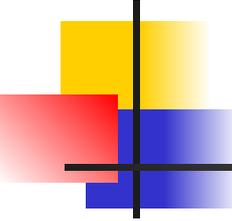




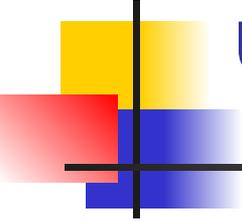
Программирование под Maemo

Маличенко Дмитрий
СПБ ГУАП



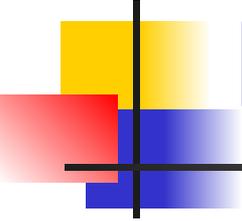
План

- Что такое Маето
- Программирование под Маето
- Работа в Scratchbox-е
- Простой пример
- Рисование в окне
- Работа с диалоговым окном
- Что еще не рассмотрено
- Ссылки



Что такое Maemo

- Maemo – это платформа для мобильных устройств основанная на Debian (www.maemo.org)
- Устройства на базе Maemo:
 - N900
 - N810
 - N800
 - N770



Какие есть версии

- Предыдущая – 4 (Diablo)
 - Библиотека GUI: GTK
 - Интерфейс: стилус
- Текущая – 5 (Fremantle)
 - Библиотека GUI: GTK
 - Интерфейс: палец + стилус
- Будущая версия – 6 (Harmattan)
 - Библиотека GUI: Qt
 - Интерфейс: только палец

Целевое устройство

N900

- Умеет звонить
- Линукс внутри
- Многозадачность
- WiFi, bluetooth
- GPS
- Сенсоры
- Интерфейс: палец + стилус
- Клавиатура: настоящая + виртуальная





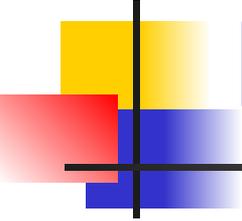
Программирование под Maemo: особенности

- Мобильное устройство
 - Маленький экран
 - Интерфейс: палец/стилус
 - Сенсоры
- Как облагородить интерфейс?
 - Библиотека Hildon – надстройка над GTK+
 - Использование меньшего числа элементов управления



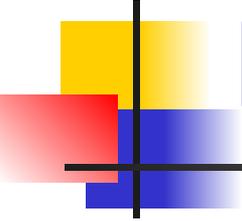
Программирование под Maemo: инструменты

- Требуется
 - Основанный на Debian дистрибутив Linux
 - Scratchbox – набор инструментов для кросс компиляции (поддерживает ARM, X86)
 - Maemo SDK
- Инструкция по установке
 - <http://maemo.org/development/>



Программирование под Maemo: связь с устройством

- Копирование файлов по USB на встроенную флешку (/home/user/MyDocs)
- USB Networking
 - есть вспомогательные утилиты для запуска на устройстве
- WiFi + ssh/scp



Работа в scratchbox - 1

- Используются два окружения
 - X86 для разработки, тестирования и отладки
 - ARM для компиляции под устройство
- Представляет собой “песочницу”

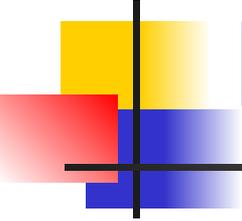
```
$ /scratchbox/login
```

```
Welcome to Scratchbox, the cross-compilation toolkit!
```

```
Use 'sb-menu' to change your compilation target.
```

```
See /scratchbox/doc/ for documentation.
```

```
[sbox-FREMANTLE_X86: ~] >
```



Работа в scratchbox - 2

- Запуск графического приложения

- в терминале 1

```
$ Xephyr :2 -host-cursor -screen 800x480x16 -dpi 96 -ac -kb &
```

- в терминале 2

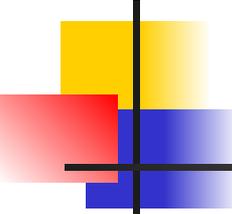
```
[sbox-FREMANTLE_X86: ~]> export DISPLAY=:2
```

```
[sbox-FREMANTLE_X86: ~]> af-sb-init.sh start
```

- Возможна интеграция с eclipse

http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide/Development_Environment/Maemo_Eclipse_Integration

http://esbox.garage.maemo.org/2nd_edition/index.html



Простой пример

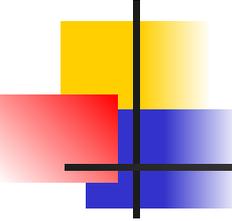
```
#include <hildon/hildon.h>

int main(int argc, char *argv[])
{
    GtkWidget *window;
    hildon_gtk_init (&argc, &argv);
    g_set_application_name ("Simplest example");
    program = hildon_program_get_instance ();
    window = hildon_window_new ();
    hildon_program_add_window (program,
        HILDON_WINDOW (window));
    gtk_widget_show (window);
    gtk_main ();
    return 0;
}
```

Задаем имя программы

Создаем главное
окно программы

Запуск главного
цикла обработки
событий



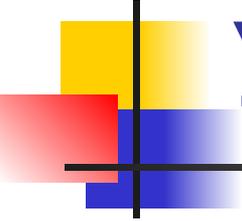
Как собрать в Scratchbox

```
$ /scratchbox/login
```

```
[sbox-FREMANTLE_X86: ~] > gcc -o hello_world `pkg-config  
--libs --cflags hildon-1` hello_world.c
```

```
[sbox-FREMANTLE_X86: ~] > ./hello_world
```

pkg-config – возвращает пути к заголовочным файлам и к файлам библиотеки



Усложненный пример - 1

- Цель: написать приложение для рисования
- Потребуется:
 - Создать главное окно программы
 - Создать область для рисования
 - Поставить фильтр на требуемые события
 - Зарегистрировать обработчики событий
 - Требуемые события: нажатие, перемещение, отпускание
 - Написать логику обработчиков событий

Рисование в окне: функция main

Добавляем
обработчик
события
«перемещение
указателя мыши»

Создание области
для рисования на
все окно

Обработка
сигнала при
закрытии
окна

Указываем какие события
хотим обрабатывать

```
...n/hildon.h>
...c, char **argv)
...ет код инициализа
...рамме
...connect (G_OBJECT (window), "destroyed", G_CALLBACK (gtk_main_quit),
...NULL,
...GtkWidget *drawing_area = gtk_drawing
...g_signal_connect (G_OBJECT (drawing_area), "motion_event",
... (GtkSignalFunc) motion_event);
// Далее так же задаем функции обраб
// и button_release_event
.....
gtk_widget_set_events (drawing_area, GDK_BUTTON_PRESS_MASK |
...GDK_BUTTON_RELEASE_MASK | GDK_POINTER_MOTION_MASK );

...gtk_container_add (GTK_CONTAINER (window), drawing_area);
// Код показа окна и запуск главного цикла обработки
.....
}
```

Рисование в окне: обработка событий

```
static gboolean button_press_event( GtkWidget *widget, GdkEventButton *event )
{
    if (event->button == 1 ) pressed=1;
    return TRUE;
}
```

```
static gboolean button_release_event( GtkWidget *widget, GdkEventButton *event )
{
    if (event->button == 1 ) pressed=0;
    return TRUE;
}
```

```
gboolean motion_event_callback( GtkWidget *widget, GdkEventMotion *event )
{
    if (pressed)
        gdk_draw_line(widget, widget->style->fg_gc, widget->window,
            x, y, event->x, event->y);
}
```

```
x = event->x;
```

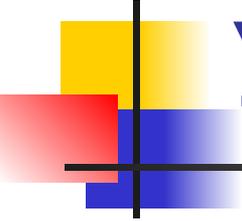
```
y = event->y;
```

```
return TRUE;
```

```
}
```

Обновление
координат

Нарисовать линию
используя стиль
текущего окна



Усложненный пример - 2

- Цель: написать приложение с кнопками
- Потребуется
 - Разобраться с расположением элементов
 - Создать необходимый контейнер
 - Создать кнопки и поместить в контейнер
 - Создать и зарегистрировать обработчики нажатий
 - Прикрепить контейнер к главному окну
 - Показать окно

Работа с диалоговым окном

```
int main (int argc, char **argv)
{
    // Инициализационная часть
    ...
    table = gtk_table_new(3,3,TRUE);
```

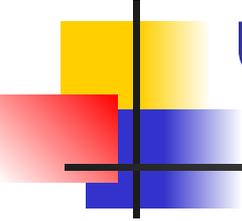
Выделенные аргументы:
левый край кнопки
правый край кнопки
верхний край
нижний край

```
    {
        _button_new_with_text (HILDON_SIZE_AUTO,
            BUTTON_ARRANGEMENT_VERTICAL,
            NULL);
        gtk_table_attach((GtkTable*)table, button,i,i+1,j,j+1,
            GTK_EXPAND, GTK_EXPAND,0,0);
        g_signal_connect (G_OBJECT (button), "clicked",
            G_CALLBACK (button_clicked), NULL);
    }
```

```
    }
    gtk_container_add (GTK_CONTAINER (window),table);
    gtk_widget_show_all (GTK_WIDGET (window));
    gtk_main ();
    return 0;
```

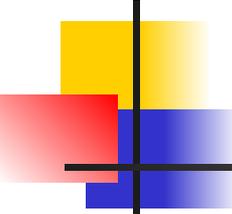
0 1 2 3

1			
2			
3			



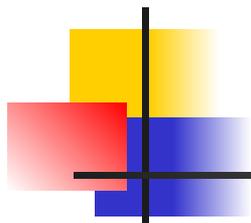
Что еще не рассмотрено

- **Элементы управления**
 - GtkRadioButtons
 - Меню (HildonAppMenu)
 - Диалоги (HildonNote, HildonBanner, HildonPickerDialog, HildonWizardDialog)
 - **Работа в сети**
 - WiFi, Bluetooth
 - Выход в интернет
- } библиотека LibConIC



Ссылки

- Главная ссылка :)
 - www.maemo.org
- Руководство для разработчиков Маемо
 - http://wiki.maemo.org/Documentation/Maemo_5_Developer_Guide
- Тьюториал по GTK для новичков
 - <http://zetcode.com/tutorials/gtktutorial/>
- Где разместить проект
 - <https://garage.maemo.org/>



Спасибо за внимание!