

РАЗРАБОТКА ПРИЛОЖЕНИЯ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ НА ANDROID

Пономарев И.В.

Научный руководитель: Шерстнев В.С.

Томский политехнический университет

e-mail: example@example.ru

Введение

Виртуальная реальность - это созданный техническими средствами мир, в котором человек ощущает себя близко к тому, как он себя ощущает в реальном мире. Виртуальная реальность находит применение в проектировании зданий и автомобилей, обучении солдат, лётчиков, космонавтов и медиков, а также в сфере развлечений.

На данный момент разными компаниями разрабатывается аппаратное и программное обеспечение для полного выхода в виртуальную реальность: Omni, Oculus Rift, Google и другие. Вполне может быть, что с развитием высоких технологий в этой сфере виртуальная реальность займет прочное место в нашей жизни и обеспечит людей огромным, практически безграничным пространством для ведения любых дел.

В данной статье будет рассмотрена реализация приложения с использованием Google Maps Android API для доступа к панорамным снимкам улиц, просмотр которых осуществляется отклонением смартфона в пространстве.

Используемая среда разработки - Android Studio. Приложение написано на Java.

Акселерометр и гироскоп

Акселерометр (G-сенсор) - это датчик, который измеряет проекцию кажущегося ускорения.

Гироскоп - датчик, который служит для определения ориентации устройства в пространстве, для отслеживания его перемещения.

Для лучшей функциональности используют сочетание гироскопа и акселерометра, так как первый чувствителен к изменению положения устройства, второй к линейным ускорениям.

Гироскоп является обязательным элементом в смартфоне, чтобы использовать приложения виртуальной реальности, потому что именно он позволяет позиционировать смартфон в пространстве.

Для чтения состояний акселерометра и гироскопа создаются два отдельных класса. Эта информация становится доступной, регистрируя слушатель `SensorEventListener`. Он заключает в себе два метода: `onSensorChanged(SensorEvent event)` и `onAccuracyChanged(Sensor sensor, int accuracy)`. Первый метод вызывается при возникновении события акселерометра или

гироскопа. Второй срабатывает при изменении точности.

Чтобы зарегистрировать слушатель `SensorEventListener`, используется системный сервис `SensorManager`.

После регистрации слушателя в методе `onSensorChanged(SensorEvent event)` становятся доступны события `SensorEvent`. При этом данный метод вызывается только при изменении состояния датчика. Частота, с которой обновляется состояние датчика, задается при регистрации слушателя.

Google Maps Android API

Этот API-интерфейс позволяет добавить карты на основе данных Google Карт, и он автоматически управляет доступом к серверам Google Карт, загрузкой данных, отображением карт и реакцией на жесты, выполняемые на картах.

Предварительно необходимо установить и настроить Google Play Services SDK, зарегистрировать проект, получить ключ Google Maps API. После выполнения этих действий нужно добавить декларацию в элемент `<application>` файла `AndroidManifest.xml`. Это укажет версию служб Google Play, с которой было скомпилировано приложение.

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```

Чтобы добавить карты в приложение нужно выполнить следующие шаги:

1. Добавить объект `Fragment` в операцию `Activity`, которая будет использоваться для работы с картой. Самый простой способ сделать это – добавить элемент `<fragment>` в файл макета для операции `Activity`.

2. Реализовать интерфейс `OnMapReadyCallback` и использовать метод обратного вызова `onMapReady(GoogleMap)`, чтобы получить дескриптор объекта `GoogleMap`. Объект `GoogleMap` является внутренним представлением самой карты. Чтобы установить параметры представления для карты, следует изменить ее объект `GoogleMap`.

3. Вызвать `getMapAsync()` для фрагмента, чтобы зарегистрировать обратный вызов.

Чтобы устанавливать маркер на карту используется `OnMapLongClickListener`, который можно установить на карте путем вызова `GoogleMap.setOnMapLongClickListener(OnMapLongClickListener)`. Этот блок прослушивания действует подобно блоку, отслеживающему обычные нажатия, и получает уведомления о длительных нажатиях через обратный вызов `onMapLongClick(LatLng)`. В этом вызове прописан

код, который позволяет создать маркер на карте с координатами касания. Значения LatLng, содержащие координаты широты и долготы, запоминаются. Затем служба StreetView использует эти координаты для отображения панорамного снимка.

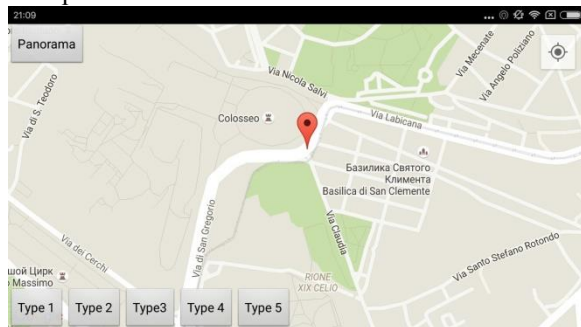


Рисунок 1. Скриншот карты с маркером

Служба Street View

Панорама Street View предоставляет 360-градусные панорамы, снятые с указанных дорог в пределах области покрытия. Каждая панорама Street View представляет собой изображение или набор изображений, обеспечивающие полный обзор из одной точки на 360 градусов. Изображения соответствуют равнопромежуточной проекции (проекции Плате-Карре), содержащей 360-градусный горизонтальный обзор (полный оборот) и 180-градусный вертикальный обзор (от направления строго вверх до направления строго вниз). Полученная 360-градусная панорама определяет проекцию на сфере с помощью переноса изображения на двухмерную поверхность этой сферы.

Чтобы добавление Street View в приложение нужно выполнить следующие шаги:

1. Добавить объект Fragment в операцию Activity, которая будет обрабатывать панораму Street View. Самый простой способ сделать это – добавить элемент<fragment> в файл макета для Activity.

2. Реализовать интерфейс OnStreetViewPanoramaReadyCallback и использовать метод обратного вызова onStreetViewPanoramaReady(StreetViewPanorama), чтобы получить дескриптор объекта StreetViewPanorama.

3. Вызвать getStreetViewPanoramaAsync() для фрагмента, чтобы зарегистрировать обратный вызов.

Для того чтобы организовать перемещение камеры используется вызов StreetViewPanorama.animateTo(). Анимация интерполирует изображение между текущими и новыми атрибутами камеры. Описание вызова представлено ниже.

```
StreetViewPanoramaCamera camera = new
StreetViewPanoramaCamera.Builder()
    .zoom(mSVP.getPanoramaCamera().zoom)
    .tilt(tiltGyro)
    .bearing(bearingGyro)
    .build();
mSVP.animateTo(camera, duration);
```

Камера имеет следующие параметры: bearing - направление камеры, указанное в градусах по часовой стрелке от истинного севера, и tilt - наклон вверх или вниз по оси Y.

Атрибуты tiltGyro (bearingGyro), получены в результате суммирования текущего значения tilt (bearing) камеры и значения гироскопа по соответствующей оси.

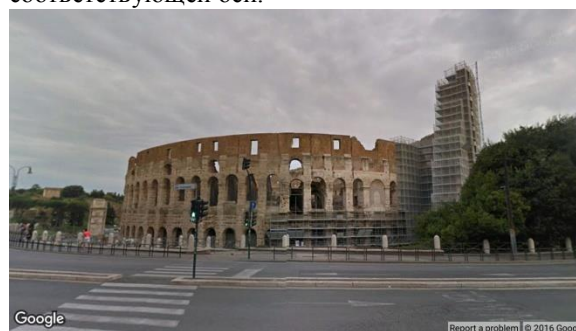


Рисунок 2. Скриншот панорамного снимка

Заключение

В работе представлен начальный этап создания приложения виртуальной реальности, в котором контентом является панорамные снимки улиц, полученные с Google карт. Использование датчиков смартфона (гироскопа и акселерометра) позволяет просматривать эти панорамные снимки, перемещая устройство в пространстве.

Список литературы

1. Обзор Google Maps Android API / URL: <https://developers.google.com/maps/documentation/android-api/intro?hl=ru> (дата обращения 06.06.2016).
2. Google VR SDK for Android / URL: <https://developers.google.com/vr/android/> (дата обращения 06.06.2016).
3. Виртуальная реальность и девайсы / URL: <http://oculus-rift.ru/virtual-reality-and-devices> (дата обращения 06.06.2016).
4. Как смартфоны чувствуют мир. Часть 1: акселерометры, гироскопы и другие сенсоры / URL: <http://www.ferra.ru/ru/techlife/review/mems-part-1/#.V1gndtmyOko> (дата обращения 06.06.2016).
5. Что такое гироскоп, акселерометр, G-сенсор / URL: <http://best-guide.ru/?p=805> (дата обращения 06.06.2016).