Google™ 09 IO

Turbo-charge your UI

Romain Guy
May 29, 2009

Disclaimer

Agenda

- Adapters
- Backgrounds and images
- Drawing and invalidating
- Views and layouts
- Memory allocations

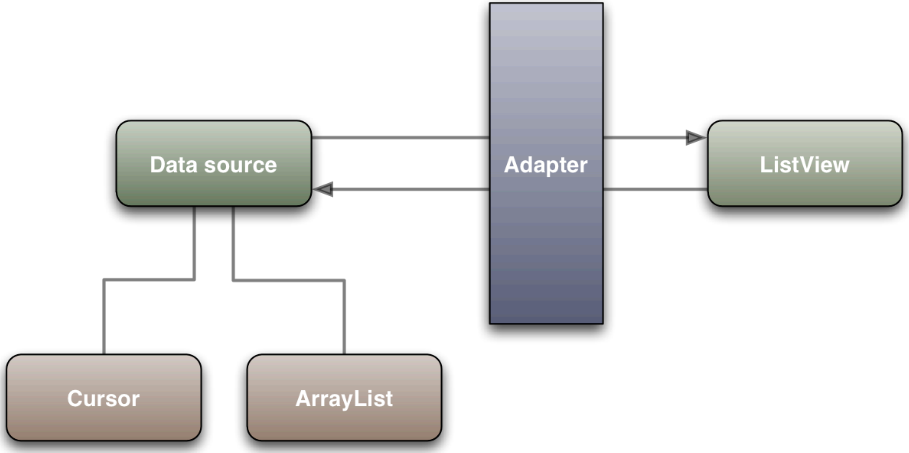
Agenda

- Adapters
- Backgrounds and images
- Drawing and invalidating
- Views and layouts
- Memory allocations

Adapters

- Awesome
- Painful
- Do you even know how ListView works?

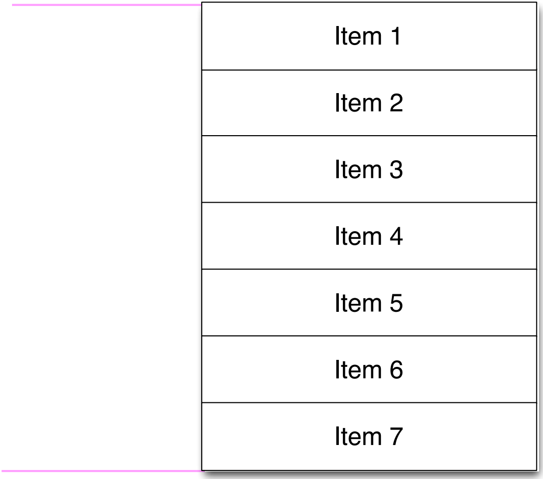
Man in the middle



Gimme views

- For each position
  - Adapter.getView()
- A new View is returned
  - Expensive
- What if I have 1,000,000 items?

Getting intimate with ListView



Don’t

```
public View getView(int position, View convertView, ViewGroup parent) {
    View item = mInflater.inflate(R.layout.list_item_icon_text, null);

    ((TextView) item.findViewById(R.id.text)).setText(DATA[position]);
    ((ImageView) item.findViewById(R.id.icon)).setImageBitmap(
            (position & 1) == 1 ? mIcon1 : mIcon2);

    return item;
}
```

Do

```
public View getView(int position, View convertView, ViewGroup parent) {
    if (convertView == null) {
        convertView = mInflater.inflate(R.layout.item, null);
    }

    ((TextView) convertView.findViewById(R.id.text)).setText(DATA[position]);
    ((ImageView) convertView.findViewById(R.id.icon)).setImageBitmap(
            (position & 1) == 1 ? mIcon1 : mIcon2);

    return convertView;
}
```

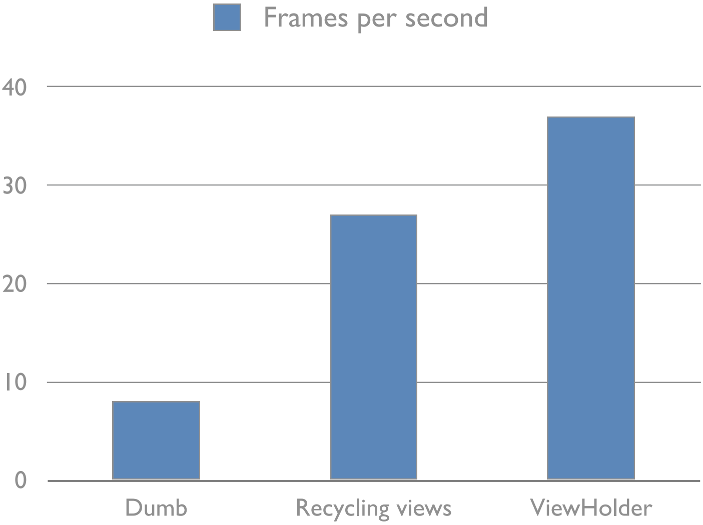
Even better

```
static class ViewHolder {
    TextView text;
    ImageView icon;
}
```

Even better

```java
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder;

    if (convertView == null) {
        convertView = mInflater.inflate(R.layout.list_item_icon_text, null);

        holder = new ViewHolder();
        holder.text = (TextView) convertView.findViewById(R.id.text);
        holder.icon = (ImageView) convertView.findViewById(R.id.icon);

        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }

    holder.text.setText(DATA[position]);
    holder.icon.setImageBitmap((position & 1) == 1 ? mIcon1 : mIcon2);

    return convertView;
}
```

Is it worth it?



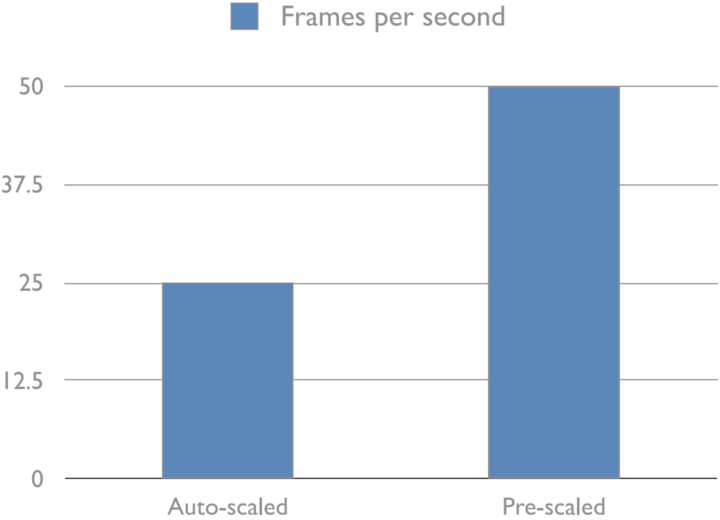
Agenda

- Adapters
- Backgrounds and images
- Drawing and invalidating
- Views and layouts
- Memory allocations

Don’t be greedy

- Background drawables always fit the view
    - Stretching may occur
- Runtime scaling is expensive

How expensive?



Pre-scaling is easy

```
// Rescales originalImage to the size of view using
// bitmap filtering for better results

originalImage = Bitmap.createScaledBitmap(
    originalImage,       // bitmap to resize
    view.getWidth(),     // new width
    view.getHeight(),    // new height
    true);               // bilinear filtering
```

Window backgrounds

- Sometimes unnecessary
  - Top-level opaque view
  - layout_width=fill_parent
  - layout_height=fill_parent
- Expensive to draw
- Dumb rendering engine
  - (And it’s my fault)

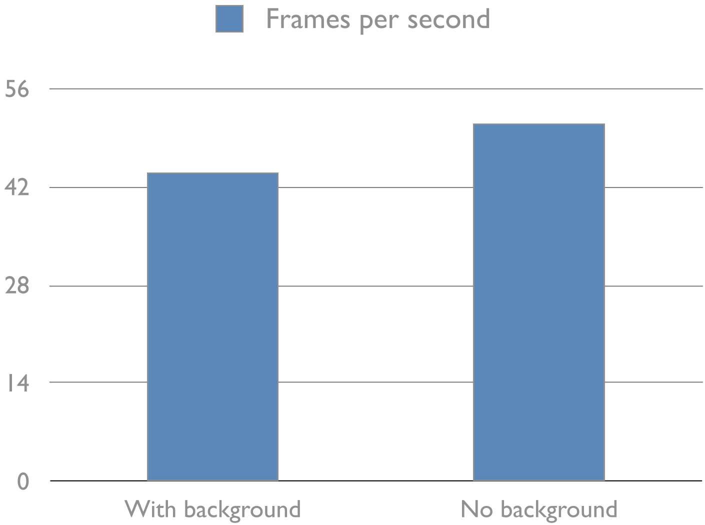
Removing the background

```xml
<!-- res/values/styles.xml -->
<resources>
    <style name="Theme.NoBackground" parent="android:Theme">
        <item name="android:windowBackground">@null</item>
    </style>
</resources>
```

Removing the background

```
<activity
    android:name="MyApplication"
    android:theme="@style/NoBackgroundTheme">

    <!-- intent filters and stuff -->

</activity>
```

What do I get?



Frames per second
56
42
28
14
0
With background
No background

Good news!

Agenda

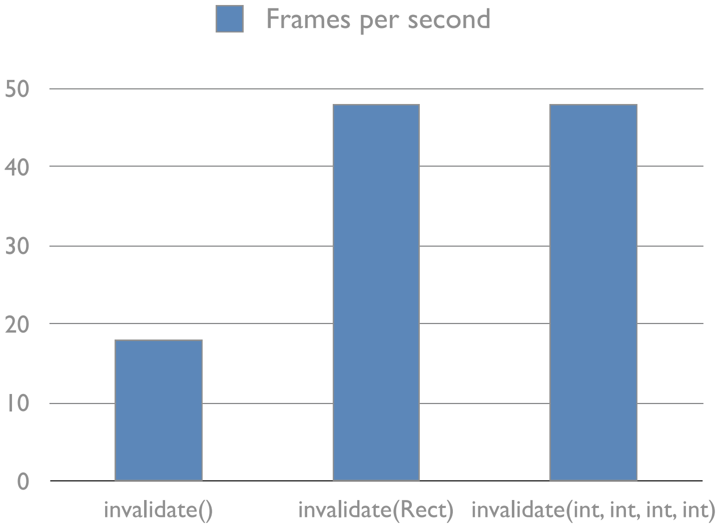
- Adapters
- Backgrounds and images
- Drawing and invalidating
- Views and layouts
- Memory allocations

Redraw efficiently

- invalidate()
  - So easy
  - So expensive
- Dirty regions
  - invalidate(Rect)
  - invalidate(left, top, right, bottom)

Demo, invalidating Home

Just do it



🟦 Frames per second
50
40
30
20
10
0
invalidate()
invalidate(Rect)
invalidate(int, int, int, int)

Agenda

- Adapters
- Backgrounds and images
- Drawing and invalidating
- Views and layouts
- Memory allocations

Fewer is better

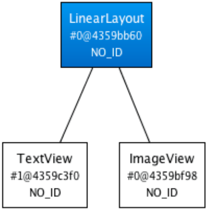
- Many views
  - Longer startup time
  - Slower measurement
  - Slower layout
  - Slower drawing
- Deep hierarchies
  - StackOverflowException
  - Slow... slow... slow...

HierarchyViewer

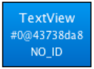
A few solutions

- TextView’s compound drawables
- ViewStub
- <merge />
- RelativeLayout
- Custom views
- Custom layouts

Compound drawables

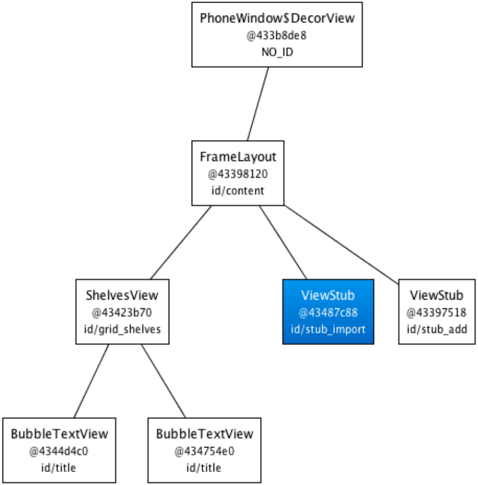


```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/icon" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```

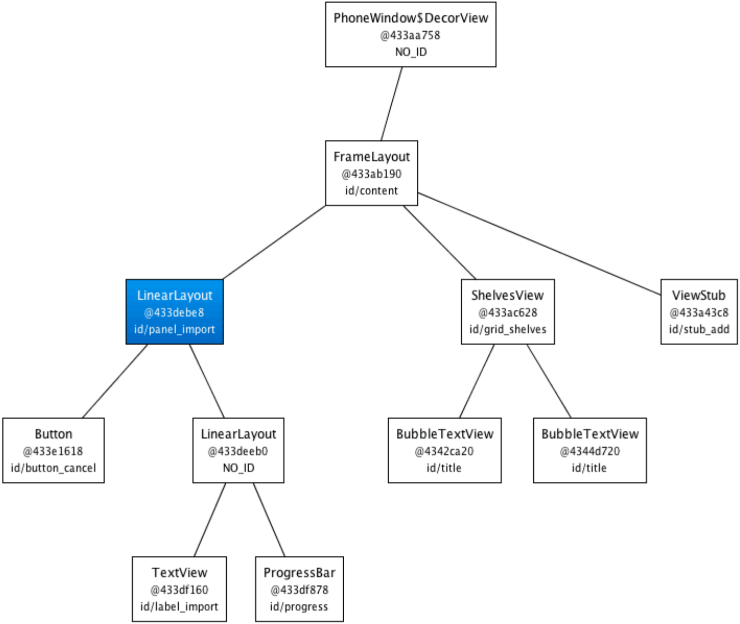


```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    android:drawableLeft="@drawable/icon" />
```

ViewStub


PhoneWindow$DecorView
@433b8de8
NO_ID
FrameLayout
@43398120
id/content
ShelvesView
@43423b70
id/grid_shelves
ViewStub
@43487c88
id/stub_import
ViewStub
@43397518
id/stub_add
BubbleTextView
@4344d4c0
id/title
BubbleTextView
@434754e0
id/title

ViewStub


PhoneWindow$DecorView
@433aa758
NO_ID
FrameLayout
@433ab190
id/content
LinearLayout
@433debe8
id/panel_import
ShelvesView
@433ac628
id/grid_shelves
ViewStub
@433a43c8
id/stub_add
Button
@433e1618
id/button_cancel
LinearLayout
@433deeb0
NO_ID
BubbleTextView
@4342ca20
id/title
BubbleTextView
@4344d720
id/title
TextView
@433df160
id/label_import
ProgressBar
@433df878
id/progress

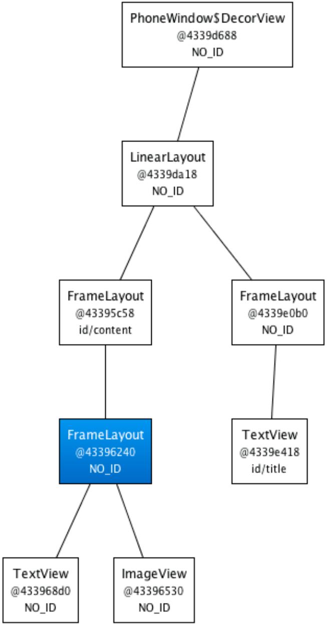
ViewStub

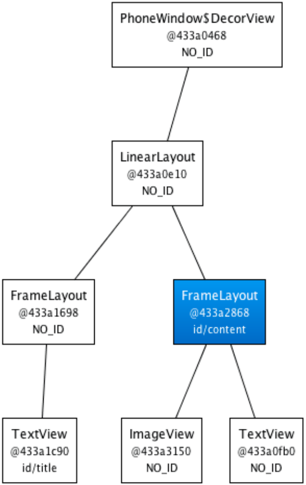
```
<ViewStub
    android:id="@+id/stub_import"
    android:inflatedId="@+id/panel_import"

    android:layout="@layout/progress_overlay"

    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom" />
```

Inflating a ViewStub

```java
findViewById(R.id.stub_import).setVisibility(View.VISIBLE);

// or

View importPanel = ((ViewStub)
        findViewById(R.id.stub_import)).inflate();
```

<merge />


PhoneWindow$DecorView
@4339d688
NO_ID
LinearLayout
@4339da18
NO_ID
FrameLayout
@43395c58
id/content
FrameLayout
@4339e0b0
NO_ID
FrameLayout
@43396240
NO_ID
TextView
@4339e418
id/title
TextView
@433968d0
NO_ID
ImageView
@43396530
NO_ID


PhoneWindow$DecorView
@433a0468
NO_ID
LinearLayout
@433a0e10
NO_ID
FrameLayout
@433a1698
NO_ID
FrameLayout
@433a2868
id/content
TextView
@433a1c90
id/title
ImageView
@433a3150
NO_ID
TextView
@433a0fb0
NO_ID

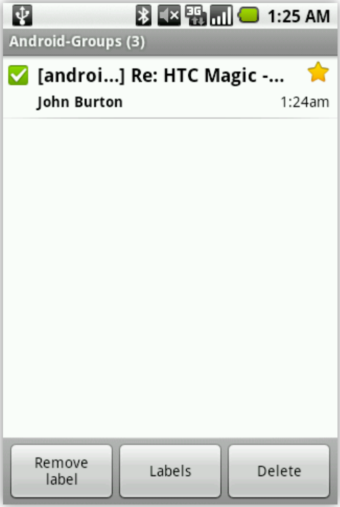
<merge />

```xml
<!-- The merge tag must be the root tag -->
<merge xmlns:android="http://schemas.android.com/apk/res/android">

    <!-- Content -->

</merge>
```

RelativeLayout

- Powerful
- Replace linear layouts
  - A horizontal LinearLayout in a vertical one
  - Or the other way around
- Sometimes hard to use
  - (And it’s all my fault)

Custom views



1:25 AM

Android-Groups (3)

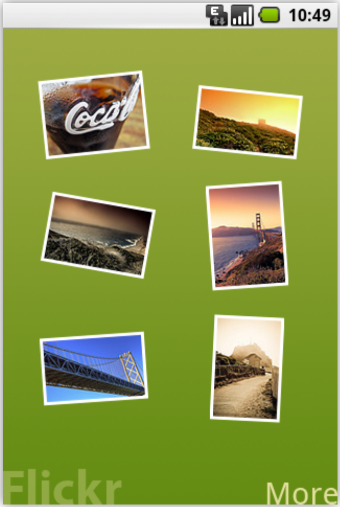
☑ [androi...] Re: HTC Magic -...

John Burton 1:24am

Custom views

```java
class CustomView extends View {
    public CustomView(Context context) {
        super(context);
    }

    @Override
    protected void onDraw(Canvas canvas) {
    }

    @Override
    protected void onMeasure(int widthMeasureSpec,
            int heightMeasureSpec) {
        setMeasuredDimension(100, 100);
    }
}
```

Custom layouts



Custom layouts

```java
public class GridLayout extends ViewGroup {
    @Override
    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
        final int count = getChildCount();
        for (int i = 0; i < count; i++) {
            final View child = getChildAt(i);
            // Define measurement spec of each child
            child.measure(childWidthSpec, childheightSpec);
        }

        setMeasuredDimension(widthSpecSize, heightSpecSize);
    }

    @Override
    protected void onLayout(boolean changed, int l, int t, int r, int b) {
        final int count = getChildCount();
        for (int i = 0; i < count; i++) {
            View child = getChildAt(i);
            if (child.getVisibility() != GONE) {
                // Compute position of each child
                child.layout(left, top, right, bottom);
            }
        }
    }
}
```

Agenda

- Adapters
- Backgrounds and images
- Drawing and invalidating
- Views and layouts
- Memory allocations

DO NOT ALLOCATE MEMORY

- As much as you can
- GC
  - Stops the world
  - Slow (~x00 ms)

Performance sensitive paths

Measurement	`onMeasure()`
Layout	`onLayout()`

Drawing	`draw()`
	`dispatchDraw()`
	`onDraw()`

Events handling	`dispatchTouchEvent()`
	`onTouchEvent()`

Adapters	`getView()`
	`bindView()`

Fail fast

```java
int prevLimit = -1;
try {
    // Limit the number of allocated objects
    prevLimit = Debug.setAllocationLimit(0);

    // Execute code that should not perform
    // any allocation
} finally {
    Debug.setAllocationLimit(prevLimit);
}
```

Demo, allocation tracker

Manage your objects

- SoftReferences
  - Excellent for memory caches
- WeakReferences
  - Avoids memory leaks

Simple cache

```
private final HashMap<String, SoftReference<T>> mCache;

public put(String key, T value) {
    mCache.put(key, new SoftReference<T>(value));
}

public T get(String key, ValueBuilder builder) {
    T value = null;
    SoftReference<T> reference = mCache.get(key);

    if (reference != null) {
        value = reference.get();
    }

    // Not in cache or gc'd
    if (value == null) {
        value = builder.build(key);
        mCache.put(key, new SoftReference<T>(value));
    }

    return value;
}
```

Resources

- http://d.android.com
- http://source.android.com
- http://android.git.kernel.org
- http://code.google.com/p/apps-for-android
- http://code.google.com/p/shelves

Q&A

Google™ 09 I O