To enable libraries to do more, library should provide new low-level capabilities that expose the possibilities of the underlying platform as closely as possible.

Custom & Native UI Frameworks for Android

www.kandroid.org
양정수 (yangjeongsoo@gmail.com)
이경민 (snailee@gmail.com)
Current State of Android UI Framework

- UI Framework and Components
- Java API vs. Native API in Android UI Framework
- Java Stack vs. Native Stack in Android UI Framework

Case Study: Native UI Framework

- Skia: SkWindow, SKView, SkWidget
- Cocos2d-x: Director, Scene, Node
- Chromium-powered WebView
- Comparison of three case studies

Design Considerations for Native UI Framework in Android

- Synchronization with Android Event Loop
- First-class Citizen of Android View Hierarchy
- Integration with Android Hardware Acceleration
- Reusing the Android Graphics Library
- Conclusion: Reusing the features of Android
UI Framework & Component: **Looper, View, Surface, Event**

- **SurfaceFlinger Process**
  - Surface Flinger Service
  - Surface

- **System Server Process**
  - Activity Manager Service
  - Window Manager Service
  - InputManager
  - Input Channel

- **User application Process**
  - Activity Thread
  - View Root Impl
  - Input Channel
  - View
  - Activity

---

15th Kandroid MinMax - Android Rendering Pipeline
UI Framework & Component: Looper, View, Surface, Event

ActivityThread$H

LAUNCH_ACTIVITY
PAUSE_ACTIVITY
PAUSE_ACTIVITY_FINISHING
STOP_ACTIVITY_SHOW
STOP_ACTIVITY_HIDE
SHOW_WINDOW
HIDE_WINDOW
RESUME_ACTIVITY
SEND_RESULT
DESTROY_ACTIVITY
BIND_APPLICATION
EXIT_APPLICATION
NEW_INTENT
RECEIVER
CREATE_SERVICE
SERVICE_ARGS
STOP_SERVICE
CONFIGURATION_CHANGED
CLEAN_UP_CONTEXT
GC_WHEN_IDLE
BIND_SERVICE
UNBIND_SERVICE
DUMP_SERVICE
LOW_MEMORY

ACTIVITY_CONFIGURATION_CHANGED
RELAUNCH_ACTIVITY
PROFILER_CONTROL
CREATE_BACKUP_AGENT
DESTROY_BACKUP_AGENT
SUICIDE
REMOVE_PROVIDER
ENABLE_JIT
DISPATCH_PACKAGE_BROADCAST
SCHEDULE_CRASH
DUMP_HEAP
DUMP_ACTIVITY
SLEEPING
SET_CORE_SETTINGS
UPDATE_PACKAGE_COMPATIBILITY_INFO
TRIM_MEMORY
DUMP_PROVIDER
UNSTABLE_PROVIDER_DIED
REQUEST_ASSIST_CONTEXT_EXTRAS
TRANSLUCENT_CONVERSION_COMPLETE
INSTALL_PROVIDER
ON_NEW_ACTIVITY_OPTIONS
CANCEL_VISIBLE_BEHIND
BACKGROUND_VISIBLE_BEHIND_CHANGED
ENTER_ANIMATION_COMPLETE

ViewRootImpl$

• ViewRootHandler
• SyntheticJoystickHandler
• SyntheticTouchNavigationHandler

MSG_INVALIDATE
MSG_INVALIDATE_RECT
MSG_DIE
MSG_RESIZED
MSG_RESIZED_REPORT
MSG_WINDOW_FOCUS_CHANGED
MSG_DISPATCH_INPUT_EVENT
MSG_DISPATCH_APP_VISIBILITY
MSG_DISPATCH_GET_NEW_SURFACE
MSG_DISPATCH_KEY_FROM_IME
MSG_FINISH_INPUT_CONNECTION
MSG_CHECK_FOCUS
MSG_CLOSE_SYSTEM_DIALOGS;
MSG_DISPATCH_DRAG_EVENT
MSG_DISPATCH_DRAG_LOCATION_EVENT
MSG_DISPATCH_SYSTEM_UI_VISIBILITY
MSG_UPDATE_CONFIGURATION
MSG_CLEAR_ACCESSIBILITY_FOCUS_HOST
MSG_DISPATCH_DONE_ANIMATING
MSG_INVALIDATE_WORLD
MSG_WINDOW_MOVED
MSG_SYNTHESIZE_INPUT_EVENT
MSG_ENQUEUE_X_AXIS_KEY_REPEAT
MSG_ENQUEUE_Y_AXIS_KEY_REPEAT
UI Framework & Component: Looper, View, Surface, Event

SurfaceFlinger Process
- Surface Flinger Service
  - GLConsumer
  - IGraphic Buffer Consumer

System Server Process
- Activity Manager Service
- Window Manager Service
  - Buffer Queue
  - SurfaceTexture

User application Process
- Activity
  - Activity Thread
    - Looper
    - View Root Handler
  - View
  - IGraphic Buffer Producer

Surface Flinger Service
UI Framework & Component: Looper, View, Surface, (Key, Motion) Event

System Server Process

- Window Manager Service
- Input Manager
- Input Channel

User application Process

- Activity Thread
  - Looper
  - Message Queue
- ViewRootImpl.
doProcess InputEvent()

Input Pipeline

- InputChannel
- InputChannel
- NativeInputEventReceiver
- WindowInputEventReceiver
- View

- SyntheticInputStage
- ViewPostImeInputStage
- NativePostImeInputStage
- EarlyPostImeInputStage
- ImeInputStage
- ViewPreImeInputStage
- NativePreImeInputStage

Looper, View, Surface, (Key, Motion) Event
UI Framework & Component:Looper, View, Surface, (Key, Motion) Event

System Server Process

User application Process

Looper

Frame Handler

ViewRoot Handler

Traversal Runnable

Consumed BatchedRunnable

ViewRootImpl.
doProcess InputEvent

Input Pipeline

View

Message Queue

Activity Thread

Looper

TLS

FrameDisplay EventReceiver

WindowInput EventReceiver

NativeDisplay EventReceiver

NativeInput EventReceiver

Input Channel

Input Manager

Window Manager Service

Input Channel

paired socket
UI Framework & API: Java vs. Native

User application Process

Activity Thread
- ViewRoot Handler
- Frame Handler
- Looper
- Message Queue
- NativeDisplay EventReceiver
- NativeInput EventReceiver
- Input Channel

Looper
- Traversal Runnable
- Animation Handler
- Consumed BatchedInput Runnable
- ViewRootImpl doProcess InputEvent
- Input Pipeline

View
- FrameDisplay EventReceiver
- WindowInput EventReceiver
- NativeDisplay EventReceiver
- NativeInput EventReceiver
- Frame Handler
- View

Canvas vs. OpenGL
- Surface

15th Kandroid MinMax - Android Rendering Pipeline
UI Framework & API: Java vs. Native

Activity Thread
- Looper
- Message Queue
  - loop()
  - TLS

Main EventLoop
- ViewRoot Handler & Frame Handler

UI Component
- Window
- View
- ViewRoot
- Widget
- Canvas & OpenGL
- Input
- Surface

UI Framework & API: Java vs. Native

Input
Surface
Canvas & OpenGL
Widget
View
Window
ViewRoot
Choreographer
Looper
Message Queue
loop()
UI Framework & API: Java vs. Native

**Main EventLoop**
- ViewRoot
- Handler & Frame Handler
- Choreographer

**UI Component**
- Window
- View
- Widget

**Activity Thread**
- Looper
- Message Queue
- Looper

**TLS**

**Canvas & OpenGL**
- Input
- Surface
UI Framework & Stack: Java vs. Native

User application Process

**Activity**
- **Thread**
  - **Looper**
  - **Message Queue**

**ViewRoot**
- **Handler**
  - **Traversal Runnable**
  - **Animation Handler**
  - **Consumed BatchedInput Runnable**
  - **ViewRootImpl.doProcess InputEvent**

**Input Pipeline**

**View**

**Canvas vs. OpenGL**
- Native Looper
- Native Event
- Native Canvas vs OpenGL
- Native Surface

**Surface**

**Looper**
- **NativeDisplay EventReceiver**
- **NativeInput EventReceiver**
- **Input Channel**

15th Kandroid MinMax - Android Rendering Pipeline
Current State of Android UI Framework
• UI Framework and Components
• Java API vs. Native API in Android UI Framework
• Java Stack vs. Native Stack in Android UI Framework

Case Study: Native UI Framework
• Skia: SkWindow, SKView, SkEventSink
• Cocos2d-x: Director, Scene, Node
• Chromium-powered WebView
• Comparison of three case studies

Design Considerations for Native UI Framework in Android
• Synchronization with Android Event Loop
• First-class Citizen of Android View Hierarchy
• Integration with Android Hardware Acceleration
• Reusing the Android Graphics Library
• Conclusion: Reusing the features of Android
Skia: **SkWindow, SkView, SkEventSink**

Source: [http://people.mozilla.org/~bgirard/doxygen/gfx/classSkOSWindow.html](http://people.mozilla.org/~bgirard/doxygen/gfx/classSkOSWindow.html)

Source: [http://people.mozilla.org/~bgirard/doxygen/gfx/classSkView.html](http://people.mozilla.org/~bgirard/doxygen/gfx/classSkView.html)
Skia: **SkWindow, SkView, SkEventSink**

Activity Thread:
- **Looper**
- **Frame Handler**

Traversal Runnable:
- **View**
- **View**
- **View**

Message Queue:

TLS

JNI

AndroidBitmap_lockPixels();

- SkSurface
- SkCanvas
- SkPaint
- SkString

draw Bitmap

AndroidBitmap_unlockPixels();
Skia: SkWindow, SkView, SkEventSink

Activity Thread
- Looper
- Frame Handler
- Message Queue

Traversal Runnable
- View

GLThread
- GLSurfaceView

SkWindow
- SkSurface
- SkCanvas
- SkPaint
- SkString
- ...

JNI

- initialize
- draw
- terminate
Cocos2d-x: CCDirector, CCNode, CCScene,

Source: http://learn-cocos2d-by-example.blogspot.kr/2013/01/understanding-ccdirector.html

Source: https://www.safaribooksonline.com/library/view/cocos2d-2/9781430244165/Sec2_9781430244165_Ch03.xhtml
Cocos2d-x: CCDirector, CCNode, CCScene,

- CCDirector
  - nativeInit
  - drawScene
  - ...

- CCScene
- CCLayer
- CCSprite
- ...

- CCDirectorIOS
- CCDirectorDisplayLink
- CCDirectorFastThreaded
- CCDirectorTimer

Activity
Thread
Looper
Message
Queue

H
Traversal
Runnable

Cocos2dxGLSurfaceView

GLThread

JNI

15th Kandroid MinMax - Android Rendering Pipeline
Chromium-powered WebView: WebViewFactory

source: http://outcesticide.hatenablog.com/entry/android_webview
Comparison of three case studies: **Skia, Cocos2d-x, Chromium-powered WebView**

<table>
<thead>
<tr>
<th></th>
<th>Skia</th>
<th>Cocos2d</th>
<th>Chromium-powered WebView</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td></td>
<td>CCDirector</td>
<td>Built-in (Back &amp; Forward)</td>
</tr>
<tr>
<td>Window</td>
<td>SkWindow</td>
<td>CCScene</td>
<td>DOM Document</td>
</tr>
<tr>
<td>View</td>
<td>SkView</td>
<td>CCNode</td>
<td>DOM Elements</td>
</tr>
<tr>
<td>Widget</td>
<td>SkWidget ...</td>
<td></td>
<td>DOM Elements</td>
</tr>
<tr>
<td>Event</td>
<td>SkEventSink</td>
<td></td>
<td>DOM Events</td>
</tr>
<tr>
<td>Canvas</td>
<td>SkCanvas</td>
<td></td>
<td>HTML5 Canvas</td>
</tr>
</tbody>
</table>
Current State of Android UI Framework

- UI Framework and Components
- Java API vs. Native API in Android UI Framework
- Java Stack vs. Native Stack in Android UI Framework

Case Study: Native UI Framework

- Skia: SkWindow, SKView, SkEventSink
- Cocos2d-x: Director, Scene, Node
- Chromium-powered WebView
- Comparison of three case studies

Design Considerations for Native UI Framework in Android

- Synchronization with Android Event Loop
- First-class Citizen of Android View Hierarchy
- Integration with Android Hardware Acceleration
- Reusing the Android Graphics Library
- Conclusion: Reusing the features of Android
Main Event Loop 동기화 : Case Study – NativeActivity

Single Thread

Threaded App
Sync between **Main Thread & NativeActivity Thread**

**Java**
- **ActivityThread**
  - Looper
  - Message Queue
- **Input Pipeline**
  - ViewRoot
  - Handle Message
  - Handle Message

**Native**
- **Looper**
- **Native Activity**
  - mainWorkRead
  - Input Queue
  - mainWorkWrite

**Custom Library**
- **android_main()**
- **android_native_app_glue**
- **ALooper_pollAll()**
- **ALooper_prepare()**
- **mDispatchReadFd**
- **mDispatchWriteFd**

**Thread**
- **mDispatchReadFd**
- **msgread**
- **msgwrite**

**CMDs**
- **CMD_FINISH**
- **CMD_SET_WINDOW_FORMAT**
- **CMD_SET_WINDOW_FLAGS**
- **CMD_SHOW_SOFT_INPUT**
- **CMD_HIDE_SOFT_INPUT**

**APP_CMDs**
- **APP_CMD_INPUT_CHANGED**
- **APP_CMD_INIT_WINDOW**
- **APP_CMD_TERM_WINDOW**
- **APP_CMD_WINDOW_RESIZED**
- **APP_CMD_WINDOW_REDRAW_NEEDED**
- **APP_CMD_CONTENT_RECT_CHANGED**
- **APP_CMD_GAINED_FOCUS**
- **APP_CMD_LOST_FOCUS**
- **APP_CMD_CONFIG_CHANGED**
- **APP_CMD_LOW_MEMORY**
- **APP_CMD_START**
- **APP_CMD_RESUME**
- **APP_CMD_SAVE_STATE**
- **APP_CMD_PAUSE**
- **APP_CMD_STOP**
- **APP_CMD_DESTROY**
View Hierarchy 연동 : SurfaceView, GLSurfaceView
View Hierarchy 연동: **SurfaceView, GLSurfaceView**

- **Activity**
  - Thread
  - Looper
  - Message Queue

- **PhoneWindow**
  - $DecorView (FrameLayout)

- **LinearLayout**

- **ArrayList<Runnable> mEventQueue**

  ```java
  void queueEvent(Runnable r) {
    mEventQueue.add(r);
  }
  ```

- **Window Manager Service**
  - LayoutParams (for window layer)

- **GLSurfaceView**
  - EGL Helper (EGL Context)
  - GLThread
  - EGL Surface

- **SurfaceView**
  - Renderer
  - Surface
  - TextView
**Skia**: SkWindow, SkView, SkEventSink

---

```java
public boolean onTouchEvent(MotionEvent event) {
    ...
    queueEvent(new Runnable() {
        @Override
        public void run() {
            mSampleRenderer.handleClick(owner, x, y, finalAction);
        }
    });
    return true;
}
```
Cocos2d-x: **CCDirector, CCNode, CCScene,**

```java
public boolean onTouchEvent(MotionEvent event) {
    ... queueEvent(new Runnable() {
        @Override
        public void run() {
            Cocos2dxGLSurfaceView.this.mCocos2dxRenderer.handleActionDown(...);
        }
    });
    return true;
}
```
Hardware Acceleration 연동: RenderThread
Hardware Acceleration 연동 : RenderThread

UI (Main) Thread

UpdateRootDisplayList(...);
updateAndDrawFrame(...);

if (canUnblockUiThread) {
    unblockUiThread();
}

if (canDrawThisFrame) {
    context->draw();
}

if (!canUnblockUiThread) {
    unblockUiThread();
}

Hardware Acceleration 연동:

Reference: http://blog.csdn.net/jinzhuojun/article/details/44062175
Hardware Acceleration 연동 : RenderThread – GLThread

Activity Thread
-Looper
-Message Queue
-Frame Handler
-Traversal Runnable
-View

GLThread
-GLSurfaceView

SkyWindow
- SkSurface
- SkCanvas
- SkPaint
- SkString
- ...

JNI

- initialize
- draw
- terminate
Hardware Acceleration 연동 : RenderThread – Animation

Activity
- Event
- Set Property Value
- Invalidate
- Something Happens

Draw
- Measure & Layout
- Prepare Draw
- Update DisplayList
- Draw DisplayList
- Swap Buffers
- Render Thread

SurfaceFlinger
- Dequeue Buffer
- Enqueue Buffer

Display
- Composite Windows
- Post Buffer

Main Thread
- Animator

Render Thread
Hardware Acceleration 연동 : RenderThread – Animation

Main Thread

Render Thread

Activity

Event → Set Property Value → Invalidate

Measure & Layout

Prepare Draw

Update DisplayList

Display List

Draw DisplayList

Swap Buffers

Something Happens

Dequeue Buffer

Enqueue Buffer

Display

Composite Windows

Post Buffer
Reusing the Android Graphics Library

<table>
<thead>
<tr>
<th>Activity</th>
<th>Skia</th>
<th>Cocos2d</th>
<th>Chromium-powered WebView</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>CCDirector</td>
<td>Built-in</td>
<td></td>
</tr>
<tr>
<td>Window</td>
<td>SkWindow</td>
<td>CCScene</td>
<td>DOM Document</td>
</tr>
<tr>
<td>View</td>
<td>SkView</td>
<td>CCNode</td>
<td>DOM Elements</td>
</tr>
<tr>
<td>Widget</td>
<td>SkWidget</td>
<td></td>
<td>DOM Elements</td>
</tr>
<tr>
<td>Event</td>
<td>SkEventSink</td>
<td></td>
<td>DOM Events</td>
</tr>
<tr>
<td>Canvas</td>
<td>SkCanvas</td>
<td></td>
<td>HTML5 Canvas</td>
</tr>
</tbody>
</table>

Diagram showing the integration of Skia, Cocos2d, and Chromium-powered WebView.
Conclusion

Reusing the features of Android