

Существующие платформы позволяют повысить точность и эффективность диагностики аритмии у пациентов не только группы высокого риска, позволяя пациенту вести нормальную жизнь, находясь под дистанционным наблюдением в режиме реального времени с помощью амбулаторного беспроводного датчика ЭКГ, но и у людей с устройством мобильного здравоохранения появляется возможность отслеживать свои показания на основе ЭКГ и предоставлять их лечащему врачу, в режиме реального времени.

Список литературы

1. О Стратегии развития здравоохранения в Российской Федерации на период до 2025 года : указ Президента РФ № 254 от 6 июня 2019 г. URL: <https://www.garant.ru/products/ipo/prime/doc/72164534/> (дата обращения: 25.09.2023).
2. Mechanic O. J., Persaud Y., Kimball A. B. Telehealth Systems. 2022, Sep. 12 // StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2023, Jan. PMID: 29083614 (дата обращения: 25.09.2023).
3. Кристоф Леклерк, Хеннинг Витт, Герхард Хиндрикс [и др.]. Носимые устройства, телемедицина и искусственный интеллект при аритмиях и сердечной недостаточности: материалы «круглого стола» Европейского общества кардиологов по сердечно-сосудистым заболеваниям (EP Europace). Сентябрь 2022. Т. 24, вып. 9. С. 1372–1383. URL: <https://doi.org/10.1093/europace/euac052> (дата обращения: 25.09.2023).
4. Baman J. R., Mathew D. T., Jiang M., Passman R. S. Mobile Health for Arrhythmia Diagnosis and Management // J. Gen. Intern. Med. 2022. Vol. 37 (1). P. 188–197. doi: 10.1007/s11606-021-07007-w

УДК 004.45

ОСОБЕННОСТИ ВЗАИМОДЕЙСТВИЯ UNITY-ПРИЛОЖЕНИЙ С ПЕРЕНОСИМЫМИ УСТРОЙСТВАМИ ПО ПРОТОКОЛУ BLUETOOTH LOW ENERGY

Н. А. Горбунов¹, Ю. В. Мамелина², А. Ю. Тимонин³

^{1,2,3}Пензенский государственный университет, Пенза, Россия

¹nik.gorbunov.02022002@mail.ru

²yurovsky01@mail.ru

³timonin@pnzgu.ru

Аннотация. Рассматриваются методы внедрения возможностей протокола Bluetooth Low Energy в приложения Unity для устройств на базе Android с целью создания многопользовательских интерактивных окружений. В качестве используемого языка программирования выбран Java. Описаны требования к таким приложениям. Также рассмотрены основные характеристики передачи данных по протоколу Bluetooth Low Energy.

Ключевые слова: программирование плагинов под Android, портативные устройства, Bluetooth Low Energy, Java программирование, Unity

INTERACTION FEATURES OF UNITY APPLICATIONS WITH PORTABLE DEVICES VIA THE BLUETOOTH LOW ENERGY PROTOCOL

N. A. Gorbunov¹, Y. V. Mamelina², A. Y. Timonin³

^{1,2,3}Penza State University, Penza, Russia

¹nik.gorbunov.02022002@mail.ru

²yurovsky01@mail.ru

³timonin@pnzgu.ru

Abstract. This article discusses methods for introducing the capabilities of the Bluetooth Low Energy protocol into Unity applications for Android devices in order to create multi-user interactive environments. Java was chosen as the programming language used. The requirements for such applications are described. The main characteristics of data transmission via the Bluetooth Low Energy protocol are also discussed.

Keywords: Android plugins programming, portable devices, Bluetooth Low Energy, Java programming, Unity

Среда разработки Unity и ее инструментарий имеют ряд особенностей. Большинство функций для работы с операционной системой Android изначально встроено в метаданные движка, например, взаимодействие с файловой системой, звуком, камерой, Wi-Fi – с теми инструментами, которые чаще всего используются при разработке приложений. Но Bluetooth не входит в часто применяемый функционал, поэтому, для его интеграции следует использовать сторонние библиотеки или писать собственные для обращения к API операционной системы. Основным языком ОС Android является Java, официальная документация включает в себя примеры, написанные именно на этом языке, поэтому взаимодействие с API желательно осуществлять на нем или на языке Kotlin.

В настоящее время Bluetooth поддерживается двумя видами устройств:

- Bluetooth Classic – применяется в беспроводных наушниках, автомобильных сигнальных системах и там, где обязательно непрерывное соединение для передачи аудиоданных;

- Bluetooth Low Energy (сокр. BLE) – та же технология, но с низким энергопотреблением. Появилась с версии Bluetooth 4.0 и используется почти повсеместно в приложениях, датчиках, устройствах, которым важно сохранять заряд на протяжении долгого времени.

Поскольку в большинстве современных устройств используются большое количество датчиков и сенсоров, BLE стал наиболее часто используемым протоколом связи.

Основная задача состоит в написании Android приложения, которое должно уметь взаимодействовать с BLE устройствами.

Первое, что необходимо для реализации – интеграция Java кода в среду разработки Unity. Для встраивания сторонних языков, необходимо написать плагин и включить его в сборку основного проекта. Это требует использования встроенных классов `AndroidJavaObject`. В переменной такого типа будет храниться экземпляр класса библиотеки. Для правильной работы плагина, необходимо использовать

шаблон проектирования Singleton. С его помощью, экземпляр класса будет сохранять свое состояние на протяжении всего жизненного цикла основного приложения, что важно при работе с Bluetooth. Далее нужно взаимодействовать с контекстом (Context) активности для обращения к базовым функциям, которые как раз включают в себя объекты и методы BLE.

При написании Java плагина для основного приложения можно с самого начала столкнуться с неявными ошибками. Во-первых – предоставление разрешений для операционной системы. При взаимодействии с большинством методов из Android API требуются разрешения от пользователя. При использовании BLE обязательными являются следующие:

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
```

Разрешение ACCESS_COARSE_LOCATION (отслеживание местоположения) Google считает «опасным» и для него требуется обязательное согласие пользователя [1].

Если приложение запущено на телефоне с Android версии 8.0 и выше, то требуются дополнительные разрешения:

```
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
```

Нужно учитывать, что на Android более низких версий данных разрешений не существует, и при запросе будет возвращено значение 2, что означает: «пользователь отклонил навсегда».

Разрешения запрашиваются асинхронно, и можно вызвать сразу несколько методов, но, например, у того же ACCESS_COARSE_LOCATION текст уведомления отличается, и если его запросить вместе со всеми, то оно автоматически отклонится операционной системой. Это нужно учитывать и разделить разрешения на группы, после чего вести опрос по очереди.

Android BLE API – это низкоуровневые операции, в реальных приложениях необходимо использовать несколько слоев абстракции, из-за чего следует реализовать самостоятельно:

- очередь на отправку и прием данных;
- отслеживание логов и обработчик ошибок;
- многопоточный доступ.

Все вышеперечисленное можно реализовать, как в плагине, так и из основного приложения.

Многопоточный доступ следует разобрать подробнее. Скорее всего, приложение, которое реализовано для взаимодействия с BLE устройствами, должно будет работать в фоновом режиме. Но проблема заключается в том, что Unity-приложение останавливает свою активность при блокировке или сворачивании. Для решения проблемы работы в фоновом режиме есть 2 возможных варианта, которые удобны для исполнения поставленной задачи:

1. Если скрипт работает не в основном потоке, то, при остановке активности, он продолжит исполняться какое-то время. До тех пор, пока операционная система не решит, что процесс занимает слишком много процессорного времени, после чего остановит его принудительно. Отталкиваясь от этой информации, можно

реализовать алгоритм взаимодействия и работы с Java плагином не из основного потока.

Данный вариант решения подойдет для тех приложений, которые должны продолжать работу с BLE на протяжении короткого времени, равному приблизительно одному часу. Но реализация многопоточности в Unity требует использования определенных «костылей».

2. Одно из лучших и верных решений, это создание сервиса, который будет работать в фоне, пока пользователь не выключит приложение принудительно. Такой вариант подходит для задач по обеспечению соединения и передаче данных по BLE на протяжении долгого времени, например, если задача состоит в отслеживании присутствия устройства. Написание собственного сервиса требует углубленных знаний Java и ОС Android. Также нужно учитывать то, что основное приложение будет находиться в паузе, поэтому вся логика должна быть прописана внутри плагина.

Также, при взаимодействии приложения с BLE устройствами [2] в фоновом режиме, нужно учесть, что:

- Запуск и остановка сканирования более 5 раз за 30 секунд временно отключает его в Android, начиная с 7 версии.
- Поиск устройств в Android, начиная с 7 версии доступен только 30 минут, после чего Android установит параметр `SCAN_MODE_OPPORTUNISTIC`.
- В фоновом режиме в Android, начиная с 8.1 версии, сканирование без фильтров блокируется. При отсутствии фильтров в `ScanFilters` система приостановит сканирование, когда экран выключен, и возобновит его при повторной активации экрана.
- Для работы приложения или сервиса необходимы разрешения `FOREGROUND_SERVICE` и `REQUEST_IGNORE_BATTERY_OPTIMIZATIONS`.

Отправка и прием данных по протоколу BLE реализованы в асинхронных методах. При их передаче нельзя, не дожидаясь окончания, начинать отправлять следующую информацию, иначе операционная система прервет подключение. Исходя из всего вышесказанного, реализовывать очередь придется вручную. Возврат ответа на запись или чтение приходит в соответствующих Gatt callback, но на прием существует 2 прерывания: `OnReadCharacteristic` и `OnChangeCharacteristic`. В какое из них придет ответ, зависит от устройства, к которому приложение должно подключиться, поэтому при реализации универсального плагина, следует учесть этот факт. Для передачи данных на отправку из основного приложения Unity в библиотеку, нужно использовать базовые типы данных, такие как `int`, `string` и массивы `Byte`, так как обращение к библиотекам через `AndroidJavaObject` довольно ресурсоемкое.

Список литературы

1. Developer.android.com: Bluetooth permissions. URL: <https://developer.android.com/guide/topics/connectivity/bluetooth/permissions#availability> (дата обращения: 03.10.2023).
2. Habr.com: Android Bluetooth Low Energy (BLE) — готовим правильно, часть #1 (scanning) // URL: <https://habr.com/ru/articles/536392/> (дата обращения: 07.10.2023).