Гречмак Д.В., студент
Харківський національний університет радіоелектроніки, м Харків
Кафедра автоматизації проектування обчислювальної техніки

ОСОБЛИВОСТІ УПРАВЛІННЯ FLUTTER FRAMEWORK ТА FLUTTER ENGINE

За винятком деяких випадків, жоден код Flutter Framework не виконується без запуску рендерингу Flutter Engine. Винятки:

- Gesture / Жест (взаємодія користувача з екраном)
- Повідомлення платформи (повідомлення, створені пристроєм)
- Повідомлення пристрою (повідомлення, які стосуються зміни стану пристрою, наприклад, орієнтація, програма, надіслана у фоновому режимі, попередження пам'яті, налаштування пристрою, тощо)
- Future або http-відповіді

Якщо якийсь код, пов'язаний з жестом, виконується і викликає візуальну зміну або якщо використовується timer для завдання періодичності завдання, яке призводить до візуальних змін (наприклад, анімація), то як це працює?

Якщо ми хочемо, щоб відбулася візуальна зміна або якийсь код виконувався на основі таймера, то нам потрібно повідомити Flutter Engine, що щось має бути змальовано. Зазвичай при наступному оновленні Flutter Engine звертається до Flutter Framework для виконання деякого коду і надає нову сцену для рендерингу.

Поведінка двигуна Flatter виглядає наступним образом:

- Деякі зовнішні події (жест, http-відповіді тощо) або навіть futures можуть запускати завдання, які призводять до необхідності оновлення відображення. Відповідне повідомлення надсилається Flutter Engine (Schedule Frame).
- Коли Flutter Engine готовий розпочати оновлення рендерингу, він створює Begin Frame запит.
- Ці завдання можуть повторно створити запит для пізнішого малювання

- Далі Flutter Engine відправляє Draw Frame, який перехоплюється Flutter Framework, який шукатиме будь-які завдання, пов'язані з оновленням макета з точки зору структури та розміру.
- Якщо на екрані є щось, що потрібно намалювати, то нова сцена (Scene) для візуалізації відправляється у Flutter Engine, який оновить екран.
- Потім Flutter Framework виконує всі завдання, які будуть виконуватися після завершення рендерингу (PostFrame callbacks), та будь-які інші подальші завдання, не пов'язані з рендерингом.

Література

1. Hunko M.A., Tkachov V.M. Development of a module for sorting the ipaddresses of user nodes in cloud firewall protection of web resources. Дев’ята міжнародна науково-технічна конференція «Сучасні напрями розвитку інформаційнокомунікаційних технологій та засобів управління». 2019. С. 30.
2. Tkachov V. Technology of Load Balancing in Anonymous Network Based on Proxy Nodes Cascade Platform / V. Tkachov, M. Hunko, M. Bondarenko, S. Artyomov // Четверта міжнародна науково-технічна конференція «Комп’ютерні та інформаційні системи і технології». Збірка наукових праць. Харків: ХНУРЕ. – 2020. – С. 82.
3. Tkachov V. Principles of Constructing an Overlay Network Based on Cellular Communication Systems for Secure Control of Intelligent Mobile Objects / Vitalii Tkachov, Andriy Kovalenko, Mykhailo Hunko and Kateryna Hvozdetska // Информационные технологии и безопасность. Материалы XIX Международной научно-практической конференции ИТБ-2020. – К.: ООО «Инжиниринг», 2020.
4. Гунько М. А. Розробка моделі інтелектуальної мобільної системи для своєчасного запобігання механічних перешкод / М. А. Гунько // «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 49)» : матеріали Міжнар. наук. Інтернет-конф., 10 червня 2020 р. – Тернопіль, 2020. – С. 7–8.
5. Krivoulya G. Implementation of mobile eye tracking systems for preventing emergency situations based on monitoring of driver behavior / Krivoulya G., Tokariev V., Tkachov V., Hunko M // Проблеми інформатизації : тези доп. 7-ї міжнар. наук.-техн. конф., 13-15 листопада 2019 р., м. Черкаси, м. Харків, м. Баку, м. Бельсько-Бяла : [у 3 т.]. Т. 3 / Черк. держ. технолог. ун-т [та ін.]. – Харків, 2019. – С. 36.