Sensors

android

Обзор сенсоров

Most Android-powered devices have built-in sensors that measure motion, orientation, and various environmental conditions. These sensors are capable of providing raw data with high precision and accuracy, and are useful if you want to monitor three-dimensional device movement or positioning, or you want to monitor changes in the ambient environment near a device. For example, a game might track readings from a device's gravity sensor to infer complex user gestures and motions, such as tilt, shake, rotation, or swing. Likewise, a weather application might use a device's temperature sensor and humidity sensor to calculate and report the dewpoint, or a travel application might use the geomagnetic field sensor and accelerometer to report a compass bearing.

https://developer.android.com/guide/topics/sensors/sensors_overview.html

Категории сенсоров

The Android platform supports three broad categories of sensors:

- **Motion sensors** - these sensors measure acceleration forces and rotational forces along three axes. This category includes accelerometers, gravity sensors, gyroscopes, and rotational vector sensors
- **Environmental sensors** - these sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers
- **Position sensors** - these sensors measure the physical position of a device. This category includes orientation sensors and magnetometers

Основные сенсоры

- Акселерометр
- Гироскоп
- Датчик освещения
- Датчик расстояния
- Датчик магнитных полей
- Барометр
- Датчик температуры окружающей среды
- Измеритель относительной влажности

На каждом устройстве может быть свой набор датчиков. В большинстве аппаратов есть акселерометр и гироскоп.

Сенсорный фреймворк

You can access sensors available on the device and acquire raw sensor data by using the Android sensor framework. The sensor framework provides several classes and interfaces that help you perform a wide variety of sensor-related tasks. For example, you can use the sensor framework to do the following:

- Determine which sensors are available on a device
- Determine an individual sensor's capabilities, such as its maximum range, manufacturer, power requirements, and resolution
- Acquire raw sensor data and define the minimum rate at which you acquire sensor data
- Register and unregister sensor event listeners that monitor sensor changes

Основные типы данных

- **SensorManager** - you can use this class to create an instance of the sensor service. This class provides various methods for accessing and listing sensors, registering and unregistering sensor event listeners, and acquiring orientation information. This class also provides several sensor constants that are used to report sensor accuracy and set data acquisition rates
- **Sensor** - you can use this class to create an instance of a specific sensor. This class provides various methods that let you determine a sensor's capabilities
- **SensorEvent** - the system uses this class to create a sensor event object, which provides information about a sensor event. A sensor event object includes the following information: the raw sensor data, the type of sensor that generated the event, the accuracy of the data, and the timestamp for the event
- **SensorEventListener** - you can use this interface to create two callback methods that receive notifications (sensor events) when sensor values change or when sensor accuracy changes

Sensor Manager

Класс **SensorManager** содержит несколько констант, которые характеризуют различные аспекты системы датчиков Android, в том числе:

- **Тип сенсора** (например - ориентация, акселерометр, свет, магнитное поле, расстояние до объектов, температура устройства и тд)
- **Частота измерений** - максимальная (для игр), обычная (для пользовательского интерфейса). Когда приложение запрашивает конкретное значение частоты отсчётов, с точки зрения сенсорной подсистемы это лишь рекомендация. Никакой гарантии, что измерения будут производиться с указанной частотой, нет
- **Точность** - высокая, средняя, низкая, ненадёжные данные

Частота измерений

Класс SensorManager содержит следующие константы для выбора подходящей частоты обновлений (в порядке убывания):

- **SensorManager.SENSOR_DELAY_FASTEST** — самая высокая возможная частота обновления показаний датчиков
- **SensorManager.SENSOR_DELAY_GAME** — частота, используемая для управления играми
- **SensorManager.SENSOR_DELAY_NORMAL** — частота обновлений по умолчанию
- **SensorManager.SENSOR_DELAY_UI** — частота для обновления пользовательского интерфейса

Точность сенсоров

Параметр **accuracy**, используемый в методах для представления степени точности датчика, использует одну из констант:

- **SensorManager.SENSOR_STATUS_ACCURACY_LOW**. Говорит о том, что данные, предоставляемые датчиком, имеют низкую точность и нуждаются в калибровке
- **SensorManager.SENSOR_STATUS_ACCURACY_MEDIUM**. Говорит о средней степени точности датчика и том, что калибровка может улучшить результат
- **SensorManager.SENSOR_STATUS_ACCURACY_HIGH**. Показатели датчика точны настолько, насколько это возможно
- **SensorManager.SENSOR_STATUS_UNRELIABLE**. Данные, предоставляемые датчиком, недостоверны. Это значит, что датчик необходимо откалибровать, иначе невозможно считывать результаты

Типы сенсоров

https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-intro (перейти по ссылке)

Sensor	Type	Description	Common Uses
TYPE_ACCELEROMETER	Hardware	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), including the force of gravity.	Motion detection (shake, tilt, etc.).
TYPE_AMBIENT_TEMPERATURE	Hardware	Measures the ambient room temperature in degrees Celsius (°C). See note below.	Monitoring air temperatures.
TYPE_GRAVITY	Software or Hardware	Measures the force of gravity in m/s^2 that is applied to a device on all three physical axes (x, y, z).	Motion detection (shake, tilt, etc.).
TYPE_GYROSCOPE	Hardware	Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y, and z).	Rotation detection (spin, turn, etc.).

Описание типов сенсоров

- **TYPE_ACCELEROMETER** - измеряет ускорение в пространстве по осям X, Y, Z
- **TYPE_AMBIENT_TEMPERATURE** - датчик для измерения температуры (с API 14) в градусах Цельсия, который заменил устаревший **TYPE_TEMPERATURE**
- **TYPE_GRAVITY** - трёхосевой датчик силы тяжести. Как правило, это виртуальный датчик и представляет собой низкочастотный фильтр для показаний, возвращаемых акселерометром
- **TYPE_GYROSCOPE** - трёхосевой гироскоп, возвращающий текущее положение устройства в пространстве в градусах по трём осям
- **TYPE_LIGHT** - измеряет степень освещенности. Датчик окружающей освещенности, который описывает внешнюю освещенность в люксах. Этот тип датчиков обычно используется для динамического изменения яркости экрана.
- **TYPE_LINEAR_ACCELERATION** - трёхосевой датчик линейного ускорения, возвращающий показатели ускорения без учёта силы тяжести. Это виртуальный датчик, использующий показания акселерометра
- **TYPE_MAGNETIC_FIELD** - датчик магнитного поля, определяющий текущие показатели магнитного поля в микротеслах по трём осям
- **TYPE_ORIENTATION** - датчик ориентации. Измеряет повороты, наклоны и вращение устройства
- **TYPE_PRESSURE** - датчик атмосферного давления (барометр), возвращающий текущее давление в миллибарах. Можно определять высоту над уровнем моря, путём сравнения атмосферного давления в двух точках. Также барометры могут применяться для прогнозирования погоды
- **TYPE_PROXIMITY** - датчик расстояния между устройством и целевым объектом в сантиметрах. Каким образом выбирается объект и какие расстояния поддерживаются, зависит от аппаратной реализации данного датчика, возможно возвращение двух значений - Близко и Далеко. Типичное его применение — обнаружение расстояния между устройством и ухом пользователя для автоматического регулирования яркости экрана или выполнения голосовой команды
- **TYPE_RELATIVE_HUMIDITY** - датчик относительной влажности в виде процентного значения (API 14)
- **TYPE_ROTATION_VECTOR** - Возвращает положение устройства в пространстве в виде угла относительно оси. Виртуальный датчик, берущий показания от акселерометра и гироскопа

Задачи при работе с сенсорами

In a typical application you use these sensor-related APIs to perform two basic tasks:

- **Identifying sensors and sensor capabilities -** Identifying sensors and sensor capabilities at runtime is useful if your application has features that rely on specific sensor types or capabilities. For example, you may want to identify all of the sensors that are present on a device and disable any application features that rely on sensors that are not present. Likewise, you may want to identify all of the sensors of a given type so you can choose the sensor implementation that has the optimum performance for your application.
- **Monitor sensor events -** Monitoring sensor events is how you acquire raw sensor data. A sensor event occurs every time a sensor detects a change in the parameters it is measuring. A sensor event provides you with four pieces of information: the name of the sensor that triggered the event, the timestamp for the event, the accuracy of the event, and the raw sensor data that triggered the event.

О чём нужно помнить

- Показания бывают очень неровными. Нужно будет использовать какое-то среднее значение показаний, но не переборщить, чтобы приложение оставалось отзывчивым
- Данные приходят неравномерно. Не нужно ждать спокойного, ровного потока данных
- Можно пытаться предугадать будущие действия пользователя. Например, если идут данные о начале вращения устройства, можно предугадать следующее движение и подготовиться к нему

Доступность сенсоров

While sensor availability varies from device to device, it can also vary between Android versions. This is because the Android sensors have been introduced over the course of several platform releases. For example, many sensors were introduced in Android 1.5 (API Level 3), but some were not implemented and were not available for use until Android 2.3 (API Level 9). Two sensors have been deprecated and replaced by newer, better sensors.

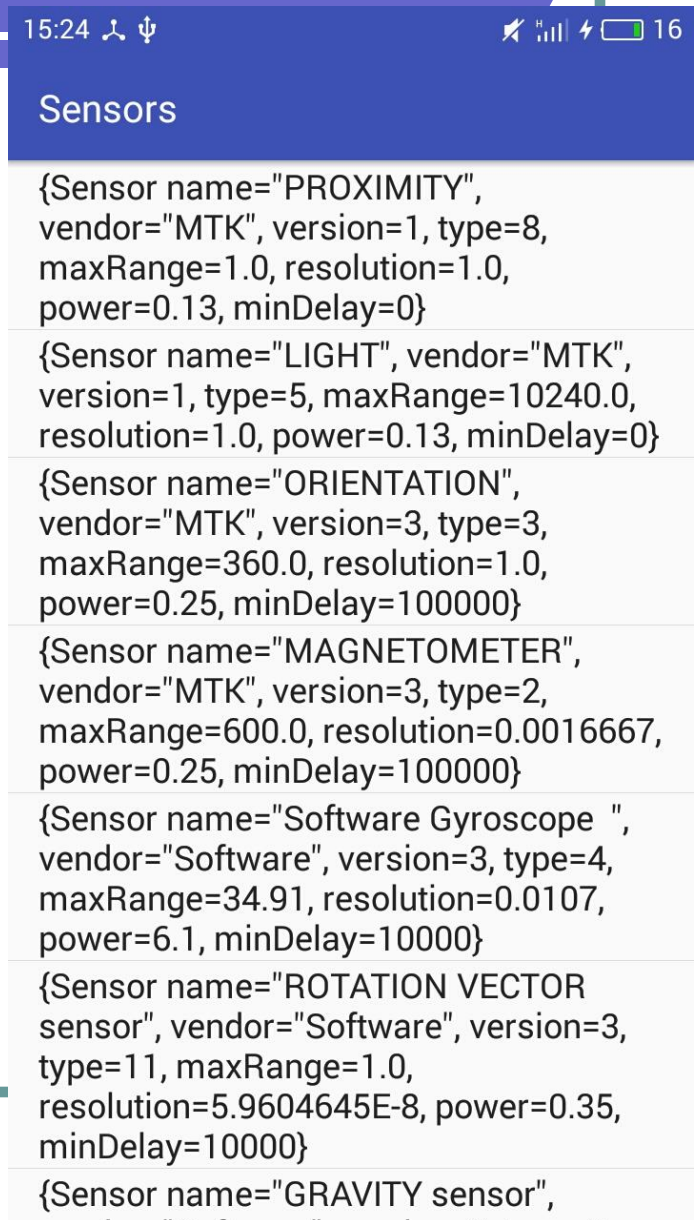
Table on the next page of this presentation summarizes the availability of each sensor on a platform-by-platform basis. Only four platforms are listed because those are the platforms that involved sensor changes. Sensors that are listed as deprecated are still available on subsequent platforms (provided the sensor is present on a device), which is in line with Android's forward compatibility policy.

Доступность сенсоров

Sensor	Android 4.0 (API Level 14)	Android 2.3 (API Level 9)	Android 2.2 (API Level 8)	Android 1.5 (API Level 3)
TYPE_ACCELEROMETER	Yes	Yes	Yes	Yes
TYPE_AMBIENT_TEMPERATURE	Yes	n/a	n/a	n/a
TYPE_GRAVITY	Yes	Yes	n/a	n/a
TYPE_GYROSCOPE	Yes	Yes	n/a[1]	n/a[1]
TYPE_LIGHT	Yes	Yes	Yes	Yes
TYPE_LINEAR_ACCELERATION	Yes	Yes	n/a	n/a
TYPE_MAGNETIC_FIELD	Yes	Yes	Yes	Yes
TYPE_ORIENTATION	Yes[2]	Yes[2]	Yes[2]	Yes
TYPE_PRESSURE	Yes	Yes	n/a[1]	n/a[1]
TYPE_PROXIMITY	Yes	Yes	Yes	Yes
TYPE_RELATIVE_HUMIDITY	Yes	n/a	n/a	n/a
TYPE_ROTATION_VECTOR	Yes	Yes	n/a	n/a
TYPE_TEMPERATURE	Yes[2]	Yes	Yes	Yes

Определение списка сенсоров

https://
git.io/v1KdA



https://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-identify

Типы сенсоров из Sensor.class

```
public static final int TYPE_ACCELEROMETER = 1;
public static final int TYPE_ALL = -1;
public static final int TYPE_AMBIENT_TEMPERATURE = 13;
public static final int TYPE_DEVICE_PRIVATE_BASE = 65536;
public static final int TYPE_GAME_ROTATION_VECTOR = 15;
public static final int TYPE_GEOMAGNETIC_ROTATION_VECTOR = 20;
public static final int TYPE_GRAVITY = 9;
public static final int TYPE_GYROSCOPE = 4;
public static final int TYPE_GYROSCOPE_UNCALIBRATED = 16;
public static final int TYPE_HEART_BEAT = 31;
public static final int TYPE_HEART_RATE = 21;
public static final int TYPE_LIGHT = 5;
public static final int TYPE_LINEAR_ACCELERATION = 10;
public static final int TYPE_MAGNETIC_FIELD = 2;
public static final int TYPE_MAGNETIC_FIELD_UNCALIBRATED = 14;
public static final int TYPE_MOTION_DETECT = 30;
/** @deprecated */
@Deprecated
public static final int TYPE_ORIENTATION = 3;
public static final int TYPE_POSE_6DOF = 28;
```

Приложение-детектор сенсоров

https://play.google.com/store/apps/details?id=imoblife.androidsensorbox

Проверка наличия сенсора

You can also determine whether a specific type of sensor exists on a device by using the getDefaultSensor() method and passing in the type constant for a specific sensor. If a device has more than one sensor of a given type, one of the sensors must be designated as the default sensor. If a default sensor does not exist for a given type of sensor, the method call returns null, which means the device does not have that type of sensor. For example, the following code checks whether there's a magnetometer on a device:

```
private SensorManager mSensorManager;
...
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
if (mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null){
  // Success! There's a magnetometer.
  }
else {
  // Failure! No magnetometer.
  }
```

Производитель и версия

In addition to listing the sensors that are on a device, you can use the public methods of the SensorIn addition to listing the sensors that are on a device, you can use the public methods of the Sensor class to determine the capabilities and attributes of individual sensors. For example, you can use the getResolution()In addition to listing the sensors that are on a device, you can use the public methods of the Sensor class to determine the capabilities and attributes of individual sensors. For example, you can use

```
if (mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY) != null){
  List<Sensor> gravSensors = mSensorManager.getSensorList(Sensor.TYPE_GRAVITY);
  for(int i=0; i<gravSensors.size(); i++) {
    if ((gravSensors.get(i).getVendor().contains("Google Inc.")) &&
        (gravSensors.get(i).getVersion() == 3)){
      // Use the version 3 gravity sensor.
      mSensor = gravSensors.get(i);
    }
  }
}
```

Отслеживание показаний

Понадобится интерфейс **android.hardware.SensorListener**. Интерфейс реализован с помощью класса, который используется для ввода значений датчиков по мере их изменения в режиме реального времени. Интерфейс включает в себя два необходимых метода:

- **onSensorChanged(int sensor, float values[])** вызывается, когда изменяется значение датчика. В число аргументов метода входит целое, которое указывает, значение какого датчика изменилось, и массив значений с плавающей запятой, отражающих собственно значения датчика. Некоторые датчики выдают только одно значение данных, тогда как другие предоставляют три значения с плавающей запятой
- **onAccuracyChanged(int sensor, int accuracy)** вызывается при изменении точности показаний датчика. Аргументами служат два целых числа: одно указывает датчик, а другое соответствует новому значению точности этого датчика

Значения сенсоров

Тип датчика	Количество значений	Содержание значений	Примечание
TYPE_ACCELEROMETER	3	value[0]:ось X (поперечная) value[1]: ось Y (продольная) value[2]:ось Y (вертикальная)	Ускорение (м/с^2) по трём осям. Константы SensorManager.GRAVITY_*
TYPE_GRAVITY	3	value[0]:ось X (поперечная) value[1]: ось Y (продольная) value[2]:ось Y (вертикальная)	Сила тяжести (м/с^2) по трём осям. Константы SensorManager.GRAVITY_*
TYPE_RELATIVE_HUMIDITY	1	value[0]:относительная влажность	Относительная влажность в процентах (%)
TYPE_LINEAR_ACCELERATION	3	value[0]:ось X (поперечная) value[1]: ось Y (продольная) value[2]:ось Y (вертикальная)	Линейное ускорение (м/с^2) по трём осям без учёта силы тяжести

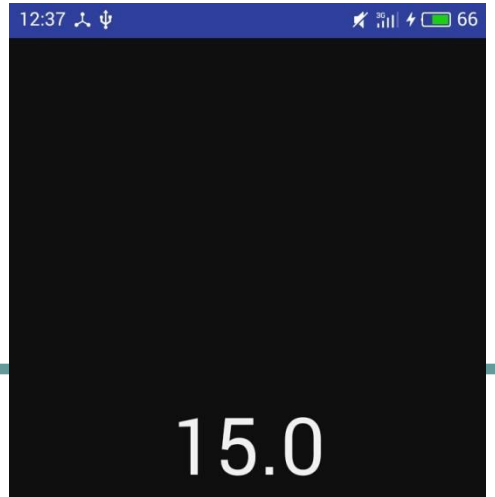
http://developer.alexanderklimov.ru/android/sensors.php

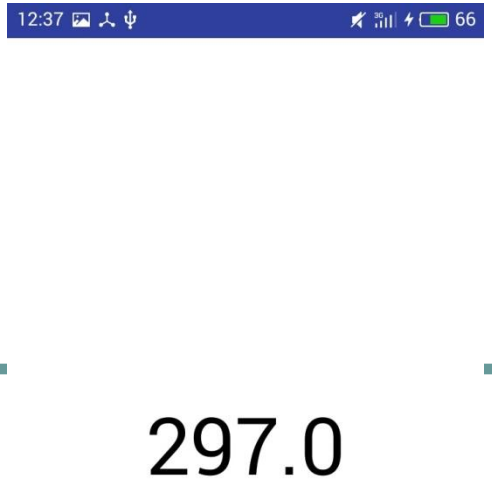
Сенсор освещения, яркость

Датчик **TYPE_LIGHT** измеряет степень освещённости в люксах. Люкс — это единица освещённости поверхности 1м² при световом потоке падающего на неё излучения, равном 1 лм (люмен). Этот тип датчиков обычно используется для динамического изменения яркости экрана. В манифесте желательно проставить требование к устройству:

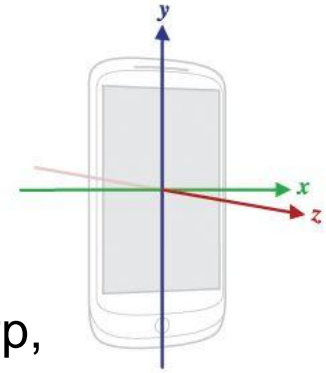
<uses-feature android:name="android.hardware.sensor.light"/>

https://git.io/v1PrI





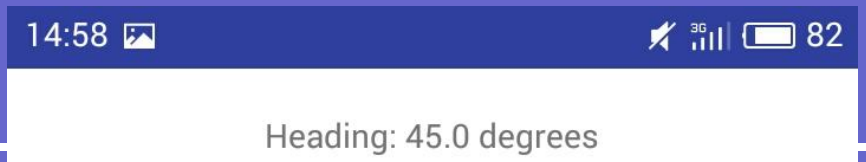
Акселерометр



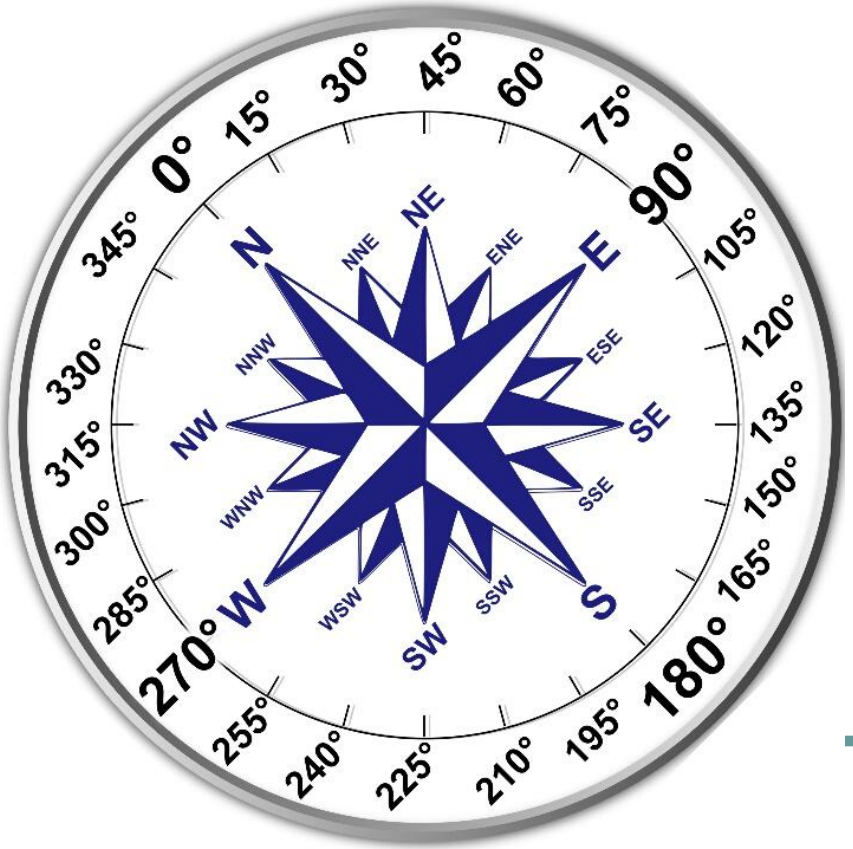
Практически любой современный телефон имеет акселерометр, позволяющий определить положение телефона относительно земли, а также ускорение в пространстве по осям X, Y, Z. Акселерометры часто выступают в качестве датчиков силы притяжения, так как они не могут определить, чем вызвано ускорение — движением или гравитацией. В результате этого в состоянии покоя акселерометр будет указывать на ускорение по оси Z (вверх/вниз), равное 9,8м/с^2. Ускорение — это производная скорости по времени, поэтому акселерометр определяет, насколько быстро изменяется скорость устройства в заданном направлении. Используя этот датчик, можно обнаруживать движение и, что более полезно, изменение его скорости. Акселерометр не измеряет скорость как таковую, поэтому нельзя получить скорость движения, основываясь на единичном замере. Вместо этого необходимо учитывать изменения ускорения на протяжении какого-то отрезка времени.

https://git.io/v1PHP

Компас



https://git.io/v1PdA



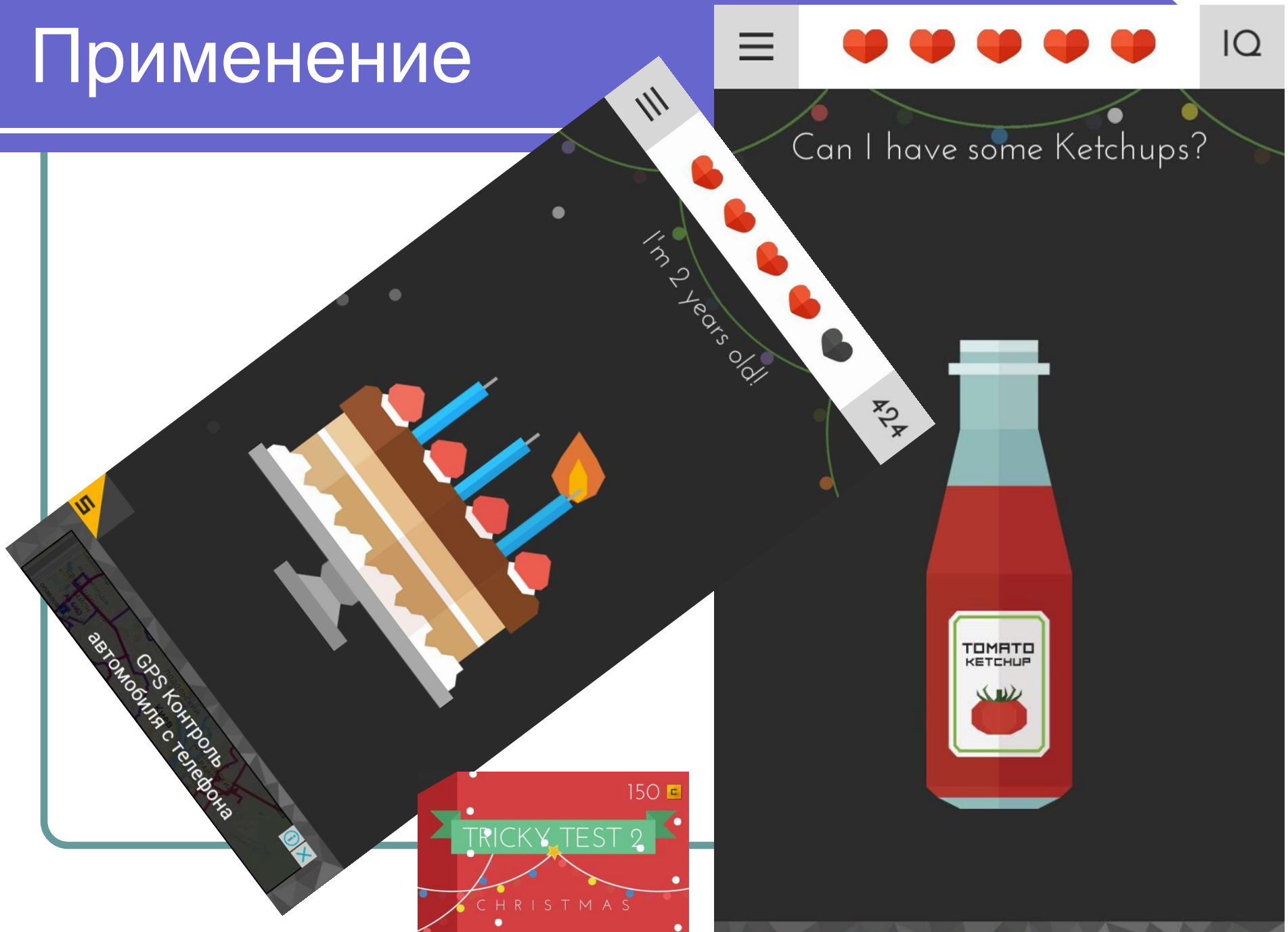
Гироскоп

Гироскоп – это устройство, способное реагировать на изменение углов ориентации тела, на котором оно установлено, относительно инерциальной системы отсчета. Простейший пример гироскопа — юла (волчок). Термин впервые введён Ж. Фуко в своём докладе в 1852 году во Французской Академии Наук. Доклад был посвящён способам экспериментального обнаружения вращения Земли в инерциальном пространстве

https://git.io/v1MOD

https://en.wikipedia.org/wiki/Gyroscope

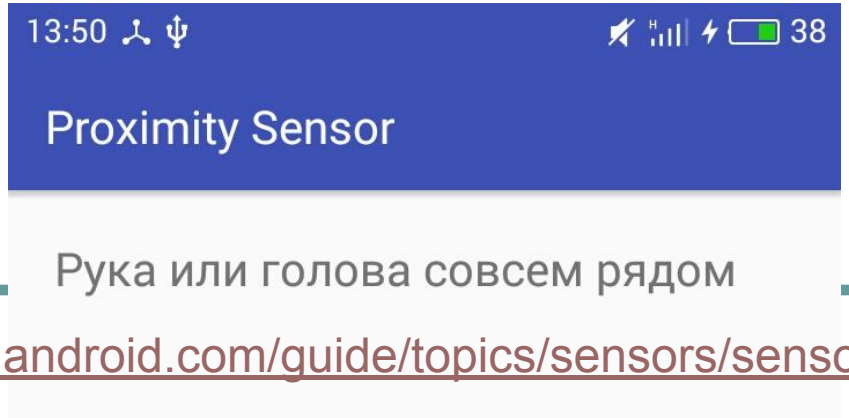
Применение


Can I have some Ketchups?
I'm 2 years old!
424
TOMATO KETCHUP
GPS Контроль автомобиля с телефона
150
TRICKY TEST 2
CHRISTMAS

Датчик расстояния

Датчик измеряет удалённость объекта в сантиметрах. Используется в основном для определения телефона у лица пользователя. Когда пользователь подносит телефон к уху, то экран автоматически выключается для экономии заряда батареи. Требование к устройству **<uses-feature android:name="android.hardware.sensor.proximity" android:required="true" />**

https://git.io/v1MBL



https://developer.android.com/guide/topics/sensors/sensors_position.html

Статьи про сенсоры

- https://developer.android.com/guide/topics/sensors/sensors_overview.html
- https://developer.android.com/guide/topics/sensors/sensors_motion.html
- https://source.android.com/devices/sensors/sensor-types.html
- https://habrahabr.ru/post/137678/
- http://www.stevesandroidguide.com/android-sensors/
- http://android.stackexchange.com/questions/29204/which-hardware-sensors-are-supported-by-android
- http://www.ssaurel.com/blog/list-all-sensors-available-on-an-android-device/
- http://startandroid.ru/ru/uroki/vse-uroki-spiskom/287-urok-137-sensory-uskorenie-orientatsija.html
- http://developer.alexanderklimov.ru/android/sensors.php

Практика (кошачья палка)

Написать приложение, которое однозначно реагирует на встряхивание телефона определённым образом, и показывает случайное предсказание (вроде «вжух, и ты набрал 10 килограмм»)

Практика (зельеварение)

Написать приложение «Зельеварение», которое предлагает пользователю сварить зелье по определённому рецепту (например, зелье «Нектар Афродиты» – в состав зелья входят соцветия вербены - 13 штук, красное вино - 1 литр, цедра лимонная - 30г, рубин порошковый - 12г, сушёная шкурка ящерицы - 1 шт. Для приготовления этого зелья потребуется котёл и очаг. Прежде всего нальём в котёл вино и доведём на огне до слабого кипения (минут 8-10). Как только на поверхности появятся первые пузырьки, положить в котел цветки вербены, строго по одному, и цедру. Поварить ещё 3 минуты, затем рассеять в ёмкость порошок рубина. После ёще 5 минут кипения снять котёл с огня, позвать какую-нибудь девушку (незамужнюю, но желательно невесту) и попросить её окунуть в котел сушёную шкурку ящерицы со словами «афродис кумалус верлютум". После охлаждения зелье готово к употреблению. В качестве приворотного зелья оно широко применяется и сейчас, это один из обязательных рецептов, которые описаны в книге «100 популярных магических зелий Дрейка». Все действия по добавлению ингредиентов должны сопровождаться определёнными движениями устройства в пространстве ☺