

Лекция № 5
Разработка приложений для
планшетных компьютеров

План

- ▶ Планшетные компьютеры
- ▶ Сенсорный экран
- ▶ Датчики
- ▶ Сенсорные возможности Android
- ▶ Интерфейс традиционных датчиков в Android API
- ▶ Датчик акселерации
- ▶ Датчик ориентации
- ▶ Датчик GPS



Что такое планшет?

- ▶ Нет клавиатуры
- ▶ Большие размеры
- ▶ В первую очередь для развлечений



Аппаратные особенности планшета

- ▶ Устройство ввода – сенсорный экран
- ▶ Набор датчиков
- ▶ Также существуют особенности пользовательских интерфейсов!

Сенсорный экран

- ▶ Сенсорный экран – универсальное устройство ввода
- ▶ Почти полный аналог “мыши”
- ▶ Ввод текста – с помощью специальных программных решений, например, экранной клавиатуры

Способы использования сенсорного экрана

- ▶ Использовать стандартные виджеты, не задумываясь о сенсорном экране
- ▶ Перехват событий Gesture
- ▶ **Перехват событий сенсорного экрана**
- ▶ Взаимодействие на уровне драйвера в ядре ОС



Как использовать события

- ▶ Обычно достаточно первой точки-прикосновения
- ▶ Можно реагировать только на TouchBegin или TouchEnd
- ▶ Можно обрабатывать событие при каждом TouchUpdate
- ▶ Можно накапливать точки при TouchUpdate и обработать при TouchEnd.

Датчики на планшетах

- ▶ Датчики измеряют разнородные параметры — положение, скорость, освещенность и т.д.
- ▶ Набор датчик различен на разных устройствах
- ▶ В будущем могут появиться новые типы датчиков
- ▶ Датчики часто работают в реальном времени



Получение данных от датчика

Поллинг

- ▶ Периодический опрос оборудования
- ▶ Простота реализации
- ▶ Низкая вычислительная эффективность

Прерывания

- ▶ Чтение по запросу от оборудования
- ▶ Высокая вычислительная эффективность
- ▶ Сложность реализации
- ▶ Использование механизма слотов и сигналов Qt упрощает реализацию!



Сенсорные возможности Android

- ▶ Унифицированный программный интерфейс для с различных аппаратных датчиков
- ▶ Набор многочисленных сенсоров для получения данных

Интерфейс традиционных датчиков в Android API

- ▶ Пакет *android.hardware* — мощная библиотека, может быть использована в приложениях, опирающихся на аппаратные возможности устройства
- ▶ Прежде чем использовать датчик, необходимо убедиться в его наличии на устройстве
- ▶ Пакет *android.os.** содержит несколько полезных классов для взаимодействия с операционной средой



Интерфейс традиционных датчиков в Android API

- ▶ *SensorManager* — менеджер управления различными датчиками устройства
- ▶ Класс *SensorManager* позволяет получить доступ к внутренним датчикам платформы Android
- ▶ Вызов `getSystemService(SensorManager.SENSOR_SERVICE)` инициализирует менеджер
- ▶ Метод `getDefaultSensor(<тип сенсора>)` класса *SensorManager* позволяет инициализировать конкретный датчик для дальнейшего его использования

Интерфейс традиционных датчиков в Android API

- ▶ *SensorEventListener* используется для получения уведомлений от *SensorManager*
- ▶ *Public abstract void onAccuracyChanged(Sensor, accuracy)* вызывается при смене точности показаний датчика
- ▶ *Public abstract void onSensorChanged(SensorEvent, values[])* вызывается при смене показаний датчика



Интерфейс традиционных датчиков в Android API

- ▶ *SensorEvent* — класс, описывающий различные события для всевозможных датчиков устройства
- ▶ Точность показаний
- ▶ Объект сгенерировавший событие
- ▶ Время возникновения события
- ▶ Значения с датчика(показания)



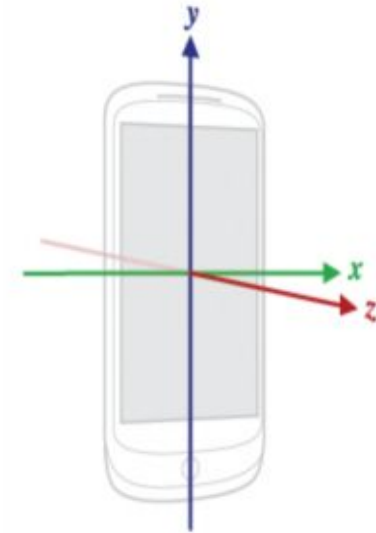
Датчик акселерации

- ▶ Датчик акселерации позволяет определить ускорение устройства в пространстве относительно трех измерений
- ▶ Ускорение определяется с учетом ускорения свободного падения
- ▶ Вызов метода `getDefaultSensor(Sensor.TYPE_ACCELEROMETER)` инициализирует датчик ускорения



Датчик ориентации

- ▶ Датчик ориентации используется для распознавания положения устройства в пространстве
- ▶ Ориентация в Android определяется тремя величинами: азимут, высота, вращение
- ▶ Вызов метода `getDefaultSensor(Sensor.TYPE_ORIENTATION)` инициализирует датчик ориентации



Датчик GPS

- ▶ Датчик GPS не является стандартным датчиком
- ▶ GPS служба предоставляет программисту координаты устройства (долгота, широта, высота над уровнем моря) для реализации логики приложения
- ▶ Для использования GPS датчика, необходим объект класса *LocationManager*
- ▶ Объект класса можно инициализировать посредством вызова метода *getSystemService* с параметром *LOCATION_SERVICE*



САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



Благодарю за внимание!

Вопросы?