надіслати певні дані для автентифікації, оскільки вони зберігаються в `cookies` браузера. Дійсно, якщо скопіювати `header Cookie` з запиту, виконаного браузером, у запит, виконаний `Curl`, то отримаємо бажаний результат — посилання дійсно працює, і ми можемо завантажити потрібний контент.

Отже, аби завантажити певну гру, знаючи посилання на її сторінку в крамниці `itch.io`, необхідно зробити наступне:

Користувач авторизується на сайті за допомогою веб-інтерфейсу.

`Cookie`, які веб-інтерфейс отримує у відповідь на авторизацію, зберігаються.

Аби отримати конкретні посилання для завантаження файлів, робимо наступне:

Завантажуємо основну HTML-сторінку в крамниці для заданої гри.

За допомогою HTML-парсера знаходимо елемент кнопки “Скачать”, вона знаходиться в області з класом `purchase_banner` і має клас `button`.

Дивимося на атрибут кнопки `href`, це і є посилання для сторінки завантаження.

Необов'язково: як тільки користувач один раз купляє гру, посилання на сторінку завантаження залишається назмінним, тому можна додати його в певний кеш нашого додатку. В такому разі нам не доведеться повторювати кроки 1-3 цього пункту при подальшому доступі до гри.

Отримавши посилання на сторінку завантаження, робимо наступне:

Завантажуємо HTML-сторінку, що містить посилання на конкретні файли для гри.

За допомогою HTML-парсера знаходимо всі елементи з класом `download_btn`.

Для кожної кнопки дивимося на атрибут `data-upload_id`, це є ідентифікаційним кодом для певного файлу.

Також аналізуємо дані, наявні поруч з кнопками завантаження (див. рис. 2), такі як операційні системи, назва файлу, розмір файлу, та ін.

Обираємо файл, який призначено для платформи Android (поруч із ним буде елемент із класом `icon_android`). Зазвичай він буде лише один, інші випадки розглянемо в наступному розділі.

Для обраного файлу дивимося на атрибут `data-upload_id` кнопки “Скачать”, це є ідентифікаційний код файлу.

Надсилаємо POST-запит за адресою `/file/[ідентифікаційний код файлу]?key=[останній елемент URL адреси сторінки завантаження]`, також додаємо `header Cookie`, який отримали під час авторизації в кроці а).

Отримаємо відповідь з тимчасовою адресою, яка дозволить завантажити файл.

Бачимо, що виконується багато кроків, проте кожен крок досить простий, і легко імплементується за допомогою існуючих бібліотек Java/Kotlin та засобів Android.

Недоліком цього методу є той факт, що такі алгоритми спираються на специфіку фронтенду (наприклад, конкретні HTML-класи), які можуть у будь-який момент змінитися. Вважатимемо, що в цьому випадку можна буде досить швидко оновити додаток, виправивши алгоритм, але в цілому процес завантаження файлів не повинен різко змінюватися.

3 РОЗРОБКА ДОДАТКУ ANDROID

3.1 Засоби розроблення

Як уже зазначалося раніше, для цієї роботи використовували інтегроване середовище розробки Android Studio та мова Kotlin. Компанія Google, що є розробником системи Android та середовища розробки Android Studio, рекомендує Kotlin для створення нових додатків.[4]

Одна з переваг мови Kotlin над Java є спрощення роботи з мультипоточністю. В мові Kotlin є поняття coroutine (з англ. “співпрограма”) — це програмний модуль, що забезпечує взаємодію з іншими модулями за принципом кооперативної мультизадачності: модуль може призупинити дію на заданому етапі та передає контроль іншому модулю, зберігаючи свій стек та положення. Вони є більш гнучкими за звичайні підпрограми.[11]

В нашому випадку мультипоточність буде використовуватися для виконання мережових запитів, коли цього потребує користувач. Скажімо, що користувач хоче завантажити нову версію гри, використовуючи алгоритм, описаний у минулому розділі. Під час виконання, скажімо, POST-запиту, алгоритм може зупинитися на невизначений срок часу, чекаючи на відповідь від сервера. Замість того, щоб блокувати потік, сопрограма просто в цей момент просто передасть контроль іншому модулю, наприклад, потоку, що обробляє користувацький інтерфейс. Це дозволяє заощадити час та ресурси, а також спрощує код, оскільки сопрограми є вбудованими в саму мову програмування, а не потребують використання певних API.

Для завантаження HTTP-даних використаємо бібліотеку OkHttp, яка має відкритий висхідний код, і широко застосовується в програмах Java. Однією з її переваг є можливість налаштування кешування, що нам знадобиться, оскільки ми періодично будемо перевіряти сторінки ігор, щоб знайти оновлення, і кешування даних може значно пришвидшити цей процес у випадках, якщо оновлення не сталося.

Для HTML-парсингу використаємо бібліотеку Jsoup, яка також є вільним програмним забезпеченням, що імплементує HTML5, зокрема дозволяє досить швидко і легко шукати елементи використовуючи DOM методи, аналогічні JavaScript-методам у браузерях.

Основна частина користувацького інтерфейсу використовує власне веб-інтерфейс сайту itch.io, завдяки вбудованій у Android можливості WebView. WebView дозволяє показувати, а також маніпулювати веб-сторінки, інтегруючись з нативними елементами інтерфейсу. Це має декілька переваг:

- Заощаджуємо час, не витрачаючи його на імплементацию функціоналу, який вже існує на сайті.
- Використаємо авторизацію сайту itch.io, зчитуючи Cookie.
- Інтерфейс виглядатиме так само, як і мобільний сайт, що може бути зручним для користувача, оскільки не доведеться звикати до зовсім нового інтерфейсу.

Такий самий підхід до розробки інтерфейсу використовує офіційний додаток Itch для Windows, macOS та Linux, а також інші програми, наприклад Slack, Discord, Steam

та інші. Для цих операційних систем зазвичай використовують фреймворк Electron, проте у випадку Android функціонал WebView вбудований і може використовуватися будь-яким додатком.

3.2 Графічний інтерфейс користувача

Оскільки наш додаток призначений для системи Android, варто дотримуватися таких же принципів розробки графічного інтерфейсу, як і інші мобільні додатки, тому притримуємося принципам матеріального дизайну.[12] Ідея дизайну полягає в інтерфейсі, поведінка і вигляд якого наслідують паперові картки в реальному житті.

Багато Android-додатків використовують такий елемент дизайну, як Top app bar — це полоса (“bar”), що знаходиться зверху (“top”) екрана, і показує певний контент та дії, пов’язані з поточним контекстом (“app”).

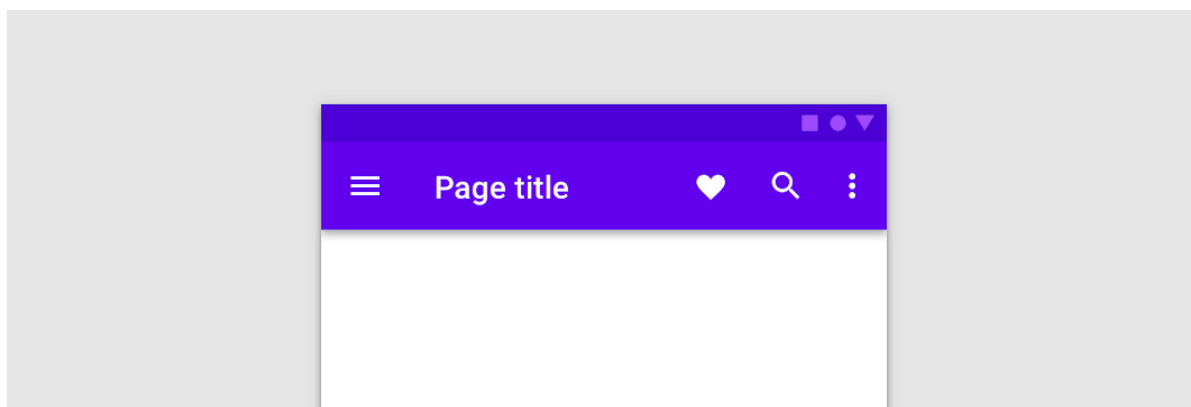


Рисунок 5 — Приклад типового top app bar інтерфейсу

На рисунку 5 бачимо кнопку у вигляді трьох горизонтальних полосок — це дає доступ до іншого функціоналу додатку. “Page title” є заголовком теперішнього контексту, з яким пов’язані всі інші кнопки.

Проблема постає у тому, що сайт itch.io вже має власні полоси зверху (рис. 6)

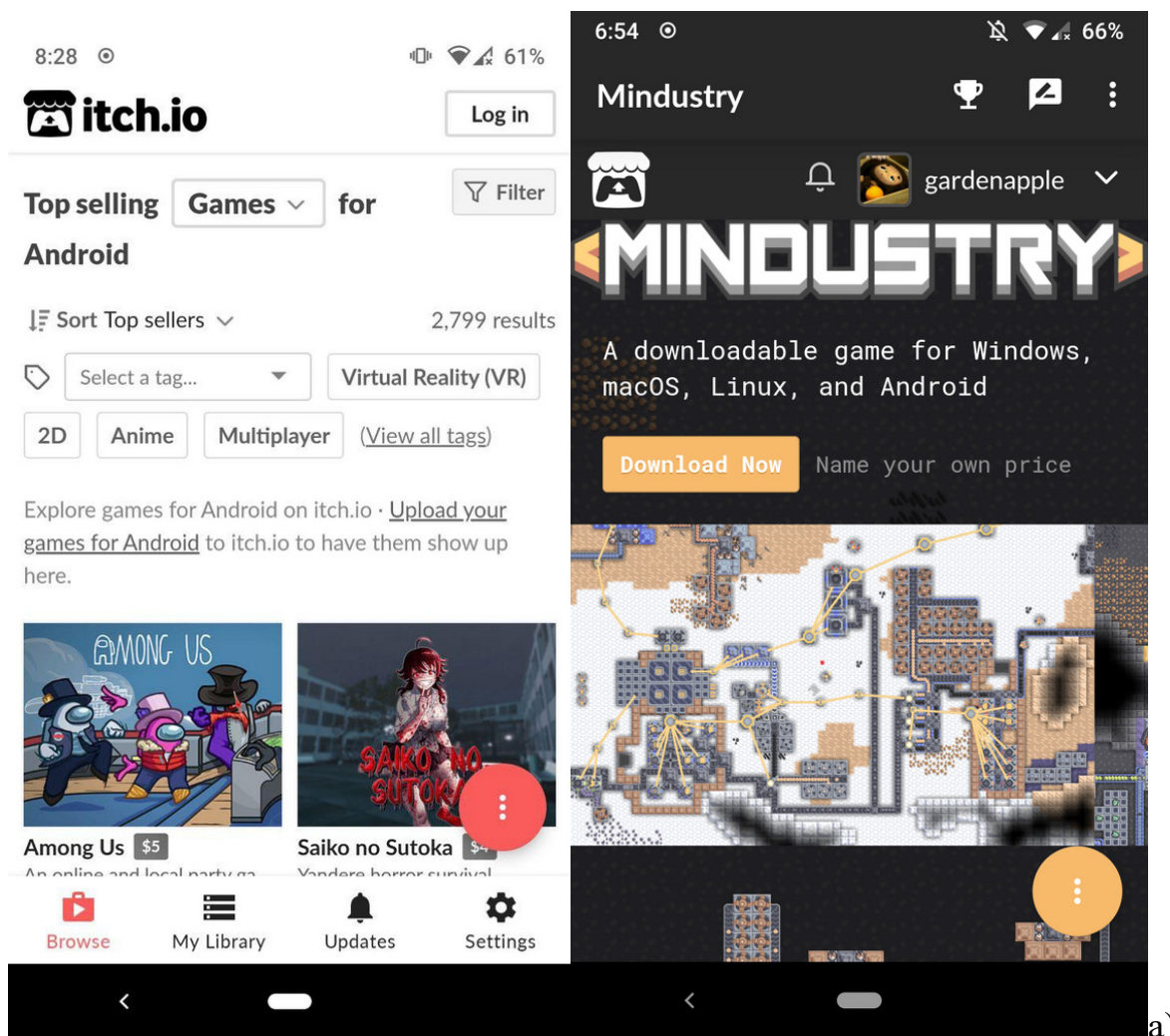


Рисунок 6 — Верхні полоси інтерфейсу сайту itch.io

Якщо використовувати веб-інтерфейс сайту, і поверх нього будувати мобільний інтерфейс, то не варто використовувати стандартний app bar. Замість цього ви-

користаємо менш вживаний елемент матеріального дизайну — bottom app bar (нижню полосу).

Кінцевий результат для оболонки над веб-інтерфейсом зображено на рисунку 7.



б)

Рисунок 7 — Веб-інтерфейс сайту, вбудований в інтерфейс додатку:
а — основна частина сайту; б — сторінка в крамниці для певної гри

В цьому випадку нижня полоса (рис. 7а) дає нам доступ до іншого функціоналу, наприклад, перевірки оновлень. Всі функції, окрім “огляду” (“browse”) мають нативний інтерфейс, тобто інтерфейс, не оснований на веб-сайті. Взагалі нативні елементи мають такі самі кольори, як і сайт, що дозволяє легко змішувати їх із веб-інтерфейсом.

Єдиний виняток — це власне сторінки ігор у крамниці, де видавець чи розробник гри може надати свою палітру кольорів (рис. 7б). В такому разі ми зховаємо bottom app bar, який стилістично не підходить до палітри, а замість цього додамо інші елементи, які співпадають за кольором, і при цьому теж дають доступ до функцій додатку.

3.3 Оновлення ігор

В минулому розділі ми проаналізували, як відбувається завантаження конкретного файлу для гри у веб-інтерфейсі `itch.io`, і як цю функцію можна імплементувати у додатку. За результатами дослідження, тепер побудуємо власне механізм перевірки оновлень.

Механізм полягатиме у тому, що періодично (наприклад, раз в добу), додаток завантажує сторінку, що містить інформацію про конкретні файли (пункт С нашого алгоритму). Далі слід перевірити, чи змінилися дані з цієї сторінки, наприклад, версія, назва файлів, ідентифікаційний номер файлу, час завантаження, тощо. Якщо інформація незмінна, то нічого не робимо, а інакше ми завантажуємо оновлення, або ж просто сповіщаємо користувача.

Варто зазначити, що є два способи публікації файлів на `itch.io`: вручну (через сайт) та автоматизовано (через офіційну програму чи API). Існують певні відмінності, які для нас важливі:

- При завантаженні файлів вручну надається новий ідентифікаційний номер, при цьому не можна задати тег з версією (як на рис. 2)
- При завантаженні файлів ідентифікаційний номер залишається незмінним, лише змінюється тег версії файла (наприклад, версія `v1.0.0` стає `v1.1.0`)

В будь-якому випадку оновлюється параметр дати, але він не є надійним показником, оскільки:

- Дата не зміниться, якщо оновлення було опубліковане в той же день, що і минуле.
- Формат дати залежить від регіону, в якому проживає користувач.

Тому краще спиратися лише на ідентифікатор файлу та тег версії.

Пропонуємо такий алгоритм:

- а) При завантаженні файлу гри зберігаємо його ідентифікатор та (за наявності) тег версії.
- б) Завантажуємо сторінку, що містить дані для файлів цієї ж гри (пункт С раніше зазначеного алгоритму).
- в) Намагаємося знайти файл з таким же ідентифікатором, як і наявний файл.
 - 1) Якщо знайшли файл з тим же ідентифікатором, дивимося, чи наявний тег версії, і чи він змінився.
 - Якщо тег версії наявний і він змінився, то завантажуємо файл з новою версією.
 - Інакше алгоритм завершується (файл не оновлюється).
 - 2) Якщо не знайшли файл з тим же ідентифікатором, то шукаємо, який файл

завантажити замість нього.

- Якщо існує файл з тією ж назвою, що і наявний (хоча із іншим ідентифікатором), завантажуюмо його.
- Інакше, якщо існує єдиний файл, що призначений для платформи Android, то завантажуюмо його.
- Інакше, виводимо сповіщення про те, що доступне оновлення. Натиснувши на сповіщення, користувач матиме змогу обрати, який саме файл завантажити.

В більшості випадків цей алгоритм не потребуватиме вводу від користувача, а просто або оновить гру, або не оновить.

У випадку, якщо у користувача встановлено досить багато ігор, можна запустили алгоритм декілька разів у паралельному режимі, тоді завдяки сопрограмам Kotlin матимемо непогану швидкодію. Проблема виникне у тому, що для кожної гри потрібно кожен раз завантажувати HTML-сторінку — це можна запобігти завдяки кешуванню в OkHttp, але загалом бажано не дуже часто виконувати ці операції, тому за замовчуванням додаток виконує перевірку оновлень 1 раз на добу.

ВИСНОВКИ

Оцінка одержаних результатів та їх відповідність сучасному рівню технічних знань і технологій.

В цій роботі ми порівняли різні засоби цифрової дистрибуції програмного забезпечення та навилися досліджувати роботу конкретних веб-сервісів за допомогою сучасних інструментів розробників. Завдяки цим результатам ми спроектували Android-додаток, який спрощує роботу з обраним веб-сервісом, відповідаючи сучасним вимогам графічних інтерфейсів.

Можливі галузі або сфери використання результатів роботи.

Додаток Mitch можна використовувати в цілях розваг, оскільки більша частину контенту, яка розповсюджується через сервіс itch.io — це саме відео-ігри. Перевагою Mitch над Google Play є практична відсутність реклами у додатках, а також той факт, що Mitch не потребує Google Services, тобто може використовуватися взагалі на будь-якому Android-пристрої.

Інформація щодо створення нового програмного продукту та соціально-економічну значущість роботи.

Створений в рамках цієї роботи додаток Mitch було опубліковано на сервісах itch.io та F-Droid; на даний момент має більше сотні користувачів, і ця кількість зростає.

Доцільність продовження розробок за відповідною тематикою.

Оскільки додаток Mitch опублікований і має користувачів, а також у певному розумінні є конкурентом Google Play, варто продовжити його розробку, додавши новий функціонал, щоб його використання було доцільним у порівнянні з альтернативами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Number of smartphone users worldwide from 2016 to 2026 [Електронний ресурс] — Режим доступу: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>
2. Гапп Б. Download distribution opening new doors for independent game developers [Електронний ресурс] - Режим доступу: <https://www.statesman.com/business/technology/download-distribution-opening-new-doors-for-independent-game-1409285.html>
3. Дуроше Х., Норбай Т., Чу К., Android Studio: An IDE built for Android». Android Developers Blog. [Електронний ресурс] — Режим доступу: <https://android-developers.blogspot.in/2013/05/android-studio-ide-built-for-android.html>
4. Kotlin is now Google's preferred language for Android app development [Електронний ресурс] : (TechCrunch) — Режим доступу: <http://social.techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/>
5. Digital Distribution Law and Legal Definition [Електронний ресурс] : (USLegal, Inc) – Режим доступу: <http://definitions.uslegal.com/d/digital-distribution/>
6. Google Play Store: number of apps 2020 [Електронний ресурс] : (Statista) – Режим доступу: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>
7. Number of available apps in the Amazon Appstore by quarter 2021.[Електронний ресурс] : (Statista) – Режим доступу: <https://www.statista.com/statistics/307330/number-of-available-apps-in-the-amazon-appstore/>
8. How would you feel about an unofficial Android itch.io client? - Режим доступу: <https://itch.io/t/495651/how-would-you-feel-about-an-unofficial-android-itchio-client>
9. <https://github.com/itchio/butler>
10. <https://github.com/golang/mobile>
11. Кнут. Д. 1.4.2: Сопрограммы // Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. Fundamental Algorithms. — 3-е изд. — : «Вильямс», 2006. — С. 229—236. — 720 с.
12. Introduction – Material Design [Електронний ресурс] : (Google) – Режим доступу: <https://material.io/design/introduction>

