Список використаних джерел

1. Самко М. Що таке Unity? LemonSchool. 2023. URI: https://lemon.school/blog/shho-take-unity.
2. Unity (ігровий рушій). Вікіпедія. 2023. URI: https://uk.wikipedia.org /wiki/Unity_(ігровий_рушій).

Мельник І. Б.,
здобувач вищої освіти першого
(бакалаврського) рівня,
спеціальності 122 «Комп'ютерні науки»,
освітньої програми «Комп'ютерні науки та інформаційні технології»;
науковий керівник:
Ставицький О. В.,
кандидат економічних наук, доцент,
доцент кафедри статистики,
ІТ та математичних методів в економіці;
Національна академія статистики, обліку та аудиту

FLUTTER КРОСПЛАТФОРМНИЙ ІНСТРУМЕНТ РОЗРОБКИ ДОДАТКІВ

Розробка додатків завжди була і залишається складною темою. Кожна платформа має свої унікальні можливості та технології. Flutter дозволяє, використовуючи спільну кодову базу, розробляти застосунки під Android, IOS, Windows, Linux, MacOS, Web.

Flutter – це інструментарій для розробки додатків створений компанією Google у 2017 році. Flutter має свій двигун для малювання інтерфейсу. Це дозволяє йому уніфікувати відображення компонентів на різних платформах. Для написання програм використовується мова Dart. Вона спеціально створена під цей інструментарій. Уся структура будується через комбінацію віджетів. Кнопка – віджет, текст – інший віджет [2].

Flutter надає ряд віджетів, які допомагають вам створювати програми, які відповідають Material Design. Програма Material починається з віджета MaterialApp, який створює низку корисних віджетів у корені вашої програми, включаючи Navigator, який керує стеком віджетів, визначених рядками, також відомих як «маршрути» [1].

Віджети це блоки з яких складається програма. Кожен блок має свій набір параметрів. Фінальний варіант виглядає у вигляді дерева. Один віджет може мати декілька дочірніх. Розробник може створити свій віджет, щоб постійно не дублювати повторювальну логіку. Також інструментарій здатний реагувати на зміни параметрів і оновлювати тільки потрібні елементи, тому розробнику більшу частину часу не потрібно думати про компоненти які залежать від змінюваних даних. Якщо ефективно оновляти структуру не виходить (це може відбутися через неправильне використання або неочевидну логіку), тоді треба самостійно змінювати код, щоб краще пояснити наміри.

Щоб легше виявляти проблеми Flutter надає багато інструментів: UI Inspector, Performance view, Memory view, Debugger, CPU Profiler, Network view. Використовуючи їх можна детально подивитись, що і як будується та працює програма. UI Inspector – дозволяє побачити структуру інтерфейсу, переглянути відступи, розміри та параметри. Performance view – дає розуміння про стабільність роботи програми та побачити її продуктивність. Debugger – дозволяє по стрічках побачити роботу програми і подивитись значення параметрів у цей момент.

Наразі розробники Flutter працюють над власним графічним ядром. Що не так з існуючим? Він не дає повного контролю над процесом побудови зображення. Потрібно постійно генерувати шейдери для швидкої роботи, цей процес не миттєвий і примушує помітно завмерти програму. Зі своїм ядром можна оптимізувати під конкретні потреби інструментарію та прибрати непотрібні затримки. Також це дозволяє зробити пряму підтримку 3D. Раніше треба було під’єднувати стороні бібліотеки і надіятись, що вони добре працюють на всіх платформах та підтримують потрібний функціонал.

Як же підключаються власні бібліотеки? Усі пакети публікуються на сайті pub.dev. Розробник може вибрати потрібний пакет і додати його у файл в проекті під назвою pubspec.yaml. Flutter автоматично завантажить потрібні файли, та повідомить про проблеми з залежностями, якщо вони будуть. Далі залишається тільки почати використання у своєму проекті.

Розглянемо переваги використання Flutter.

- Спільна кодова база. Не потрібно окремо писати код під кожну платформу. Достатньо написати під одну і воно запуститься на всіх. Для доступу до унікальних можливостей платформи можна написати обгортку і використовувати її в програмі.
- Сталий вигляд. Інтерфейс виглядає однаково на усіх платформах. Це дозволяє не витрачати зайвий час на налаштування компонентів на різних платформах.
- Відкритий код. Flutter має відкритий код, тому при потребі можна подивитись на внутрішню реалізацію, змінити або додати щось своє.
- Простота. Потрібно менше людей, щоб створити програму. Не потрібно вивчати новий інструментарій під кожну систему. Це пришвидшує та спрощує розробку для невеликих команд.

Розглянемо деякі недоліки кроссплаформеної розробки.

- Унікальні можливості та особливості кожної платформи. Кожна платформа підтримує різні можливості і має різний алгоритм взаємодії з ним. Так як код працює з усіма платформами потрібно постійно перевіряти чи підтримується потрібний функціонал.
- Продуктивність на слабких девайсах. Через використання власного двигуна для малювання на менш потужних девайсах можуть бути деякі проблеми. Це надзвичайно помітно при запуску Flutter у Web на телефонах.

Де ж використовувати дану технологію? Вона підходить для невеликих команд або коли програма не зав’язана на унікальних можливостях платформ. Інструментарій надає можливість зробити красивий і чуйний інтерфейс. Багато потрібних компонентів йдуть з коробки.

Підсумовуючи все вище сказане, Flutter варто спробувати для розробки кросплатформного рішення. Він дозволяє швидко і зручну будувати красиві та чуйні додатки. Має великий набір інструментів для діагностики та постійно отримує оновлення.

Список використаних джерел

1. Flutter documentation. Flutter. 2022. URI: https://docs.flutter.dev/.

2. Flutter (software). Wikipedia. 2017. URI: https://en.wikipedia.org/wiki/Flutter_(software).