

Лабораторная работа № 5
Обеспечение положительного User
Experience/Usability в сложных
пользовательских интерфейсах

Цель лабораторной работы

Демонстрация процесса разработки практического приложения.
Отображение элементов управления пользовательского интерфейса программы.

www.math.spbu.ru/user/gran/Atom21/lab05

User Experience

- ▶ User Experience (опыт взаимодействия, UX) – ощущения и реакция человека, вследствие использования или предполагаемого использования продукта, системы или услуги [ISO 9241-210]
- ▶ UX – ответственность не только программиста, но и маркетолога, специалиста по usability и др.
- ▶ С точки зрения программиста положительный UX достигается в том числе созданием оптимального пользовательского интерфейса



MeeGoTouch

- ▶ MeeGoTouch – надстройка над библиотекой Qt
- ▶ Пользовательский интерфейс Qt работоспособен в MeeGo, но кажется неадаптированным
- ▶ Пользовательский интерфейс MeeGoTouch работоспособен в MeeGo и выглядит “как родной”



Необходимые навыки и инструменты

- ▶ Базовые знания языка C++
- ▶ Знание основ фреймворка Qt.
Приветствуется начальное знакомство с библиотекой MeeGoTouch



Страницы

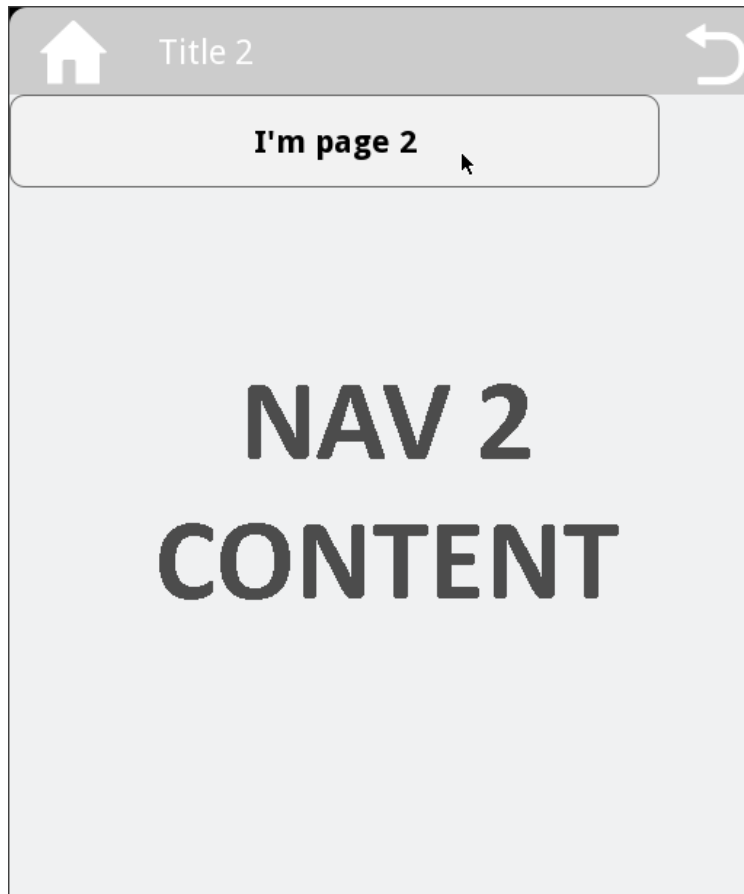
- ▶ Аналог закладок, позволяют разместить много элементов управления на небольшом экране
- ▶ Класс `MApplicationPage`, встроенная поддержка
- ▶ Альтернатива – прокрутка больших элементов управления



Навигация между страницами

- ▶ Несколько страниц не могут быть доступны одновременно
- ▶ Навигацию между страницами программист реализует самостоятельно:
- ▶ *page2.appear(&window);*
- ▶ MeeGoTouch отслеживает очередь открытых страниц и предоставляет кнопку "Back"

Примеры навигации



- ▶ Также возможна навигация через меню



Списки

- ▶ Реализованы классом MList
- ▶ Только одна отображаемая колонка
- ▶ Требуется создание классов для модели данных и визуализатора ячеек (в Android достаточно массива строк)

Модель данных

- ▶ Подкласс QAbstractListModel
- ▶ QVariant ListModel::data(const QModelIndex &index, int role) const
- ▶ {
- ▶ if (role == Qt::DisplayRole) {
- ▶ QStringList rowData;
- ▶ rowData << QString("Row %1").arg(index.row());
- ▶ rowData << "some details";
- ▶ return QVariant(rowData);
- ▶ }
- ▶ return QVariant();
- ▶ }

Визуализатор ячеек

- ▶ Подкласс MAbstractCellCreator<MContentItem>
- ▶ MContentItem – виджет, отображающий иконку и две строки текста
- ▶ void CellCreator::updateCell(const QModelIndex& index, QWidget *cell) const
- ▶ {
- ▶ MContentItem *contentItem = qobject_cast<MContentItem *>(cell);
- ▶ QVariant data = index.data(Qt::DisplayRole);
- ▶ QStringList rowData = data.value<QStringList>();
- ▶ contentItem->setTitle(rowData[0]);
- ▶ contentItem->setSubtitle(rowData[1]);
- ▶ contentItem->setImage(makeImage(50,50,rowData[0],8,Qt::red));
- ▶ }

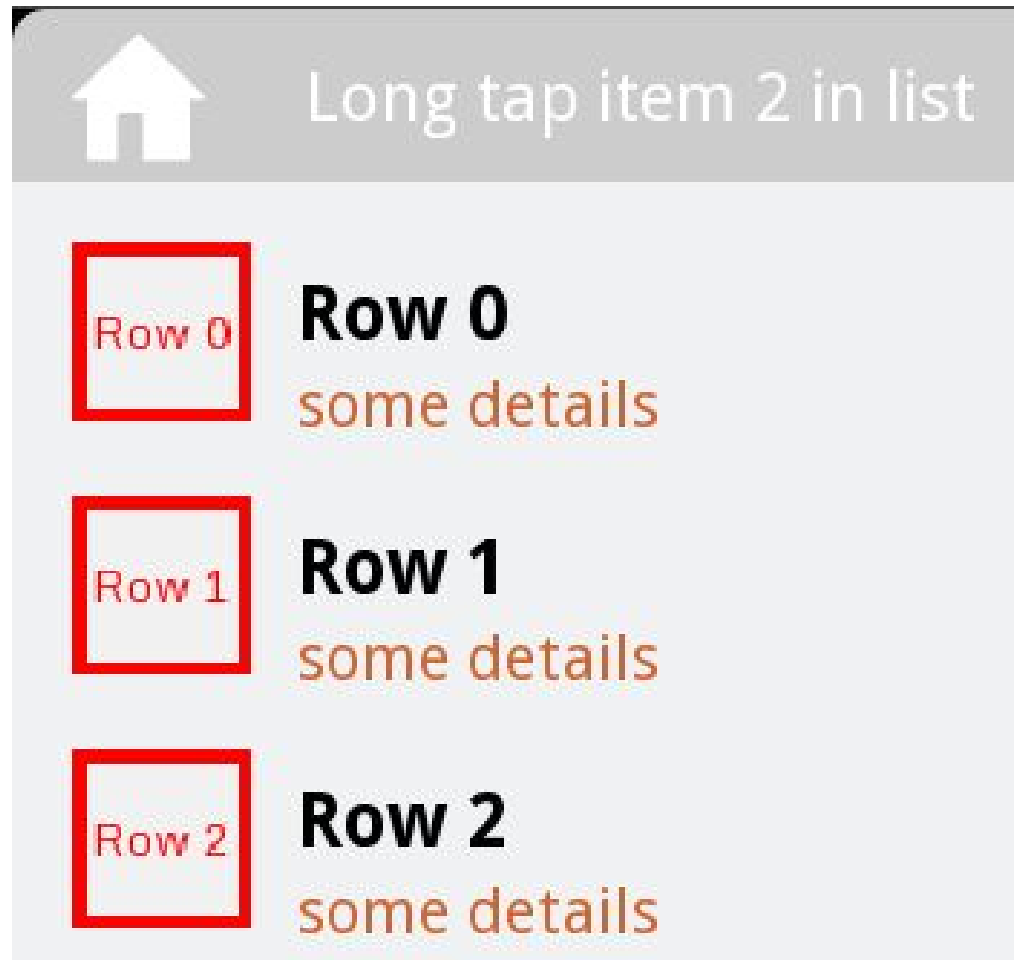


Обработка нажатий

- ▶ Сигнал для коротких кликов:
- ▶ *void MList::itemClicked(const QModelIndex& index)*
- ▶ Сигнал для долгих кликов:
- ▶ *void MList::itemLongTapped(const QModelIndex& index)*
- ▶ Можно обрабатывать сигналы MContentItem
- ▶ Можно выбирать элементы списка, в том числе несколько одновременно



Пример списка



Домашние задания

- ▶ 1. Создать приложение, в котором переключение страниц осуществляется с помощью меню приложения.
- ▶ 2. Усовершенствовать приведенный пример списка так, чтобы QVariant включал в себя не только две строки, но и QImage.
- ▶ 3. Создать приложение, в котором пользователь может выбрать несколько значений в списке и затем как-то их использовать.

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



Благодарю за внимание!

Вопросы?