



Android Graphics and Animations

Romain Guy and Chet Haase
Google



Welcome

@romainguy

@chethaase

Agenda

- Architecture
- Graphics
- Animations

Agenda

- Architecture
- Graphics
- Animations

Glossary

Canvas	2D drawing context
Skia	2D drawing API
OpenGL	3D rendering API
RenderScript	Language + API

Glossary

Surface

Drawing buffer

SurfaceFlinger

Surface manager

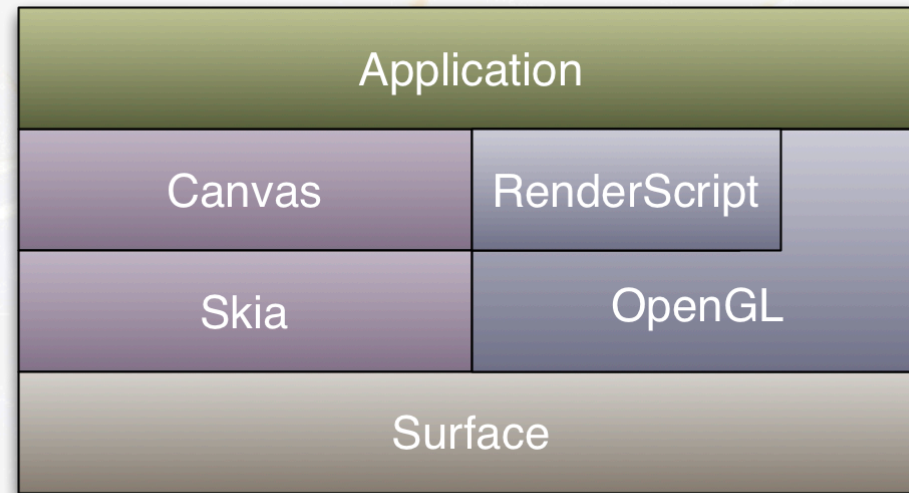
PixelFlinger

Rasterizer

Glossary

View	UI widget
ViewGroup	View container
SurfaceView	Render in a surface

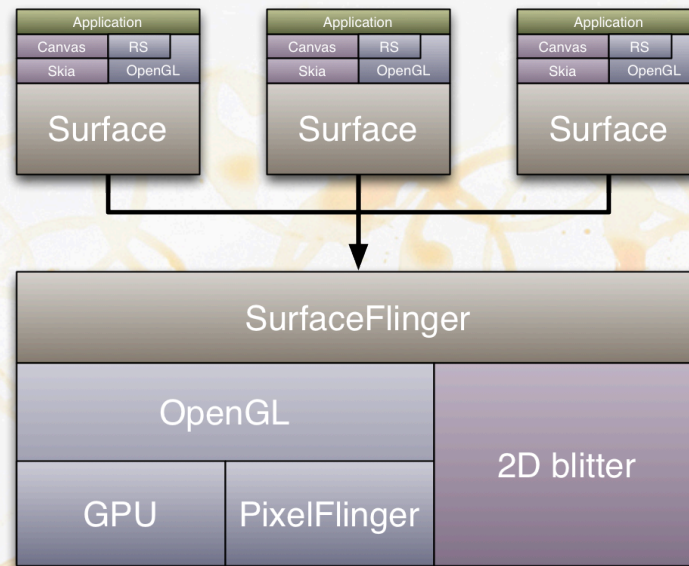
Architecture



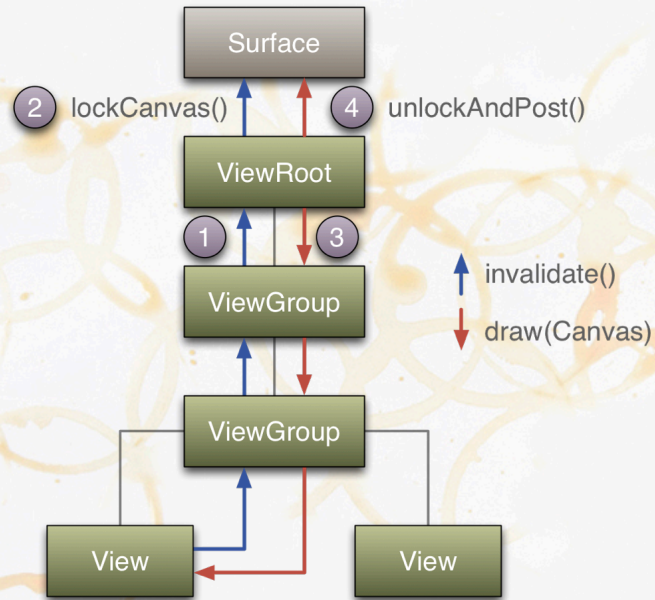
OpenGL



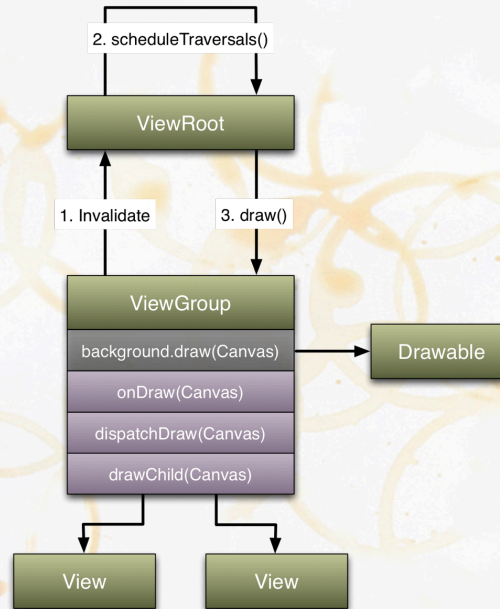
Compositor



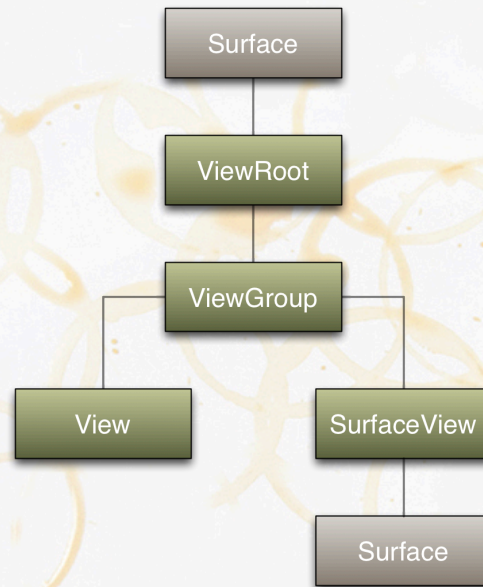
Views



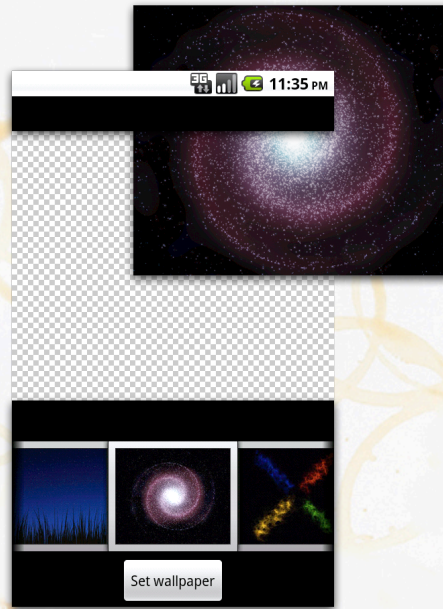
ViewGroups



SurfaceView

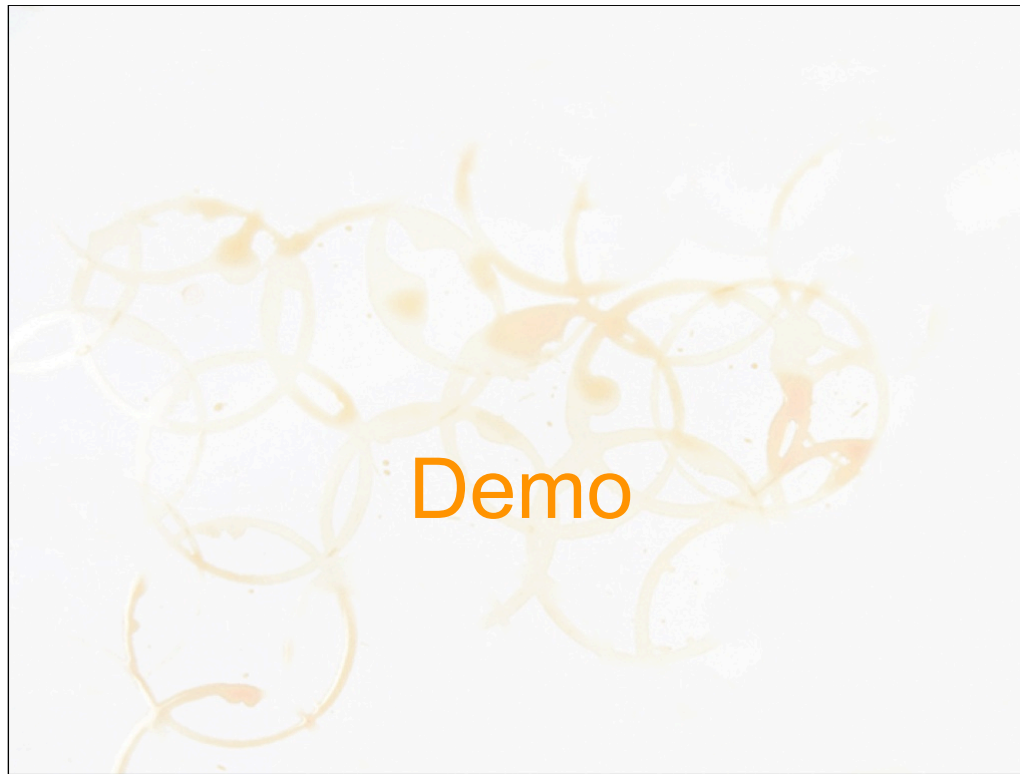


SurfaceView



Agenda

- Architecture
- Graphics
- Animations



Drawing tools

- Paints
- Shaders
- ColorFilters
- Xfermodes
- Bitmaps

Paints

- Canvas is almost stateless
 - Transformations
 - Layers
- Paint has many states
 - Color, opacity, filtering, dithering, anti-aliasing...
- Don't allocate paints onDraw()
 - Paint is not cheap

Shaders

- Draw horizontal span of colors
 - Text, paths, rounded rectangles, etc.
- Kinda like fragment shaders
- Pre-defined set
 - LinearGradient
 - RadialGradient
 - SweepGradient
 - BitmapShader
 - ComposeShader

Color filters

- Math operation on each pixel
- Pre-defined set
 - ColorMatrixColorFilter
 - LightingColorFilter
 - PorterDuffColorFilter

Xfermodes

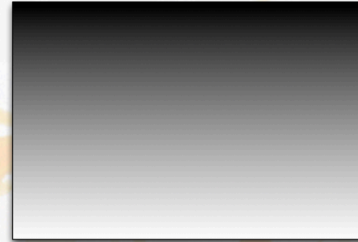
- Weird name for blending modes
- Porter-Duff alpha blending
 - SrcOver
 - DstOut
 - etc.
- Color blending
 - Darken
 - Lighten
 - Multiply
 - Screen

Reflection time

BitmapShader



LinearGradient



ComposeShader

Reflection time

```
1 Shader gradientShader = new LinearGradient(  
2     0, 0, 0, b.getHeight(), 0xFF000000, 0,  
3     TileMode.CLAMP);  
  
4 Shader bitmapShader = new BitmapShader(bitmap,  
5     TileMode.CLAMP, TileMode.CLAMP);  
  
6 Shader composeShader = new ComposeShader(  
7     bitmapShader, gradientShader,  
8     new PorterDuffXfermode(Mode.DST_OUT));  
  
9 Paint paint = new Paint();  
10 paint.setShader(composeShader);  
  
11 c.drawRect(0.0f, 0.0f,  
12     b.getWidth(), b.getHeight(), p);
```

Bitmaps

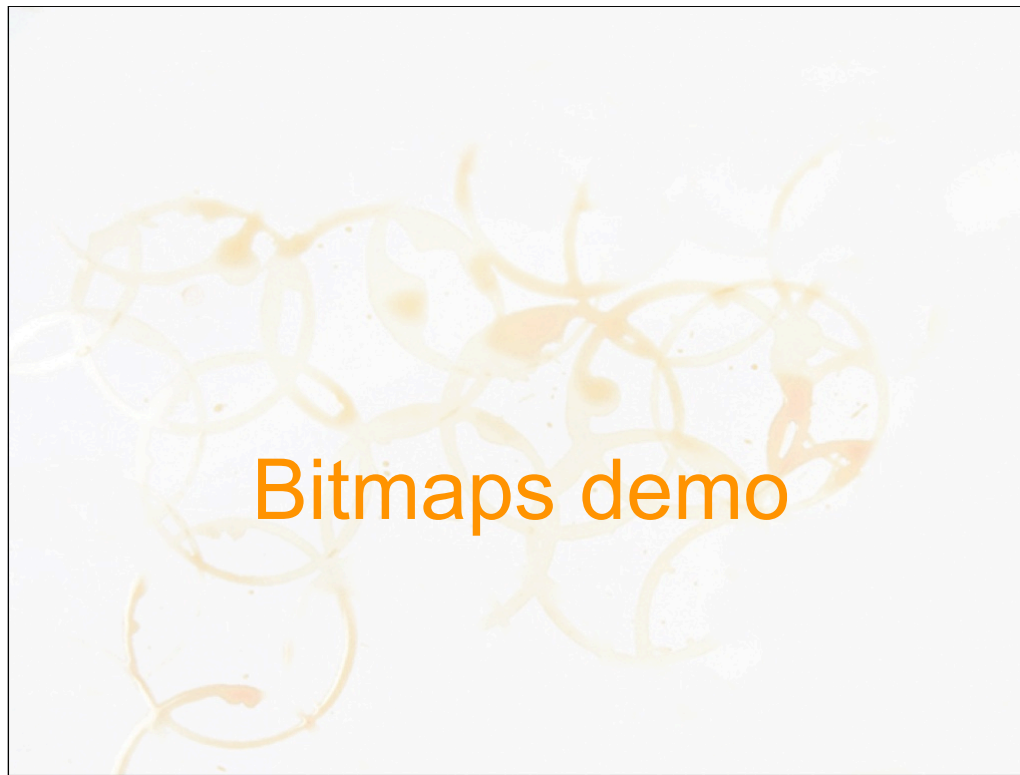
- Mutable or immutable
- Density
- Recyclable
- 4 formats

Formats

- ALPHA_8
To store alpha masks (font cache, etc.)
- ARGB_4444
Don't use it
- ARGB_8888
Full color, translucency, it's awesome
- RGB_565
No alpha channel, saves memory, dithering

Choose the right format

- Quality
 - Preload Bitmaps in the right format
 - `Paint.setDither(true)`
 - `Drawable.setDither(true)`
- Performance
- Avoid blending
 - Opaque ARGB_8888 bitmaps are optimized
- Render onto compatible surfaces
 - Draw 32 bits onto 32 bits windows, etc.
 - `getWindow().getAttributes().format`



Performance

	16 bits	16 bits dithered	32 bits
ARGB_8888	6.0 ms	7.5 ms	2.0 ms
ARGB_4444	4.0 ms	5.0 ms	3.5 ms
RGB_565	0.5 ms	0.5 ms	6.0 ms

Performance measured with HVGA 2.2 emulator

In Gingerbread...

- All windows are 32 bits
 - Transparent: RGBA_8888
 - Opaque: RGBX_8888
- OpenGL surfaces are 16 bits
- All bitmaps loaded in 32 bits (ARGB_8888)
- For quality reasons
 - No more banding, no more dithering

Create, save, destroy

- `Bitmap.createBitmap()`
- `Bitmap.createScaledBitmap()`
- `BitmapFactory.decode*()`
- `Bitmap.copy()`
- `Bitmap.compress()`
- `Bitmap.recycle()`

Draw on Bitmap

- Only if mutable

```
1 Paint p = new Paint();
2 p.setAntiAlias(true);
3 p.setTextSize(24);
4
5 Bitmap b = Bitmap.createBitmap(256, 256,
6     Bitmap.Config.ARGB_8888);
7 Canvas c = new Canvas(b);
8 c.drawText("Devovx", 0.0f, 128.0f, p);
```

Copy a View

```
1 int spec = MeasureSpec.makeMeasureSpec(  
2     0, MeasureSpec.UNDEFINED);  
3 view.measure(spec, spec);  
4 view.layout(0, 0, view.getMeasuredWidth(),  
5     view.getMeasuredHeight());  
6  
7 Bitmap b = Bitmap.createBitmap(  
8     view.getWidth(), view.getHeight(),  
9     Bitmap.Config.ARGB_8888);  
10 Canvas c = new Canvas(b);  
11 c.translate(-view.getScrollX(), -view.getScrollY());  
12 view.draw(c);
```


Copy a View

```
1 view.setDrawingCacheEnabled(true);  
2 Bitmap b = view.getDrawingCache();  
3 // Make a copy, then call  
4 // view.destroyDrawingCache()
```

Agenda

- Architecture
- Graphics
- Animations

Animation in Android

- Motivation
- Current Animation Support
 - Animation superclass
 - Animation subclasses
- The Future!

Why Animation?

- Liven up the UI
- Transition between application states
- Keep the user engaged

Animation Superclass

- Timing: duration, startDelay
- Repetition: repeatMode, repeatCount
- Easing: Interpolator
- End state: fillAfter, fillBefore

Animation Types

- Transforming
 - Translation, Rotation, Scale
- Fading
- Sequences
- Cross-fading
- Layout

Transform Animations

- TranslateAnimation
- RotateAnimation
- ScaleAnimation
- Changes rendering matrix of View
 - But not the object's *real* matrix

```
<translate  
    android:fromYDelta="0"  
    android:toYDelta="100%"  
    android:duration="200"/>
```

Fading

- FadeAnimation
- Changes translucency of View's rendering
 - Not the translucency of the object itself

```
<alpha  
    android:fromAlpha="1"  
    android:toAlpha="0"  
    android:duration="200"/>
```


Sequences

- AnimationSet
- Plays child animations together
 - Use startDelay on children to stagger start times

```
<set>  
  <translate ... /  
  <alpha ... />  
</set>
```

Crossfading

- TransitionDrawable
- Crossfades between multiple Drawables
 - startTransition(duration): crossfades to top drawable
 - reverseTransition(duration): crossfades to bottom

```
<transition>  
  <item android:drawable="@drawable/start" />  
  <item android:drawable="@drawable/end" />  
</transition>
```

Layout Animations

- Single animation applied to layout's children
- Start times are staggered

res/anim/layout_fade:

```
<gridLayoutAnimation  
  android:columnDelay="50%"  
  android:directionPriority="row"  
  android:direction="right_to_left|bottom_to_top"  
  android:animation="@anim/fade" />
```

```
<GridView android:layoutAnimation="@anim/layout_fade"  
  android:layout_width="fill_parent"  
  android:layout_height="fill_parent"/>
```

Performance Tips

- enable drawing cache
 - `setDrawingCacheEnabled(true);`
- layout animations
 - `ViewGroup: animationCache = true`

The Future!

- Problem:
 - Current animations handle only specific View actions
 - Not easy to animate anything else
 - Drawable's alpha property
 - Paint's color property
 - Custom object's properties
 - Animated views haven't actually been altered
 - Just drawn in a different location/orientation
- Solution:
 - Property animation system
 - “Animate ‘x’ on Foo”
 - Detects and calls set/get functions

The Future

Disclaimer: This API example does not represent a commitment to any particular implementation or interface in any future release. If there were such a release, which is not guaranteed.

```
<objectAnimator  
  android:propertyName="x"  
  android:valueFrom="0"  
  android:valueTo="100"/>
```

```
ObjectAnimator anim = new ObjectAnimator(shape,  
    "alpha", 0, 100);  
anim.start();
```

For More Information

- Android developer site
 - developer.android.com
- Romain
 - [@romainguy](https://twitter.com/romainguy)
 - curious-creature.org
- Chet
 - [@chethaase](https://twitter.com/chethaase)
 - graphics-geek.blogspot.com

