Лаборатория Системного ПРограммирования и ИНформационных Технологий СПбГУ

Лабораторная работа № 3
Использование датчика ориентации для управления пользовательским интерфейсом

Цель лабораторной работы

Демонстрация процесса разработки практического приложения.

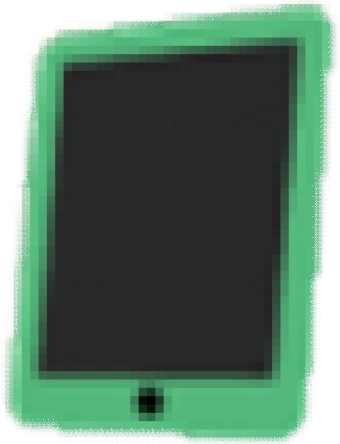
www.math.spbu.ru/user/gran/Atom21/lab03

Необходимые навыки и инструменты

- Базовые знания языка C++
- Знание основ фреймворка Qt. Желательно знакомство с библиотеками Qt Mobility и MeeGoTouch
- Представление о процессе кросс-разработки приложений
- **Устройство с датчиком ориентации — планшет или смартфон**

Постановка задачи

- Получить список датчиков на устройстве
- Подписаться на получение данных от датчика ориентации
- Обновить интерфейс пользователя согласно полученным данным



Доступ к датчикам

- Компонента sensors библиотеки Qt Mobility
- Подключаем в файле проекта:
- CONFIG += mobility
- MOBILITY = sensors
- Добавляем макрос QTM_USE_NAMESPACE
- Основной класс QSensor
- Датчики измеряют очень разнородные параметры — положение, скорость, освещенность и т.д.

Перечень датчиков

- QSensor::sensorTypes()
- список типов датчиков
- QSensor::sensorsForType()
- список идентификаторов датчиков данного типа
- Типы и идентификаторы — строки в форме QByteArray

Датчик QOrientationSensor

Класс данных QOrientationReading

Возможные значения:

- QOrientationReading::Undefined
- QOrientationReading::TopUp
- QOrientationReading::TopDown
- QOrientationReading::LeftUp
- QOrientationReading::RightUp
- QOrientationReading::FaceUp
- QorientationReading::FaceDown

Последние два значения мой планшет определять не умеет

Создание датчика

```
QSensor sensor(type);
// или QOrientationSensor sensor;
sensor.setIdentifier(identifier);

// проверка на работоспособность
if (!sensor.connectToBackend()) {
        // fail...
}
// Можно создать много QSensor
// для одного датчика!
```

Способы чтения данных

- **Поллинг**
  - Периодический опрос оборудования
  - Простота реализаци
  - Низкая вычислительная эффективность
- **Прерывания**
  - Чтение по запросу от оборудования
  - Высокая вычислительная эффективность
  - Сложность реализации
  - Использование механизма слотов и сигналов Qt упрощает реализацию!

Подписка на датчик

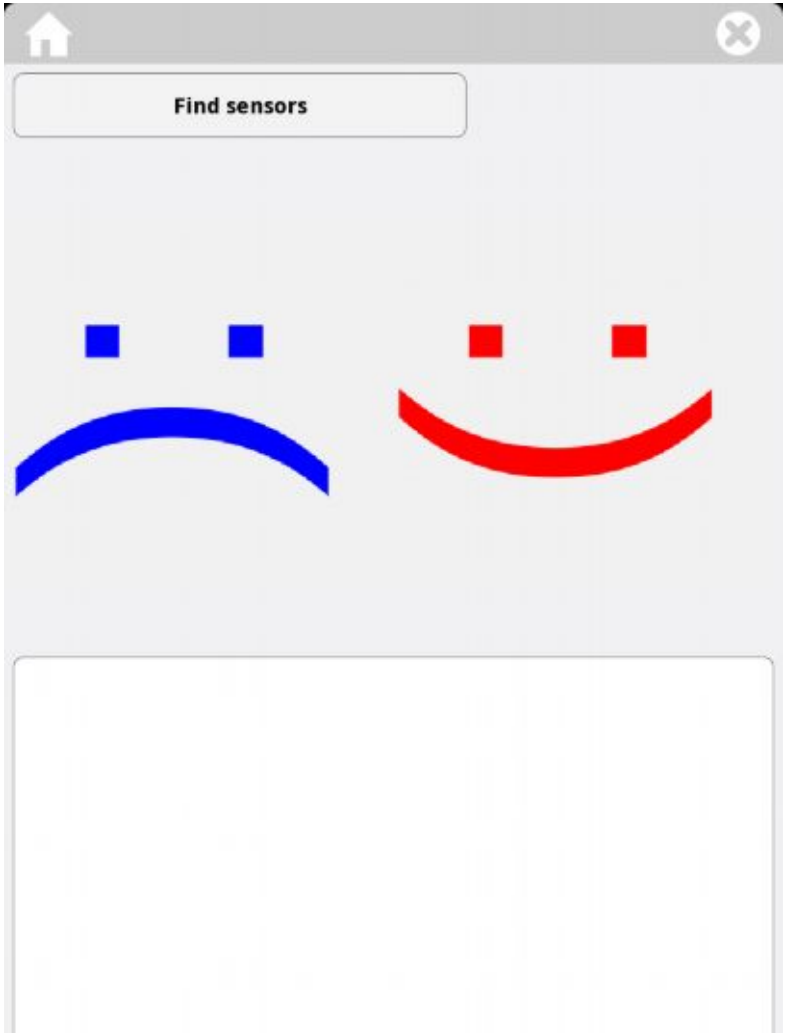
```
// Сигнал readingChanged() испускается
// когда новые данные готовы для чтения
QObject::connect(mySensor,SIGNAL(readingChanged()),
                                this,SLOT(readSensor()));
// Запускаем датчик
// (физически датчик мог работать и раньше!)
mySensor->start();
```

Чтение датчика QOrientationSensor

```
// Слот, связанный с readingChanged()
void MyApp::readSensor() {
        QOrientationReading *read = this->mySensor-
>reading();
        int orientation = read->orientation();
        // полученную ориентацию можно использовать
для
        // изменения внешнего вида приложения
}
```

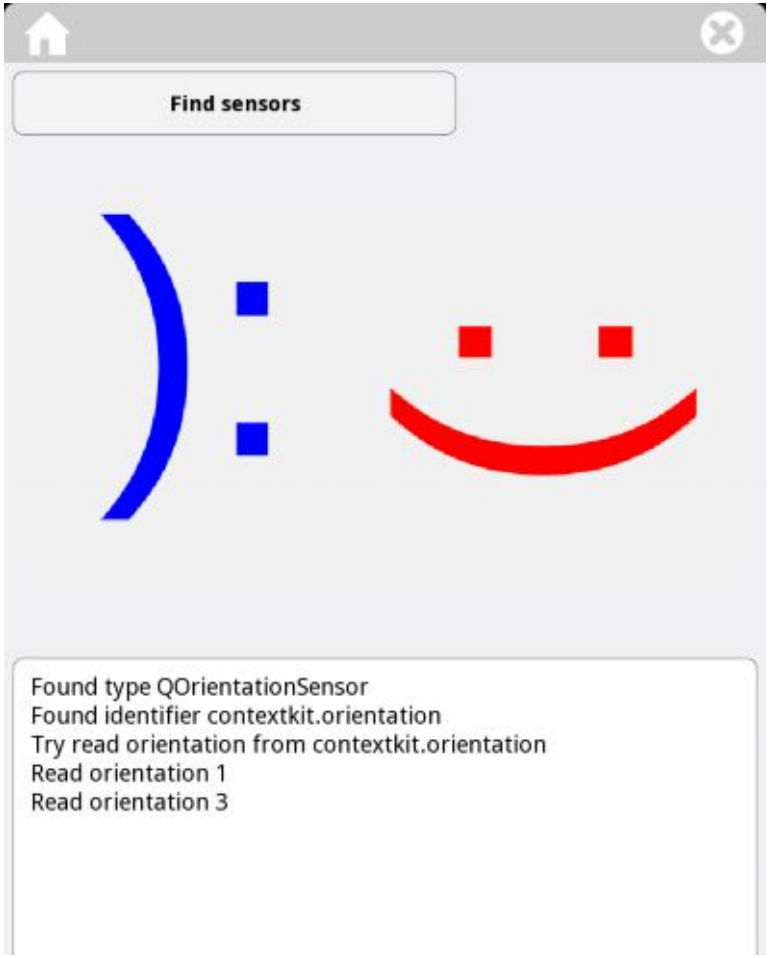
Библиотека MeeGoTouch: GUI

- GUI MeeGoTouch имеет встроенную поддержку датчика ориентации
- Красный смайл ориентируется с помощью MeeGoTouch (по умолчанию)
- Синий смайл ориентируется непосредственно с помощью датчика



Библиотека MeeGoTouch: GUI

- После активации датчика синий смайл сохраняет ориентацию относительно корпуса планшета



Вращение смайла

```
int angle = 90;
switch (orient) {
case QOrientationReading::Undefined:
case QOrientationReading::TopUp:
default:
    // do nothing
    break;
case QOrientationReading::LeftUp:
    angle += 90;
    break;
case QOrientationReading::TopDown:
    angle += 180;
    break;
case QOrientationReading::RightUp:
    angle += 270;
    break;
}
```

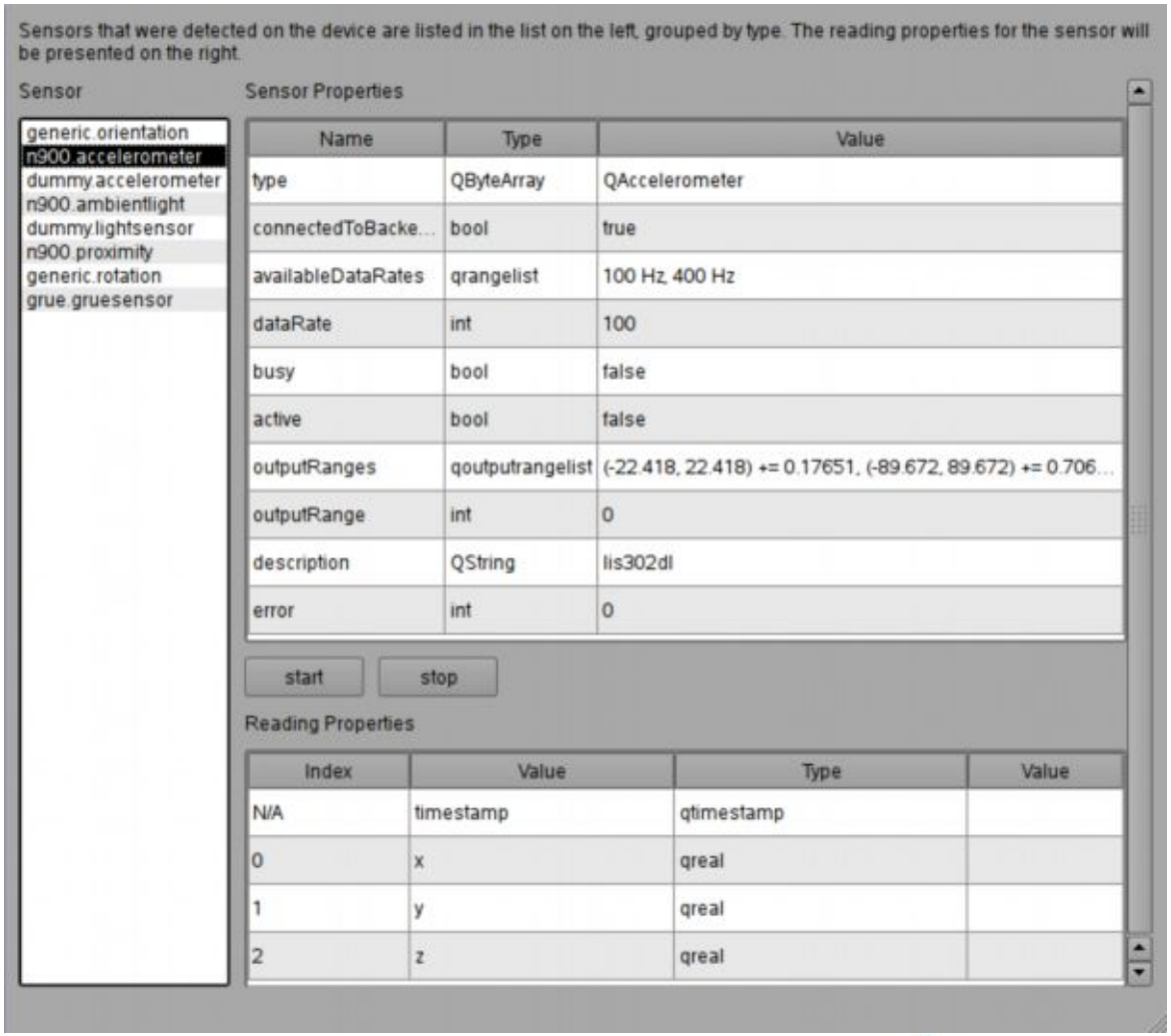
```
QImage image(w,h,QImage::Format_RGB888);
QPainter* painter = new QPainter(&image);
QSize size = image.size();

// Задаем преобразование
painter->translate(size.height()/2,
        size.height()/2);
painter->rotate(angle);
painter->translate(-size.height()/2,
        -size.height()/2);

painter->drawText(image.rect(),
        Qt::AlignCenter, text);
```

/usr/lib/qtmobility/examples/sensor_explorer

- Список всех датчиков
- Монитор состояния
- Доступен в пакете qt-mobility-examples



Домашние задания

1. Создать приложение для работы с двумя датчиками одновременно
2. Создать Qt-приложение без использования MeeGoTouch, так чтобы его пользовательский интерфейс реагировал на датчик ориентации

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Благодарю за внимание!

Вопросы?