Google I/O 2011

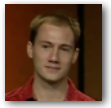
Honeycomb Highlights 報告

日本Androidの会 運営委員、(株) 富士通研究所

由良 淳一

“Honeycomb Highlights” in Google I/O 2011

- May 10, 10:15AM – 11:15AM / Room 11
  - 1日目のKeynoteの直後、Room 11（一番大きい部屋）が満員
- 発表者



Roman Guy (Graphics Acceleration)　http://curious-creature.org/

Chet Haase (Animation Framework)　http://graphics-geek.blogspot.com/

- 発表ビデオ／資料
  - http://www.google.com/events/io/2011/sessions/honeycomb-highlights.html
  - http://www.slideshare.net/romainguy/google-io-2011-android-honeycomb-highlights

Honeycombとは

- タブレット端末向けのリリース
  - スマートフォンとの違い
    - スクリーンサイズ
    - 入力手法（キーボード、マウス、…）
    - 端末能力（大きいメモリ、高速なCPU、GPU）

- 発表は今年の1月（@CES）
  - 2/22　Android 3.0 SDK リリース
  - 5/10　Android 3.1 SDK リリース

ユーザ向けの改善／機能

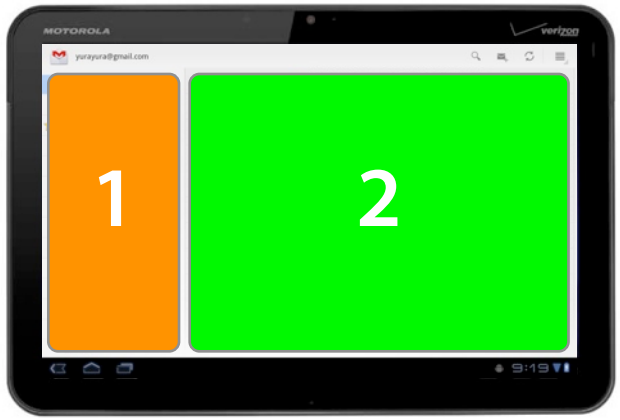
- ユーザインタフェース
  - 新しい "Holo" テーマ
  - ホーム画面の改良
  - タブレット向けキーボード
  - テキスト選択
  - USBデバイス対応
  - アクションバー、システムバー
  - 最近使ったアプリの表示
- ウィジェット
  - リッチ ＋ インタラクティブ
- アプリケーション
  - 新規
    - Books, Movie Studio
  - 更新
    - Market, Browser, Contact, Music, Gmail, …

開発者向けの改善／機能

- Fragments
- System Bar
- Action Bar
- Renderscript
- Graphics Acceleration
- Animation Framework

Fragments

- 別々のフォームファクタ（タブレット端末／スマートフォン）のUIを作るのは大変
  - 共有できるものは再利用したい
  - 画面の向き（縦／横）、画面サイズ（大／小）
- Fragment: Activityの小さいもの
  - Fragmentを組合わせてActivityを作る



Fragments

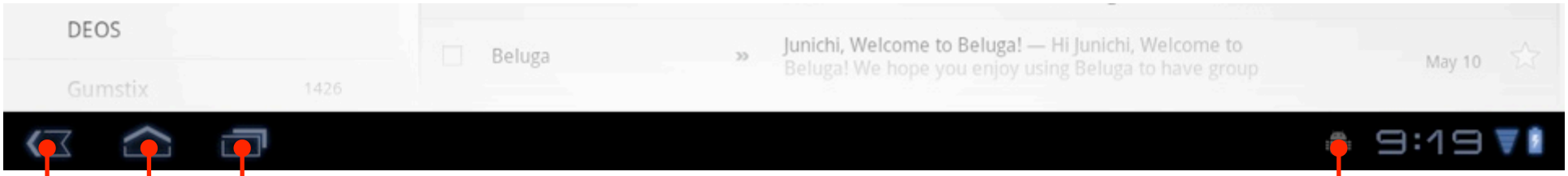
1アクティビティ内に1つのフラグメント

1アクティビティ内に2つのフラグメント

- 参考：The Android 3.0 Fragments API
  - http://android-developers.blogspot.com/2011/02/android-30-fragments-api.html

System Bar

- システム全体に関連する情報を表示する領域



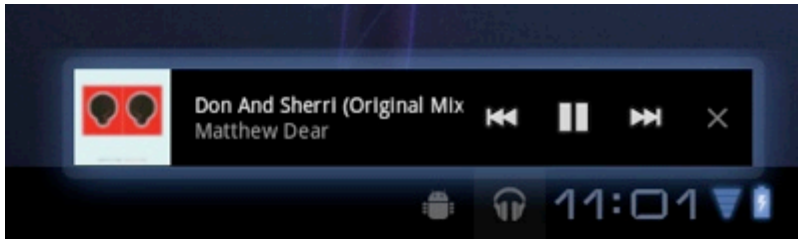
最近使ったアプリ

ホーム

バック

通知領域

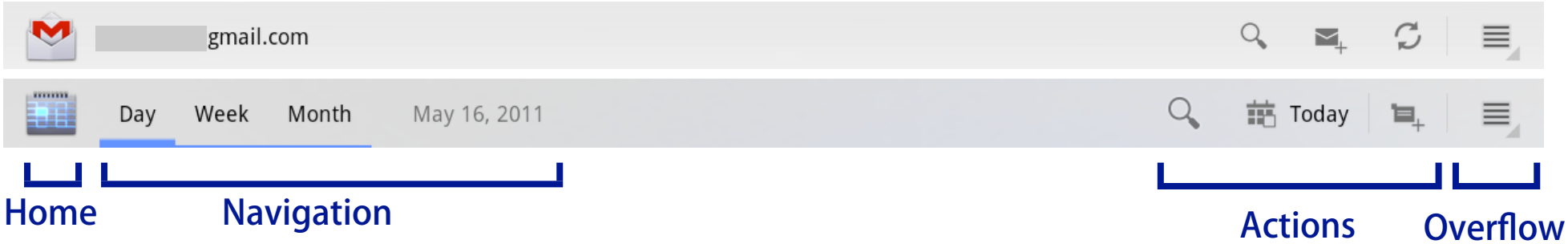
- よりインタラクティブな通知が可能
  - 音楽の場合、再生や停止など



- ハードウェアボタンの代わり
- 全画面表示している時でも有効

Action Bar

- アプリケーションに依存する情報を表示する領域
  - 従来のTitle Barに代わるもの



  - Home：起動中のアプリアイコンが表示される
  - Navigation：Activityごとに変化するアイテム
  - Actions：オプションメニューの代替
    - Overflow：メニューアイテムで表示できないものが格納される

Action Bar

```
@Override public boolean onCreatedOptionsMenu(Menu menu) {
  MenuInflater inflater = getMenuInflater();
  inflater.inflate(R.menu.actions, menu);
  return true;
}
```

actions.xml

```
<menu>
  <item android:id="@+id/action_edit"
        android:icon="@andoird:drawable/ic_menu_edit"
        android:showAsAction="always"
        android:title="@string/action_bar_edit" />
  <item android:id="@+id/action_share"
        android:icon="@andoird:drawable/ic_menu_share"
        android:showAsAction="ifRoom"
        android:title="@string/action_bar_share" />
</menu>
```

常にアイコンを表示

余裕があればアイコンを表示

Action Bar

- Action Barは、状況に応じて変更できる
  - Activityに応じた表示

Renderscript

- 描画や計算処理のためのスクリプト（ネイティブコード）
  - C99ベースの文法
  - プラットフォーム非依存
    - CPU(マルチコア対応)、GPUを自動的に選択してくれる
  - RenderScript (.rs) → 中間コード (.apk) → JITコンパイルして実行
  - Honeycomb以前にもLive Wallpaperで使われていた
- 参考：Jason Sams's Renderscript articles
  - http://android-developers.blogspot.com/2011/02/introducing-renderscript.html
  - http://android-developers.blogspot.com/2011/03/renderscript.html

Renderscript

HelloWorldRS.java

```
public class HelloWorldRS {
    private ScriptC_helloworld mScript;
    public void init(RenderScriptGL rs, Resources res) {
        mScript = new ScriptC_helloworld(rs, res, R.raw.helloworld);
        rs.bindRootScript(mScript);
    }
    public void onActionDown(int x, int y) {
        mScript.set_gTouchX(x);
        mScript.set_gTouchY(y);
    }
}
```

スクリプトを実行

スクリプト内のプロパティを変更

helloworld.rs

```
int gTouchX;
int gTouchY;
void init() {
    gTouchX = 50.0f; gTouchY = 50.0f;
}
int root(int launchID) {
    rsgClearColor(0.0f, 0.0f, 0.0f, 0.0f);
    rsgFontColor(1.0f, 1.0f, 1.0f, 1.0f);
    rsgDrawText("Hello World!", gTouchX, gTouchY);
    return 20;
}
```

画面描画時に呼ばれる

描画処理

次回の描画処理までの時間 (ms)

Graphics Acceleration

- Honeycombでは、多くのグラフィックス描画がOpenGLを用いて高速化される
  - 例）Canvas.drawLine(), Canvas.drawBitmap()
- 有効にするためには …

ApplicationManifest.xml

```
<application android:hardwareAccelerated="true">
```

  - アクティビティごとに有効化するには、<activity> の要素に指定する
  - エミュレータでは効かないので、実機でテストしてね！
- 参考：Android 3.0 Hardware Acceleration
  - http://android-developers.blogspot.com/2011/03/android-30-hardware-acceleration.html

Animation Framework

- プロパティベースのアニメーション
  - CSS AnimationやCoreAnimation (iOS)と同じ手法

```
ObjectAnimator.ofFloat(target, "alpha", 0f).start();
```

- プロパティ
  - 透明度、移動、拡大、回転 ...
- Honeycomb3.1からViewPropertyAnimatorが導入されて簡単になる
- 参考：Animation in Honeycomb
  - http://android-developers.blogspot.com/2011/02/animation-in-honeycomb.html

その他の更新

- クリップボード
- ドラッグ＆ドロップ
- HTTPライブストリーミング
- プラガブルなDRMフレームワーク
- 暗号化ストレージ
- 新規／更新されたコンポーネント
  - DatePicker, NumberPicker, StackView, CalendarView, …
- 開発ツール
  - UI Builder, コード補完, …

Android 3.1

- Android 3.1での追加機能
  - USB（キーボード以外）
  - 外部カメラのサポート
  - サイズ変更可能なウィジェット
  - RTP API → 音声／動画通信用
  - パフォーマンスの改善／最適化
  - ViewPropertyAnimatorによるアニメーションの高度化／簡易化
- 3.1の後は？
  - Honeycomb UIをより小さい画面のデバイスに適用していく
  - More, better, fancier, faster, lovelier