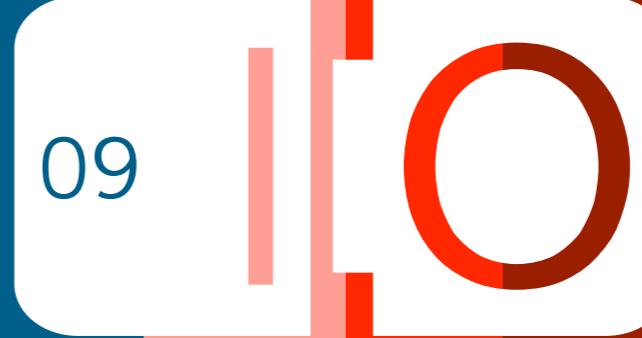
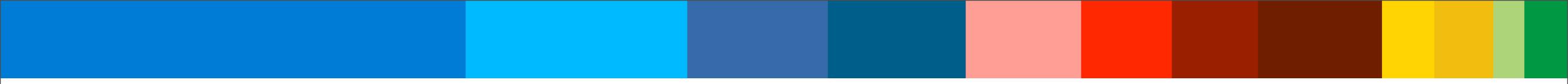


Google™







Debugging Arts of the Ninja Masters

Justin Mattson
Developer Advocate
Android Team @ Google
28/5/2009



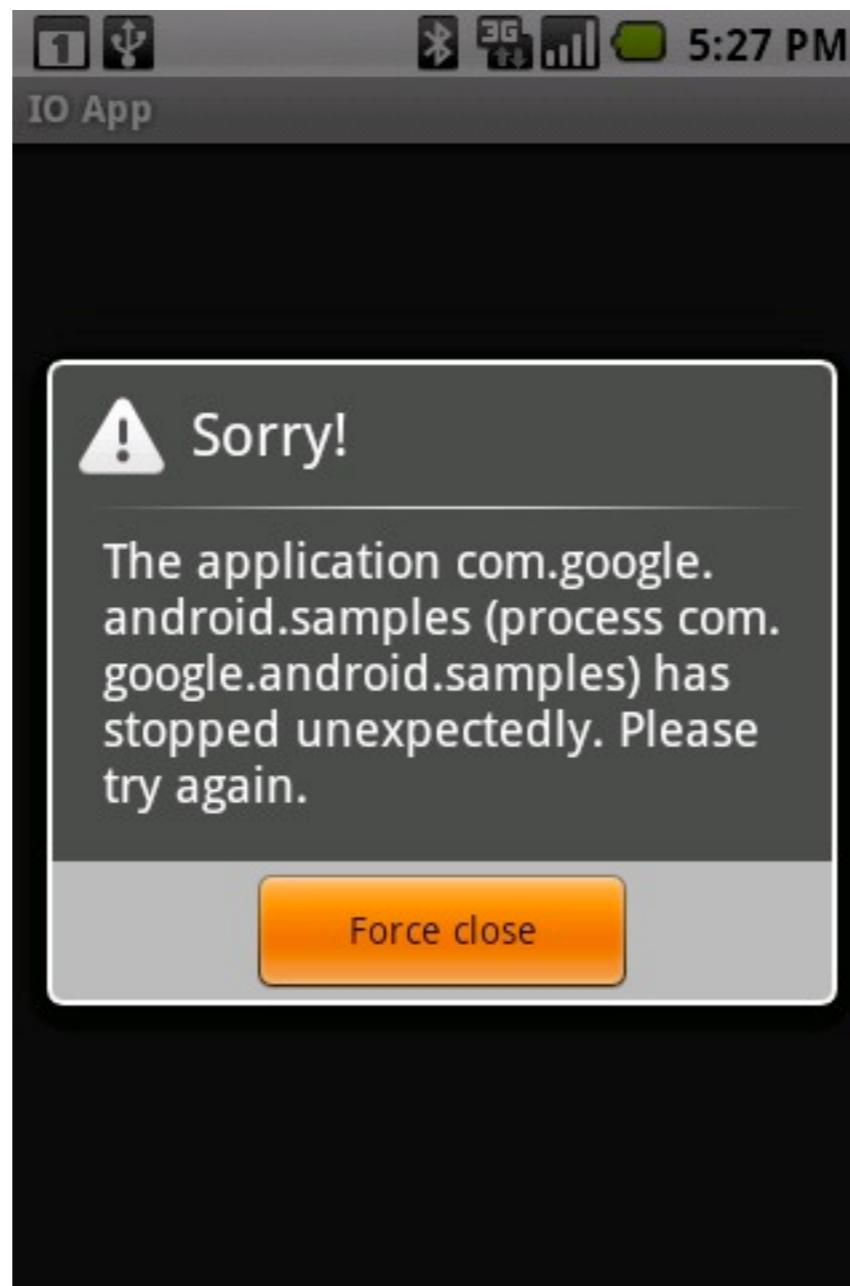
Agenda

- Tool tour
 - logcat
 - traceview
 - hierarchyviewer
- Real world usage
- Pop quiz!
- Q&A (pop quiz for me)



logcat

The dialog of death



Forensics

Where to begin the autopsy

```
E/AndroidRuntime( 1401): java.lang.RuntimeException: Unable to start activity ComponentInfo{com.google.android.samples/com.google.android.samples.tasks.IoSampleApp}: java.lang.NullPointerException: uriString
E/AndroidRuntime( 1401):         at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2337)
E/AndroidRuntime( 1401):         at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2353)
E/AndroidRuntime( 1401):         at android.app.ActivityThread.access$1800(ActivityThread.java:112)
E/AndroidRuntime( 1401):         at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1758)
E/AndroidRuntime( 1401):         at android.os.Handler.dispatchMessage(Handler.java:99)
E/AndroidRuntime( 1401):         at android.os.Looper.loop(Looper.java:123)
E/AndroidRuntime( 1401):         at android.app.ActivityThread.main(ActivityThread.java:4021)
E/AndroidRuntime( 1401):         at java.lang.reflect.Method.invokeNative(Native Method)
E/AndroidRuntime( 1401):         at java.lang.reflect.Method.invoke(Method.java:521)
E/AndroidRuntime( 1401):         at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:782)
E/AndroidRuntime( 1401):         at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:540)
E/AndroidRuntime( 1401):         at dalvik.system.NativeStart.main(Native Method)
E/AndroidRuntime( 1401): Caused by: java.lang.NullPointerException: uriString
E/AndroidRuntime( 1401):         at android.net.Uri$StringUri.<init>(Uri.java:417)
E/AndroidRuntime( 1401):         at android.net.Uri$StringUri.<init>(Uri.java:407)
E/AndroidRuntime( 1401):         at android.net.Uri.parse(Uri.java:379)
E/AndroidRuntime( 1401):         at com.google.android.samples.tasks.IoSampleApp.syncWithServer(IoSampleApp.java:27)
E/AndroidRuntime( 1401):         at com.google.android.samples.tasks.IoSampleApp.checkStatus(IoSampleApp.java:23)
E/AndroidRuntime( 1401):         at com.google.android.samples.tasks.IoSampleApp.initUi(IoSampleApp.java:19)
E/AndroidRuntime( 1401):         at com.google.android.samples.tasks.IoSampleApp.onCreate(IoSampleApp.java:15)
E/AndroidRuntime( 1401):         at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1123)
E/AndroidRuntime( 1401):         at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2300)
E/AndroidRuntime( 1401):         ... 11 more
I/Process ( 245): Sending signal. PID: 1401 SIG: 3
```

logcat

Your news feed

```
D/dalvikvm( 245): GC freed 2534 objects / 115912 bytes in 283ms
I/ActivityManager( 245): Starting activity: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] flg=0x2800000 cmp=com.google.android.gm/.ConversationListActivity }
V/Gmail   ( 498): onCreate(), id=com.google.android.gm.ConversationListActivity@435dea50
D/gmail-ls( 339): MailProvider.query: content://gmail-ls/labels/jmatt@google.com(null, null)
D/gmail-ls( 339): MailProvider.query: content://gmail-ls/conversations/jmatt@google.com(, null)
V/Gmail   ( 498): onStart(), id=com.google.android.gm.ConversationListActivity@435dea50
V/Gmail   ( 498): onResume()
I/ActivityManager( 245): Displayed activity com.google.android.gm/.ConversationListActivity: 794 ms (total 1039 ms)
D/gmail-ls( 339): clearNewUnreadMailForNotificationLabelIfNeeded. Count: 51, label: -5/^unseen-^i
D/gmail-ls( 339): sending notification. Adjusted values are label=2, oldCount=26400, newCount=0, unseenCount=0, getAttention=false
I/gmail-ls( 339): Sending notification intent: Intent { act=android.intent.action.PROVIDER_CHANGED data=content://gmail-ls/unread/^i typ=gmail-ls (has extras) }
D/gmail-ls( 339): clearNewUnreadMailForNotificationLabelIfNeeded. Count: 0, label: -6/^unseen-^vimi
I/ActivityManager( 245): Starting activity: Intent { cmp=com.google.android.gm/.HtmlConversationActivity (has extras) }
V/Gmail   ( 498): onSaveInstanceState(), id=com.google.android.gm.ConversationListActivity@435dea50
V/Gmail   ( 498): onPause(), id=com.google.android.gm.ConversationListActivity@435dea50
D/gmail-ls( 339): MailProvider.query: content://gmail-ls/conversations/jmatt@google.com/1301874412399572048/messages(null, null)
D/dalvikvm( 498): GC freed 4045 objects / 206296 bytes in 181ms
I/ActivityManager( 245): Displayed activity com.google.android.gm/.HtmlConversationActivity: 2477 ms (total 2477 ms)
I/ActivityManager( 245): Process android.process.media (pid 1225) has died.
D/dalvikvm( 1012): GC freed 10663 objects / 613816 bytes in 189ms
D/Gmail   ( 498): com.google.android.gm.HtmlConversationActivity@4362bf50 onKeyDown() keyCode:4 event: KeyEvent{action=0 code=4 repeat=0 meta=0 scanCode=158 mFlags=8}
D/gmail-ls( 339): MailProvider.delete(): removing label ^u from conversation 1301874412399572048
D/Gmail   ( 498): com.google.android.gm.HtmlConversationActivity@4362bf50 onPause() finished
V/Gmail   ( 498): onRestart(), id=com.google.android.gm.ConversationListActivity@435dea50
V/Gmail   ( 498): onStart(), id=com.google.android.gm.ConversationListActivity@435dea50
V/Gmail   ( 498): onResume()
D/gmail-ls( 339): clearNewUnreadMailForNotificationLabelIfNeeded. Count: 0, label: -5/^unseen-^i
D/gmail-ls( 339): clearNewUnreadMailForNotificationLabelIfNeeded. Count: 0, label: -6/^unseen-^vimi
```

logcat Levels Defined

- [E]rror
 - A unexpected critical or non-recoverable failure happened
- [W]arning
 - Something bad happened, but it was handled gracefully
- [I]nfo
 - An important event occurred
- [D]ebug
 - Something happened that may be useful in isolating a problem
- [V]erbose
 - Something occurred in the normal course of operation that was expected.

A simple case

The code can't break!

```
public class UriReader extends Thread {  
    public String uriToRead = null;  
    @Override  
    public void run() {  
        DefaultHttpClient client = new DefaultHttpClient();  
        HttpGet request = new HttpGet(uriToRead);  
        try {  
            InputStream reader =  
                client.execute(request).getEntity().getContent();  
            while (reader.read() != -1);  
        } catch (ClientProtocolException e) {  
            Log.w(TAG, "Protocol exception while reading URL.");  
        } catch (IOException e) {  
            Log.e(TAG, "General I/O exception, exact type is: " +  
                e.getClass().getCanonicalName());  
        }  
    }  
}
```

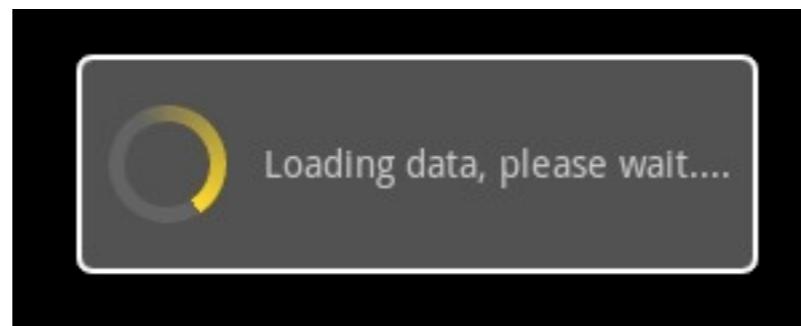
The scene of the crime

```
reuse 7200 seconds
D/SettingsWifiEnabler( 1461): Received network state changed to NetworkInfo: type: WIFI[], state: CONNECTED/CONNECTED, reason: (unspecified), extra: (none), roaming: false, failover: false, isAvailable: true
D/WifiWatchdogService( 245): (android.server.ServerThread) GoogleGuest (00:0b:86:eb:f0:60) requires the watchdog
I/ActivityManager( 245): Start proc com.android.mms for broadcast com.android.mms/.transaction.MmsSystemEventReceiver: pid=1720 uid=10013 gids={3003}
D/GpsLocationProvider( 245): state: DISCONNECTED apnName: epc.tmobile.com reason: dataDisabled
I/ActivityManager( 245): processNextBroadcast: waiting for ProcessRecord{4388aa98 1720: com.android.mms/10013}
D/GpsLocationProvider( 245): updateNetworkState available
D/GpsLocationProvider( 245): NetworkThread wait for 13551489ms
W/Smack/Packet( 339): notify conn break (IOEx), close connection
D/Smack ( 339): [XMPPConn] close connection, notifyClosed=false
I/jdwp ( 1720): received file descriptor 20 from ADB
06:36 I/ddms: Created: [Debugger 8604-->1720 inactive]
I/jdwp ( 1720): Ignoring second debugger -- accepting and dropping
D/SamplesUriReader( 1672): General I/O exception, exact type is: java.net.SocketException
D/WifiWatchdogService( 245): 0% packet loss (acceptable is 25%)
D/WifiWatchdogService( 245): (WifiWatchdogThread) AP is alive: GoogleGuest (00:0b:86:eb:f0:60)
```

Take a good look around

```
reuse 7200 seconds
D/SettingsWifiEnabler( 1461): Received network state changed to NetworkInfo: type: WIFI[], state: CONNECTED/CONNECTED, reason: (unspecified), extra: (none), roaming: false, failover: false, isAvailable: true
D/WifiWatchdogService( 245): (android.server.ServerThread) GoogleGuest (00:0b:86:eb:f0:60) requires the watchdog
I/ActivityManager( 245): Start proc com.android.mms for broadcast com.android.mms/.transaction.MmsSystemEventReceiver: pid=1720 uid=10013 gids={3003}
D/GpsLocationProvider( 245): state: DISCONNECTED apnName: epc.tmobile.com reason: dataDisabled
I/ActivityManager( 245): processNextBroadcast: waiting for ProcessRecord{4388aa98 1720: com.android.mms/10013}
D/GpsLocationProvider( 245): updateNetworkState available
D/GpsLocationProvider( 245): NetworkThread wait for 13551489ms
W/Smack/Packet( 339): notify conn break (IOEx), close connection
D/Smack ( 339): [XMPPConn] close connection, notifyClosed=false
I/jdwp ( 1720): received file descriptor 20 from ADB
06:36 I/ddms: Created: [Debugger 8604-->1720 inactive]
I/jdwp ( 1720): Ignoring second debugger -- accepting and dropping
D/SamplesUriReader( 1672): General I/O exception, exact type is: java.net.SocketException
D/WifiWatchdogService( 245): 0% packet loss (acceptable is 25%)
D/WifiWatchdogService( 245): (WifiWatchdogThread) AP is alive: GoogleGuest (00:0b:86:eb:f0:60)
```

The dialog of despair



Warning signs

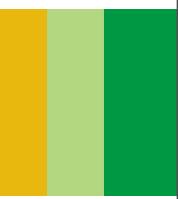
What's that noise?

- Same process id, similar object count, similar sizes

```
D/dalvikvm( 1825): GC freed 7 objects / 313984 bytes in 121ms
D/dalvikvm( 1825): GC freed 6 objects / 316016 bytes in 115ms
D/dalvikvm( 1825): GC freed 7 objects / 318080 bytes in 120ms
D/dalvikvm( 1825): GC freed 6 objects / 320112 bytes in 123ms
D/dalvikvm( 1825): GC freed 7 objects / 322176 bytes in 122ms
D/dalvikvm( 1825): GC freed 7 objects / 324312 bytes in 120ms
D/dalvikvm( 1825): GC freed 20 objects / 326920 bytes in 115ms
D/dalvikvm( 1825): GC freed 6 objects / 328272 bytes in 120ms
D/dalvikvm( 1825): GC freed 7 objects / 330336 bytes in 118ms
D/dalvikvm( 1825): GC freed 6 objects / 332368 bytes in 116ms
D/dalvikvm( 1825): GC freed 7 objects / 334432 bytes in 119ms
D/dalvikvm( 1825): GC freed 6 objects / 336464 bytes in 115ms
D/dalvikvm( 1825): GC freed 7 objects / 338528 bytes in 119ms
D/dalvikvm( 1825): GC freed 6 objects / 340560 bytes in 120ms
D/dalvikvm( 1825): GC freed 7 objects / 342624 bytes in 114ms
D/dalvikvm( 1825): GC freed 6 objects / 344656 bytes in 120ms
D/dalvikvm( 1825): GC freed 7 objects / 346720 bytes in 115ms
D/dalvikvm( 1825): GC freed 6 objects / 348752 bytes in 138ms
```

Data Management

- Filters
 - TAG:SEVERITY
- Smart tagging
 - Meaningful
 - Related
- !SPAM
 - Precise
 - Concise
 - Privacy sensitive



traceview

tracing and traceview

- Records every function entry and exit point
- Records how long execution took
- Provides a graphical representation of collected data
 - Timeline view
 - Call tree view
- Is a must-use for any developer serious about performance

How to start tracing

- Starting and stopping trace data collection is simple

```
public class ProfiledActivity extends Activity {  
    protected void onCreate(Bundle savedInstanceState) {  
        Debug.startMethodTracing();  
    }  
    ...  
    public void onPause() {  
        Debug.stopMethodTracing();  
    }  
}
```

```
justin$ adb shell am profile com.example.foo profile start /  
sdcard/trace_file  
...  
justin$ adb shell am profile com.example.foo stop
```

- Controls tracing for entire VM
- Tracing only what you need simplifies analysis
- Tracing has a big impact on performance

Execution timeline



Function statistics

Name	Incl %	Inclusive	Excl %	Exclusive	Calls+Recur	Calls/T	Time/Call
► 0 (toplevel)	100.0%	19761.933	0.8%	155.751	16+0	1235.121	
► 1 java/lang/Thread.run ()V	35.7%	7048.340	0.0%	0.092	2+0	3524.170	
► 2 java/util/concurrent/ThreadPoolExecutor\$Worker.run ()V	35.7%	7048.248	0.0%	0.184	2+0	3524.124	
► 3 java/util/concurrent/ThreadPoolExecutor\$Worker.runTask ()V	35.6%	7042.510	0.0%	0.488	4+0	1760.628	
► 4 com/google/android/apps/finance/CachedNetworkFetch	35.6%	7040.773	0.0%	1.161	4+0	1760.193	
► 5 android/os/Handler.dispatchMessage (Landroid/os/Mes	28.9%	5701.541	0.0%	2.992	130+0	43.858	
► 6 com/google/android/apps/finance/data/NewsRunnable	26.9%	5308.686	0.0%	0.092	2+0	2654.343	
► 7 com/google/android/apps/finance/data/NewsRunnable	26.9%	5308.594	0.0%	0.031	2+0	2654.297	
► 8 com/google/android/apps/finance/data/NewsManager.	26.9%	5308.563	0.0%	7.655	2+0	2654.282	
► 9 android/view/ViewRoot.handleMessage (Landroid/os/Mes	20.8%	4112.275	0.0%	3.574	46+0	89.397	
► 10 org/json/JSONObject.<init> (Ljava/lang/String;)V	19.1%	3773.040	0.0%	0.183	2+0	1886.520	
► 11 org/json/JSONObject.<init> (Lorg/json/JSONTokener;)V	19.1%	3772.491	0.3%	61.596	2+112	33.092	
► 12 org/json/JSONTokener.nextValue ()Ljava/lang/Object;	19.1%	3770.417	0.8%	157.097	4+904	4.152	
► 13 org/json/JSONArray.<init> (Lorg/json/JSONTokener;)V	19.1%	3766.174	0.1%	11.908	2+16	209.232	
► 14 android/view/ViewRoot.performTraversals ()V	18.9%	3736.450	0.0%	6.502	18+0	207.581	
► 15 android/os/Binder.execTransact (IIII)Z	18.2%	3605.227	0.1%	19.992	198+0	18.208	
► 16 android/content/ContentProviderNative.onTransact (ILa	17.0%	3354.797	0.0%	7.353	30+0	111.827	
► 17 org/json/JSONTokener.nextString (C)Ljava/lang/String;	14.1%	2795.441	2.7%	526.857	324+0	8.628	
► 18 android/view/View.layout (IIII)V	11.5%	2278.992	0.1%	10.780	13+362	6.077	
► 19 android/widget/FrameLayout.onLayout (ZIIII)V	11.5%	2277.437	0.0%	7.659	13+46	38.601	
► 20 android/widget/LinearLayout.onLayout (ZIIII)V	11.5%	2273.438	0.0%	1.865	13+93	21.448	
► 21 android/widget/ViewRootImpl.dispatchLayoutInclusive (ZV	11.5%	2272.222	0.1%	18.541	12+60	21.140	

Function statistics

Details

Name	Incl %	Inclusive	Excl %	Exclusive	Calls+Recur Calls/Total	SORT	Time/Call
1 java/lang/Thread.run 0V	35.7%	7048.340	0.0%	0.092	2+0	3524.170	
2 java/util/concurrent/ThreadPoolExecutor\$Work	35.7%	7048.248	0.0%	0.184	2+0	3524.124	

- Name
- “Incl %” - percentage of time including descendants
- Inclusive - raw execution time including descendants
- “Excl %” - percentage of time sans descendants
- Exclusive - time spent executing code only within this function
- “Calls+Recur Calls/Total” - Number of times this method is called
- Time/Call - Average execution time per call

Getting to the bottom (or top) of things

- The source of the issue may be above or below your code
- traceview allows you to easily navigate the call tree

Getting to the bottom (or top) of things

- The source of the issue may be above or below your code
- traceview allows you to easily navigate the call tree

Name	Incl %	Inclusive	Excluded
122 com/google/android/apps/finance/NetworkUtils.fetchUrl (Ljava/lang/String;)Ljava/lang/String;	2.9%	570.984	
123 java/lang/AbstractStringBuilder.append0 (Ljava/lang/String;)V	2.9%	568.250	
124 android/database/sqlite/SQLiteDatabase.rawQueryWithFactory (Landroid/database/sqlite/SQLiteDatabase;Ljava/lang/String;Landroid/net/Uri;[Ljava/lang/String;Ljava/lang/String;)V	2.8%	553.196	
125 android/content/ContentProvider\$Transport.query (Landroid/net/Uri;[Ljava/lang/String;Ljava/lang/String;)Landroid/os/Parcel	2.8%	551.144	
126 org/apache/harmony/luni/internal/net/www/protocol/http/HttpURLConnection.readIn ()Ljava/lang/String;Landroid/os/Parcel	2.8%	545.346	
127 android/database/sqlite/SQLiteDatabase\$CursorDriver.query (Landroid/database/sqlite/SQLiteDatabase;Landroid/os/Parcel;Ljava/lang/String;Ljava/lang/String;)Landroid/os/Parcel	2.7%	543.124	
128 android/content/ContentValues\$1.createFromParcel (Landroid/os/Parcel;)Ljava/lang/Object	2.7%	540.862	
129 android/content/ContentValues\$1.createFromParcel (Landroid/os/Parcel;)Landroid/content/ContentValues	2.7%	540.066	
130 dalvik/system/VMDebug.startGC ()V	2.7%	539.948	
131 android/widget/LinearLayout.measureHorizontal (II)V	2.7%	535.739	
132 android/os/Parcel.readHashMap (Ljava/lang/ClassLoader;)Ljava/util/HashMap	2.7%	535.639	
133 android/widget/CursorTreeAdapter.getChildrenCount ()I	2.7%	535.339	
134 android/widget/BaseExpandableListAdapter.notifyDataSetChanged ()V	2.7%	535.066	
135 java/io/ObjectInputStream.readHierarchy (Ljava/lang/Object;Ljava/io/ObjectStreamClass;)V	2.7%	533.631	
136 android/widget/ExpandableListConnector\$MyDataSetObserver.onChanged ()V	2.7%	529.633	
137 java/io/ObjectInputStream.readObjectForClass (Ljava/lang/Object;Ljava/io/ObjectStreamClass;)V	2.7%	529.479	
138 android/widget/ExpandableListView.dispatchDraw (Landroid/graphics/Canvas;)V	2.7%	528.504	
139 android/os/Parcel.readMapInternal (Ljava/util/Map;ILjava/lang/ClassLoader;)V	2.7%	528.164	
140 android/widget/LinearLayout.measureVertical (II)V	2.7%	527.069	
141 java/util/ArrayList.readObject (Ljava/io/ObjectInputStream;)V	2.7%	526.581	
142 com/google/android/apps/finance/HomeActivity\$8.bindChildView (Landroid/view/View;Landroid/child/BindViewData;)V	2.7%	525.294	

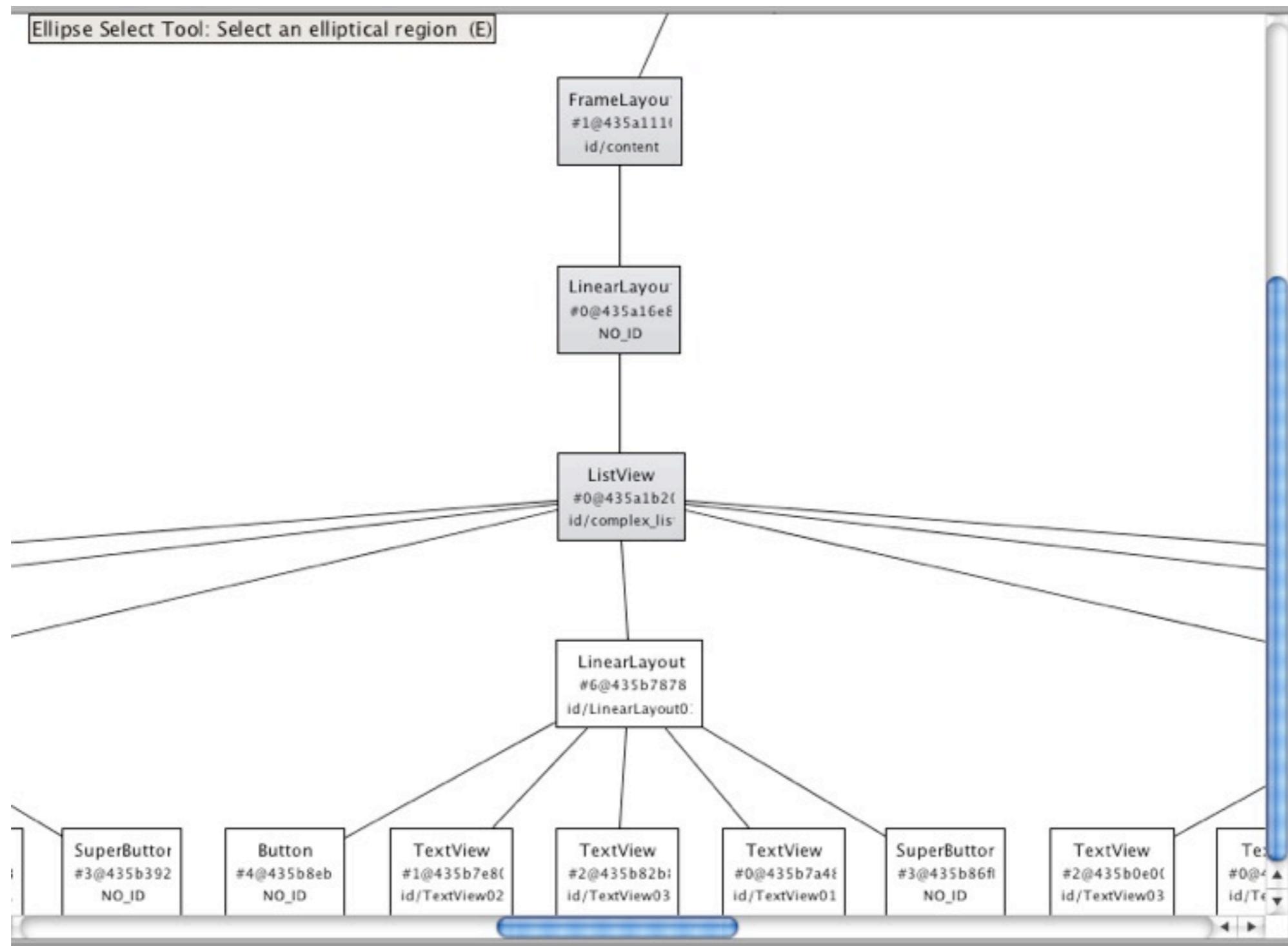
Optimize reality

- traceview gives you a lot of power, take advantage of it
- Design sensibly, build proof of concept, test, and optimize



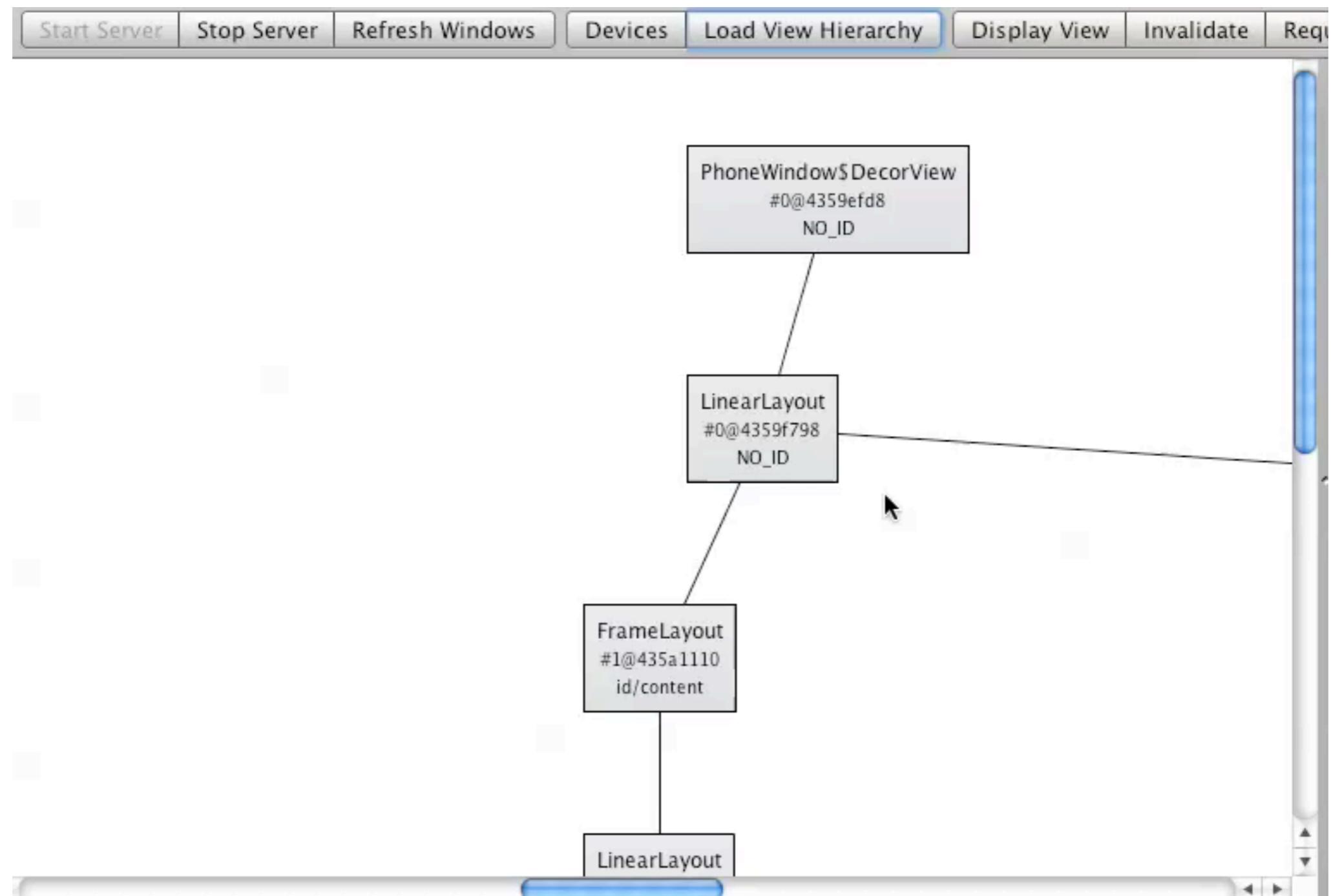
HierarchyViewer

Exploring the UI with HierarchyViewer



Climbing the tree

Climbing the tree



Questionable families

- “Infertile parents” - layout with no or single layout children
- “Clones” - parents with same-type children

<merge /> your @includes

- Merge is a placeholder layout
- Children of <merge> go into the layout where they are included

```
<merge xmlns:android="http://schemas.android.com/apk/res/  
        android">  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/header_section"  
        android:textColor="@colors/header_color" />  
    <ImageView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:src="@drawable/logo" />  
</merge>
```




Traveler's Tales: Betrayal

TraceView on a shipping product



- Why is something running on the main thread for that long?
- Must be a network request someone forgot to move to a thread

com.ibm.icu4jni.util.Resources.getTimeZonesNative

- A call about time zones is taking this long?
- Wait, I'm not even doing anything with time zones

com.ibm.icu4jni.util.Resources.getTimeZonesNative

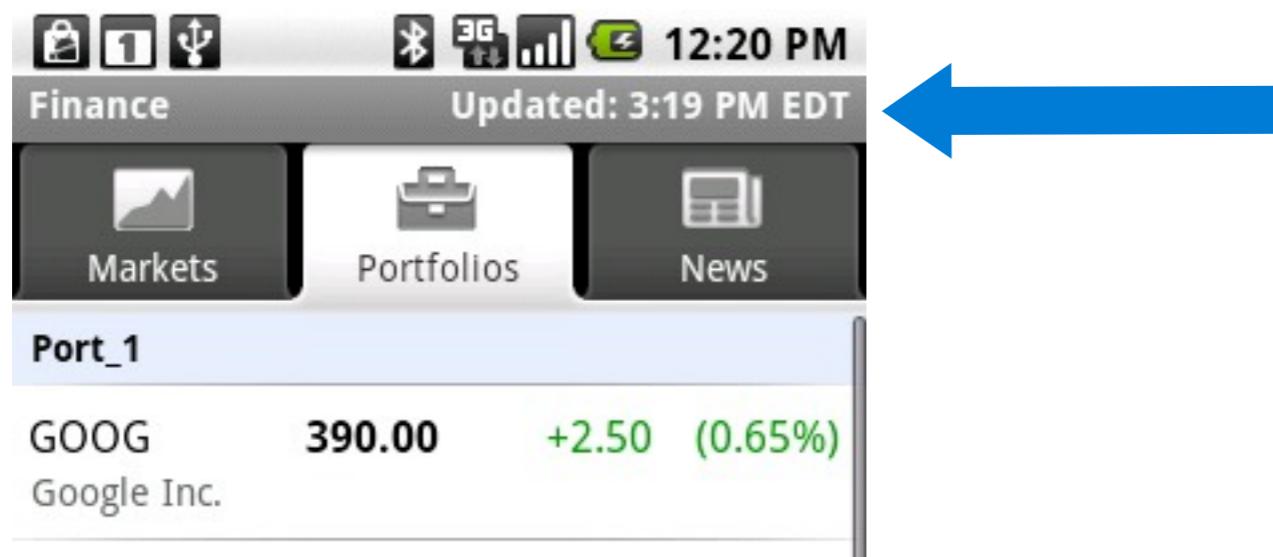
- A call about time zones is taking this long?
- Wait, I'm not even doing anything with time zones

Name	Incl %	Inclusive
28 com.ibm.icu4jni.util.Resources.getTimeZonesNative ([[Ljava/lang/String;Ljava/lang/String;)V	13.7%	2036.472
└ Parents		
27 com.ibm.icu4jni.util.Resources.createTimeZoneNamesFor (Ljava/lang/String;)[[Ljava/lang/Stri	100.0%	2036.472
└ Children		
self (context switch)	100.0%	2036.472
0.0% 0.000		
29 android/view/View.layout (IIII)V	12.0%	1784.042
30 android/widget/FrameLayout.onLayout (ZIIII)V	12.0%	1783.095
31 android/widget/LinearLayout.onLayout (ZIIII)V	12.0%	1781.785
32 android/widget/LinearLayout.layoutVertical ()V	12.0%	1781.662
33 android/widget/LinearLayout.setChildFrame (Landroid/view/View;IIII)V	12.0%	1779.140
34 android/widget/AbsListView.onLayout (ZIIII)V	11.8%	1757.338

com.ibm.icu4jni.util.Resources.getTimeZonesNative

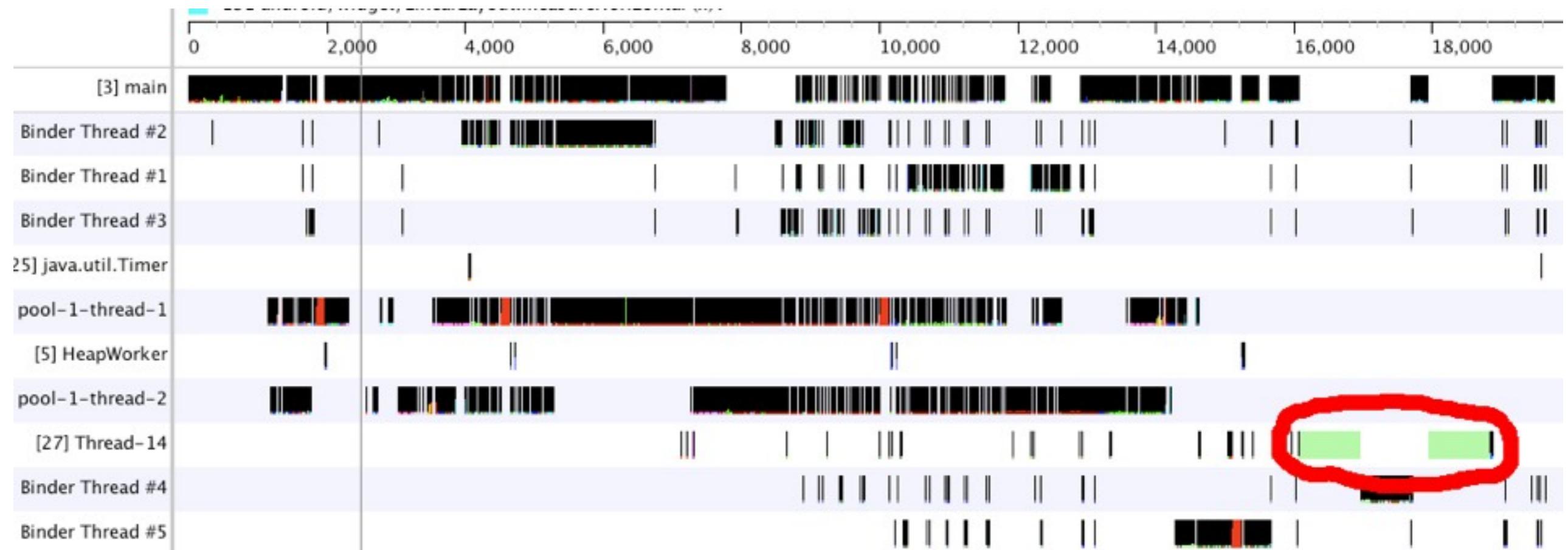
- A call about time zones is taking this long?
- Wait, I'm not even doing anything with time zones

Name	Incl %	Inclusive
28 com.ibm.icu4jni.util.Resources.getTimeZonesNative ([[Ljava/lang/String;Ljava/lang/String;)V	13.7%	2036.472
└ Parents		
27 com.ibm.icu4jni.util.Resources.createTimeZoneNamesFor (Ljava/lang/String;)[[Ljava/lang/Stri	100.0%	2036.472
└ Children		
self (context switch)	100.0%	2036.472
0.0% 0.000		
29 android/view/View.layout (IIII)V	12.0%	1784.042
30 android/widget/FrameLayout.onLayout (ZIIII)V	12.0%	1783.095
31 android/widget/LinearLayout.onLayout (ZIIII)V	12.0%	1781.785
32 android/widget/LinearLayout.layoutVertical ()V	12.0%	1781.662
33 android/widget/LinearLayout.setChildFrame (Landroid/view/View;IIII)V	12.0%	1779.140
34 android/widget/AbsListView.onLayout (ZIIII)V	11.8%	1757.338



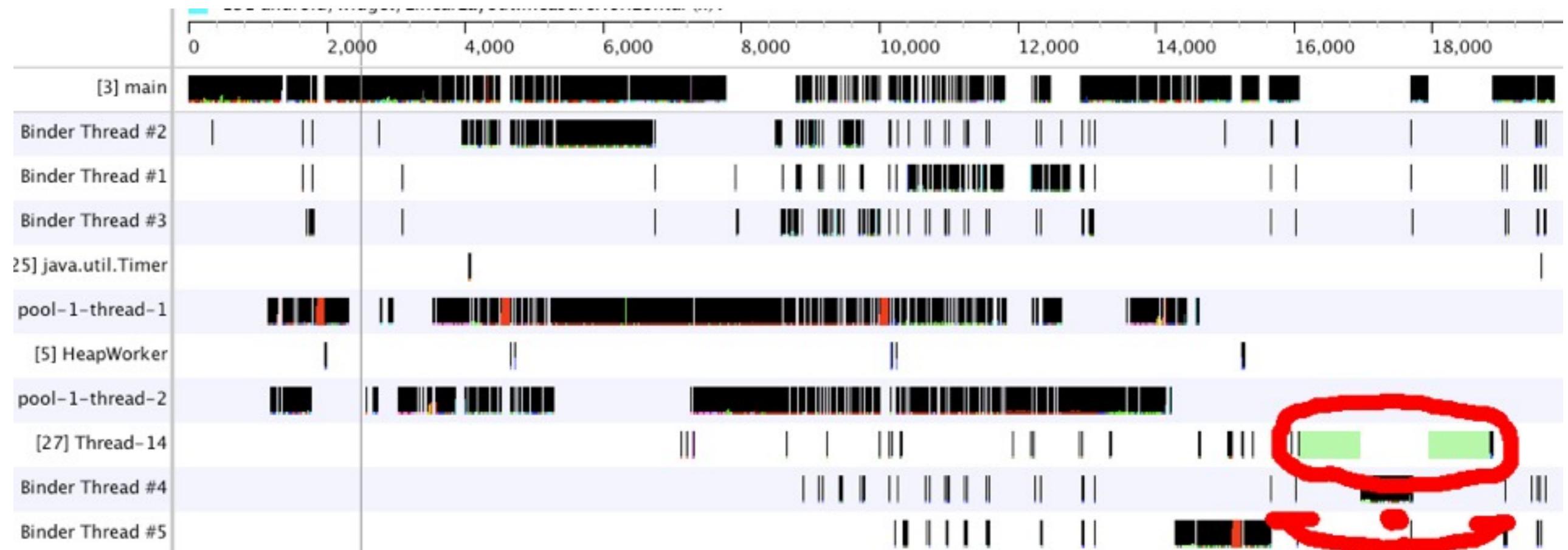
Fixing the problem

- Move the call
- De-prioritize its execution
- Result: Better perceived startup time



Fixing the problem

- Move the call
- De-prioritize its execution
- Result: Better perceived startup time



Key Learnings

- Trust no one
- A little profiling goes a long way
- Priority matters



Traveler's Tales: Finding shortcuts

Search for hidden inefficiencies

- Look at the function listing for less obvious performance issues
- Sort by exclusive execution
 - Shows inside which methods the program spends most of its time
 - Long average time methods with a few calls are good targets
 - Explore reducing and eliminating these calls

Search for hidden inefficiencies

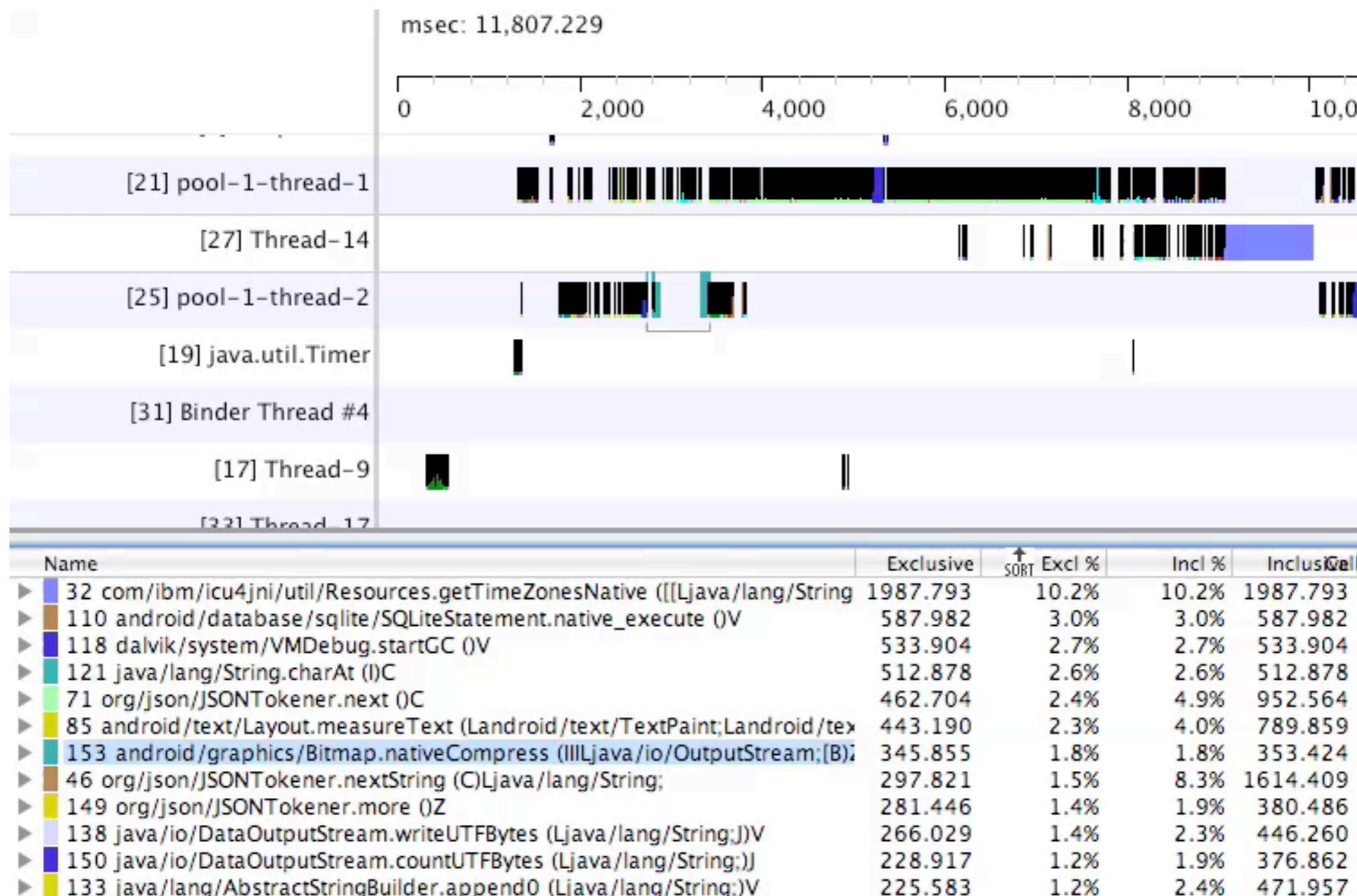
Name	SORT	Excl %	Exclusive	Time/Call
32 com/ibm/icu4jni/util/Resources.getTimeZonesNative ([[Lja	10.2%	1987.793	1987.793	
110 android/database/sqlite/SQLiteDatabase.native_execute	3.0%	587.982	10.138	
118 dalvik/system/VMDebug.startGC ()V	2.7%	533.904	133.476	
121 java/lang/String.charAt (I)C	2.6%	512.878	0.010	
71 org/json/JSONTokener.next ()C	2.4%	462.704	0.084	
85 android/text/Layout.measureText (Landroid/text/TextPair	2.3%	443.190	1.744	
153 android/graphics/Bitmap.nativeCompress (IIILjava/io/Ou	1.8%	345.855	176.712	
46 org/json/JSONTokener.nextString (C)Ljava/lang/String;	1.5%	297.821	9.610	
149 org/json/JSONTokener.more ()Z	1.4%	281.446	0.033	
138 java/io/DataOutputStream.writeUTFBytes (Ljava/lang/Stri	1.4%	266.029	1.924	
150 java/io/DataOutputStream.countUTFBytes (Ljava/lang/Str	1.2%	228.917	1.624	
133 java/lang/AbstractStringBuilder.append0 (Ljava/lang/Stri	1.2%	225.583	0.093	
119 java/lang/StringBuilder.append (C)Ljava/lang/StringBuild	1.1%	212.627	0.051	
154 java/lang/AbstractStringBuilder.append0 (C)V	1.0%	200.260	0.029	
226 java/lang/String.length ()I	1.0%	193.333	0.009	
237 android/graphics/Paint.measureText (Ljava/lang/String;I	0.9%	180.724	0.282	
163 android/text/StaticLayout.generate (Ljava/lang/CharSequ	0.9%	170.793	54.077	
0 (toplevel)	0.7%	142.537	1394.854	

Take a look around

- Explore ancestors and descendants of the method
- Use timeline line view to check what happens before and after the method

Take a look around

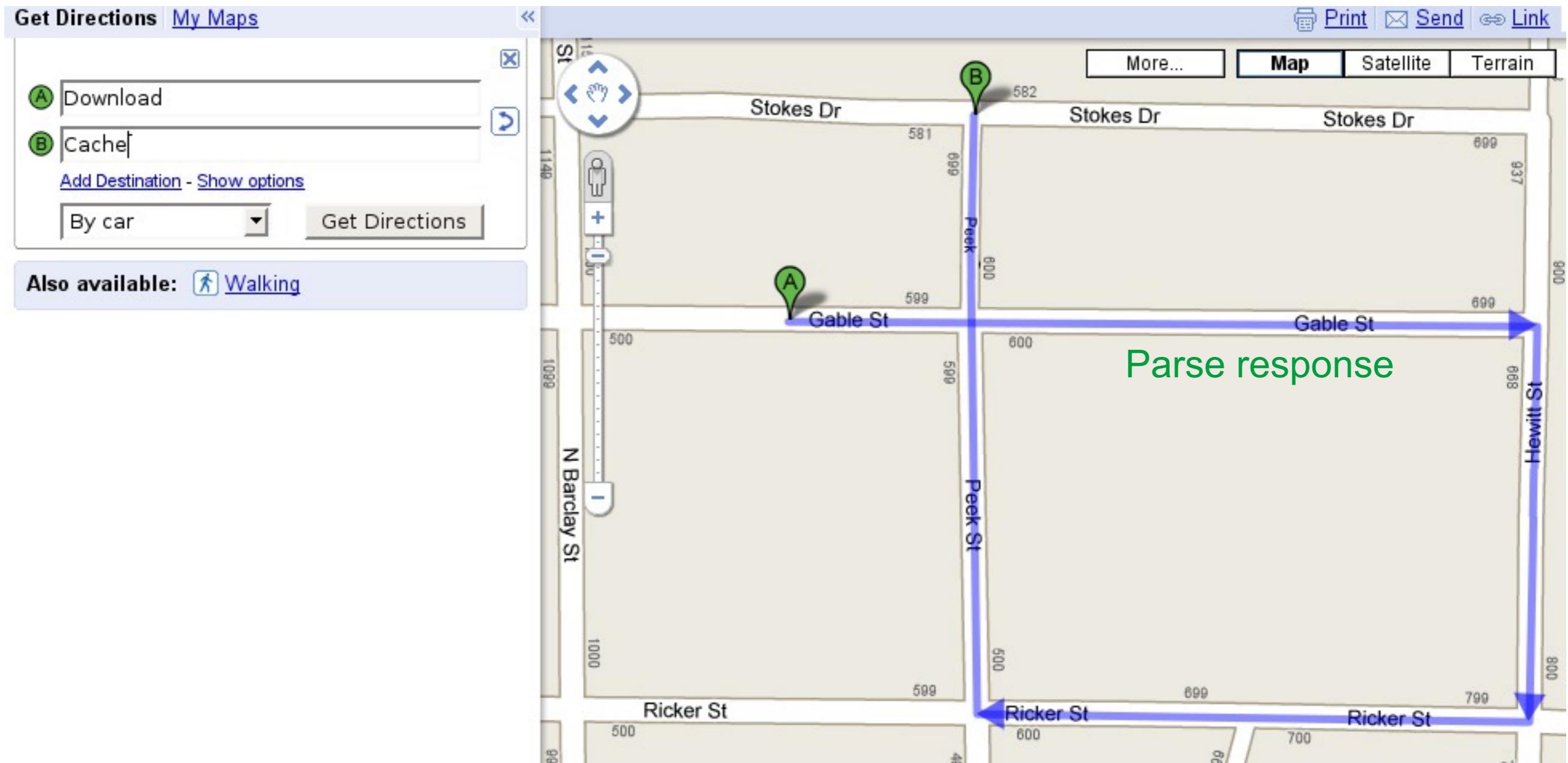
Take a look around



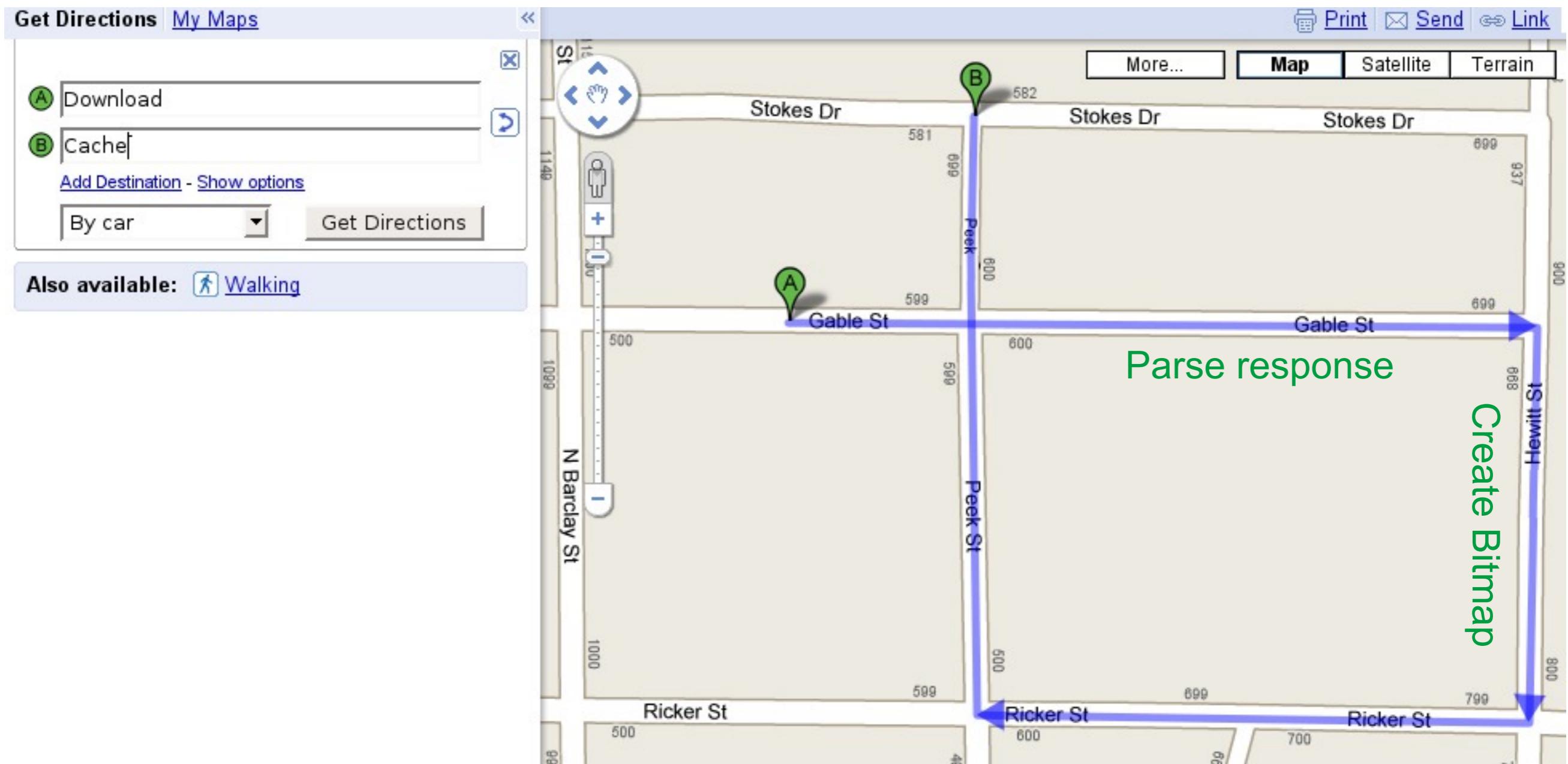
Iterate the route into a circle (square?)

The figure is a screenshot of a Google Maps interface. At the top, there are buttons for 'Get Directions' and 'My Maps'. On the right, there are buttons for 'Print', 'Send', and 'Link'. Below these are buttons for 'More...', 'Map' (which is selected), 'Satellite', and 'Terrain'. On the left, there is a sidebar with a 'Download' button (labeled A) and a 'Cache' button (labeled B). Below these are links for 'Add Destination' and 'Show options'. A dropdown menu shows 'By car' and a 'Get Directions' button. Below this, there is a link for 'Also available: Walking'. The main area shows a street map with a blue route line. Two green markers, labeled A and B, are placed on the map. Marker A is on Gable St at the intersection with N Barclay St. Marker B is on Stokes Dr at the intersection with Peek St. The map also shows other streets like Ricker St, Hewitt St, and Stokes Dr, with house numbers ranging from 500 to 1000. A zoom control with '+' and '-' buttons is located on the left side of the map.

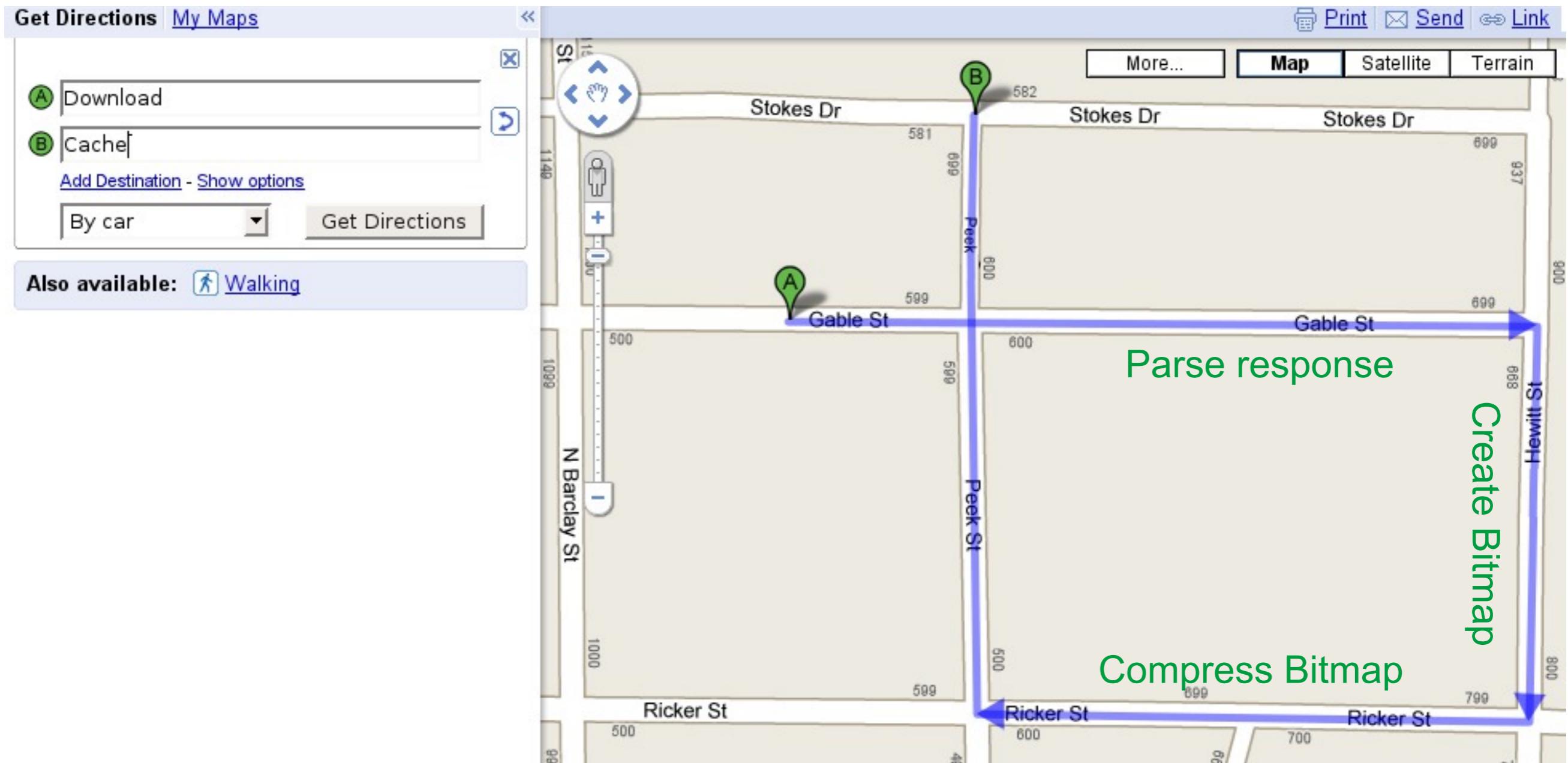
Iterate the route into a circle (square?)



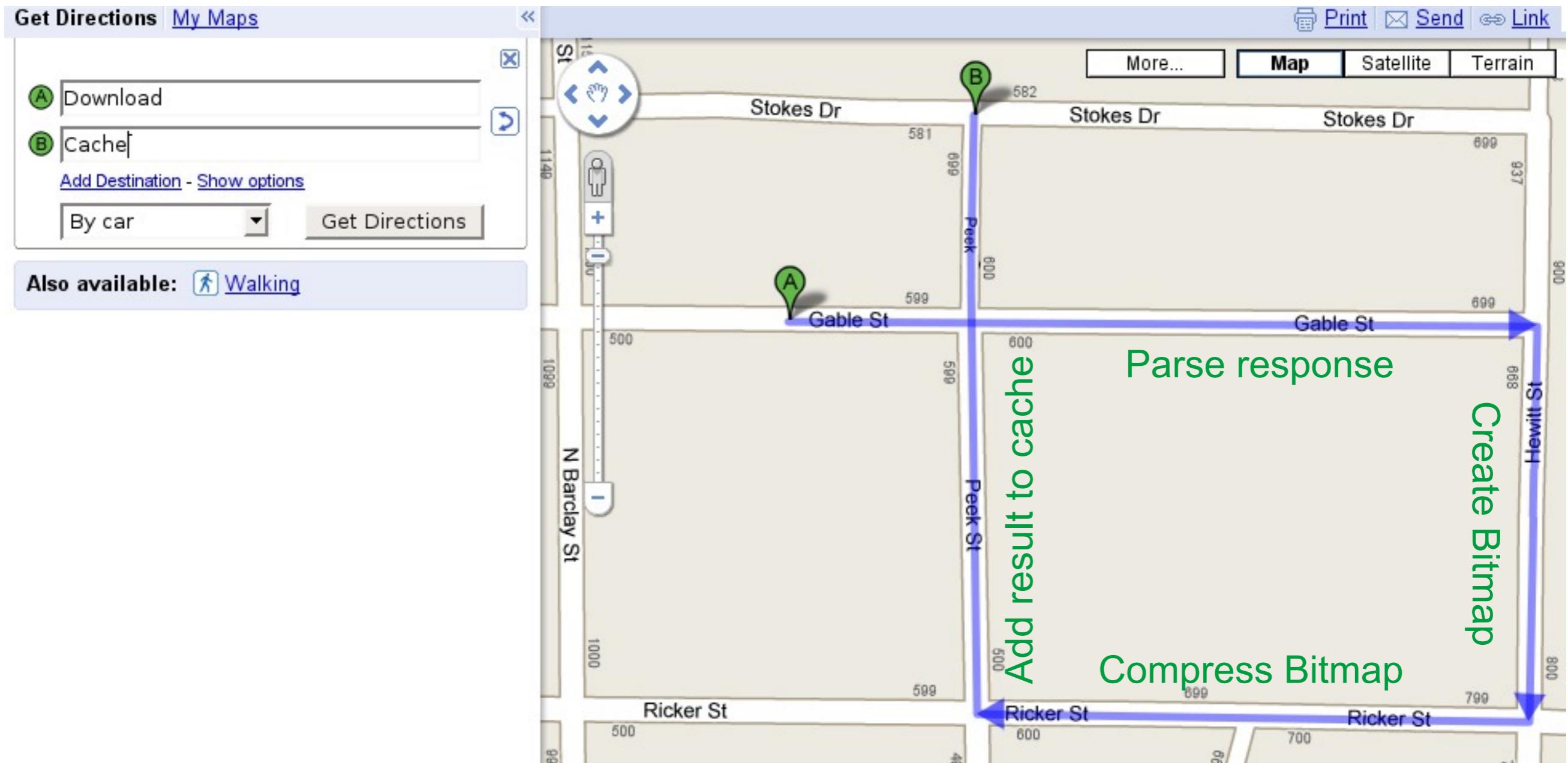
Iterate the route into a circle (square?)



Iterate the route into a circle (square?)



Iterate the route into a circle (square?)

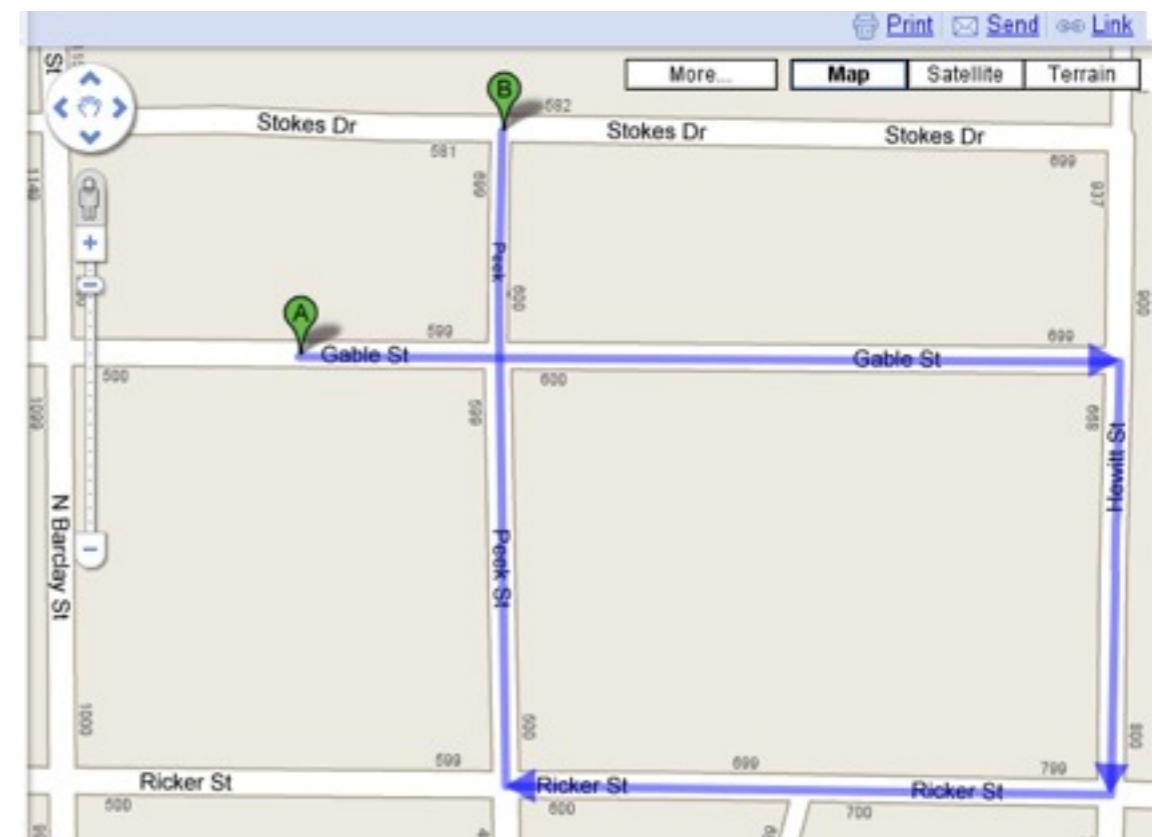


Fixing the problem

- Read in the full response
- Store the PNG
- Inflate Bitmap from PNG data
- Discard PNG data
- Result: Decrease startup time by 350ms
 - 2% in “debug time”, larger real time effect since its native code

Key Learnings

- Big is beautiful
- Explore the area first
- Three rights can make a left





Traveler's Tales: Packing mistakes

Sorting by inclusive execution

- Includes execution time of methods code and its descendants
- List is somewhat like a breadth-first traversal of the call tree

Sorting by inclusive execution

- Includes execution time of methods code and its descendants
- List is somewhat like a breadth-first traversal of the call tree

SORT	Name	Inclusives	Calls + Recur	Calls /
▶	0 (toplevel)	19527.952	14+0	
▶	3191 android/app/Activity\$ManagedCursor.<init> (Landroid/database/Cursor;)V	0.244	18+0	
▶	2101 android/app/Activity.createPendingResult (Landroid/content/Intent;)Landroid/content/PendingResult	1.832	1+0	
▶	191 android/app/Activity.dispatchActivityResult (Ljava/lang/String;Landroid/content/Intent;)V	234.588	1+0	
▶	817 android/app/Activity.findViewById (I)Landroid/view/View;	18.951	11+0	
▶	3894 android/app/Activity.getComponentName ()Landroid/content/ComponentName	0.031	1+0	
▶	1517 android/app/Activity.getSystemService (Ljava/lang/String;)Ljava/lang/Object	4.639	20+0	
▶	3362 android/app/Activity.getWindow ()Landroid/view/Window;	0.182	17+0	
▶	4050 android/app/Activity.isChild ()Z	0.030	1+0	
▶	3386 android/app/Activity.setVisibility (I)V	0.153	1+0	
▶	167 android/app/Activity.managedQuery (Landroid/net/Uri;[Ljava/lang/String;Landroid/widget/CursorFactory;)Landroid/database/Cursor	316.404	18+0	
▶	4051 android/app/Activity.onActivityResult (ILandroid/content/Intent;I)V	0.030	1+0	
▶	4052 android/app/Activity.onPause (I)V	0.030	1+0	
▶	3511 android/app/Activity.onCreate (Landroid/os/Bundle;)V	0.122	1+0	
▶	3669 android/app/Activity.onResume (I)V	0.091	1+0	
▶	4053 android/app/Activity.onResume (I)V	0.030	2+0	
▶	3768 android/app/Activity.onTitleChanged (Ljava/lang/CharSequence;I)V	0.061	1+0	
▶	3731 android/app/Activity.onWindowAttributesChanged (Landroid/view/WindowAttributes;)V	0.062	2+0	
▶	670 android/app/Activity.performPause (I)V	27.405	1+0	
▶	3769 android/app/Activity.performRestart (I)V	0.061	1+0	

Taking a look at the code

```
try {
    List<MyDataType> itemsProcessed = new ArrayList<MyDataType>();
    for (int i = 0; i < source.getNumItems(); i++) {
        Item currentItem = source.get(i);
        MyDataType processedItem = new MyDataType(currentItem);
        itemsProcessed.add(processedItem);
        itemProcessedCallback(itemsProcessed);
    }
} catch (IOException e) {
    Log.W(TAG, "Reading data failed, retrying");
} catch (JSONException e) {
    Log.E(TAG, "JSON parsing failed");
}
```

- Items will be processed several times, this seems wrong

Fixing the problem

- Bug introduced when exception handling was refactored

```
try {  
    List<MyDataType> itemsProcessed = new ArrayList<MyDataType>();  
    for (int i = 0; i < source.getNumItems(); i++) {  
        Item currentItem = source.get(i);  
        MyDataType processedItem = new MyDataType(currentItem);  
        itemsProcessed.add(processedItem);  
    }  
    itemProcessedCallback(itemsProcessed);  
} catch (IOException e) {  
    Log.W(TAG, "Reading data failed, retrying");  
} catch (JSONException e) {  
    Log.E(TAG, "JSON parsing failed");  
}
```

- Result: Fewer calls to callback, 5% decrease in startup time

Key Learnings

- Each angle of approach is valuable
- Analyzing performance can reveal unknown bugs

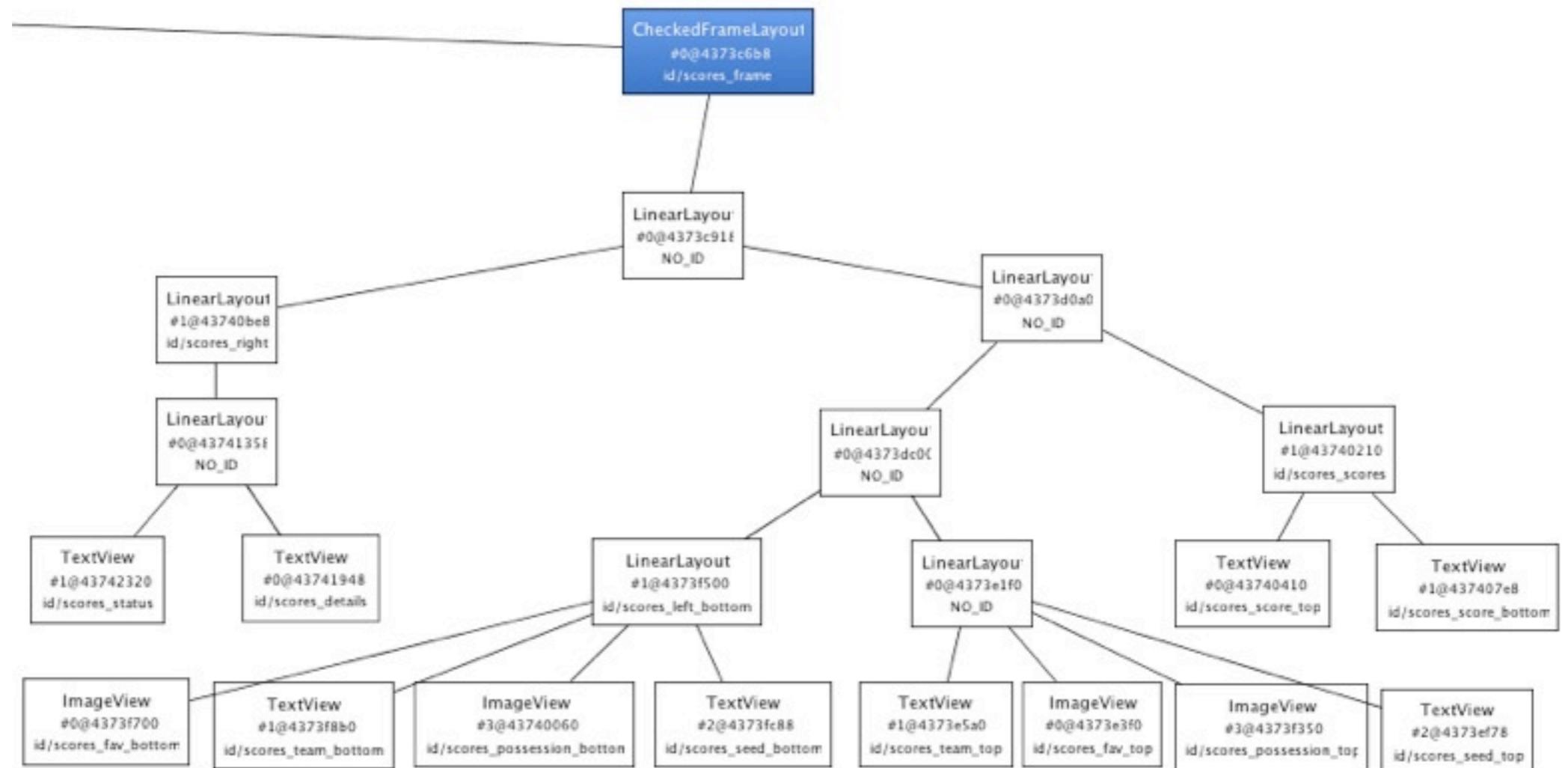
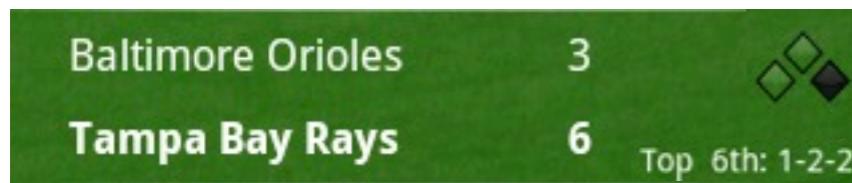


Traveler's Tales: Consolidating containers

A “simple” layout examined



A “simple” layout examined



Fixing the Problem



Baltimore Orioles

3



Tampa Bay Rays

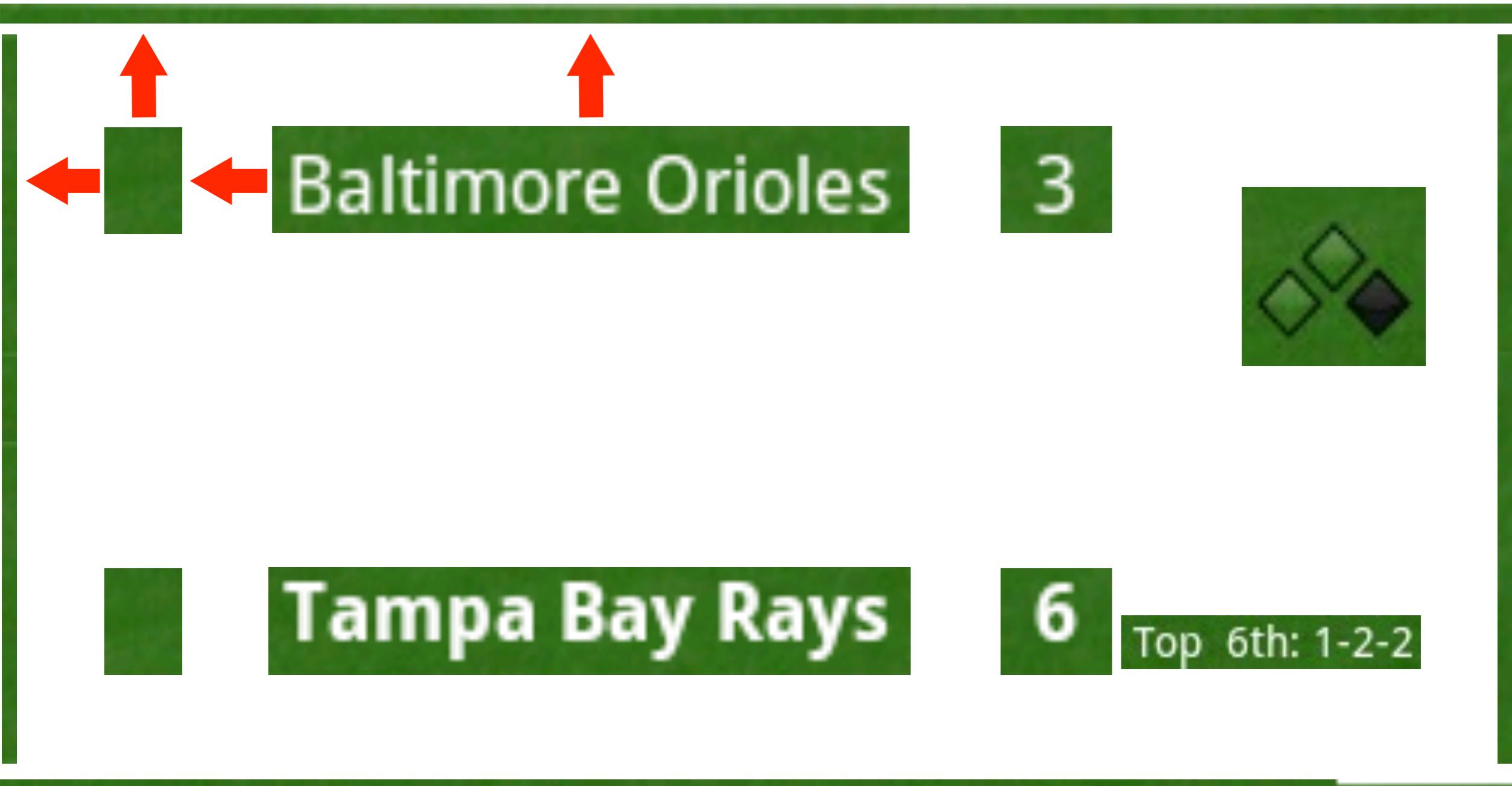
6

Top 6th: 1-2-2

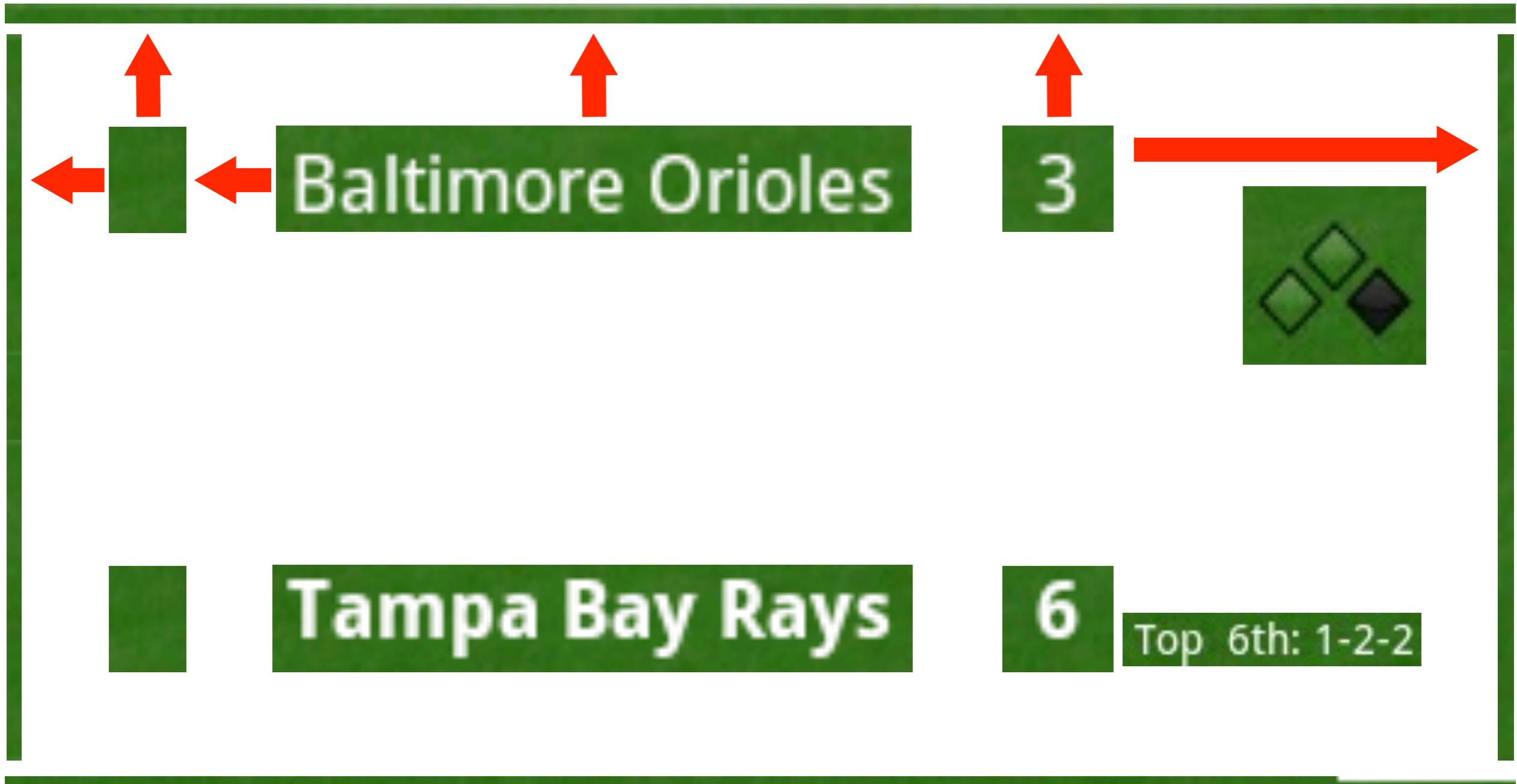
Fixing the Problem



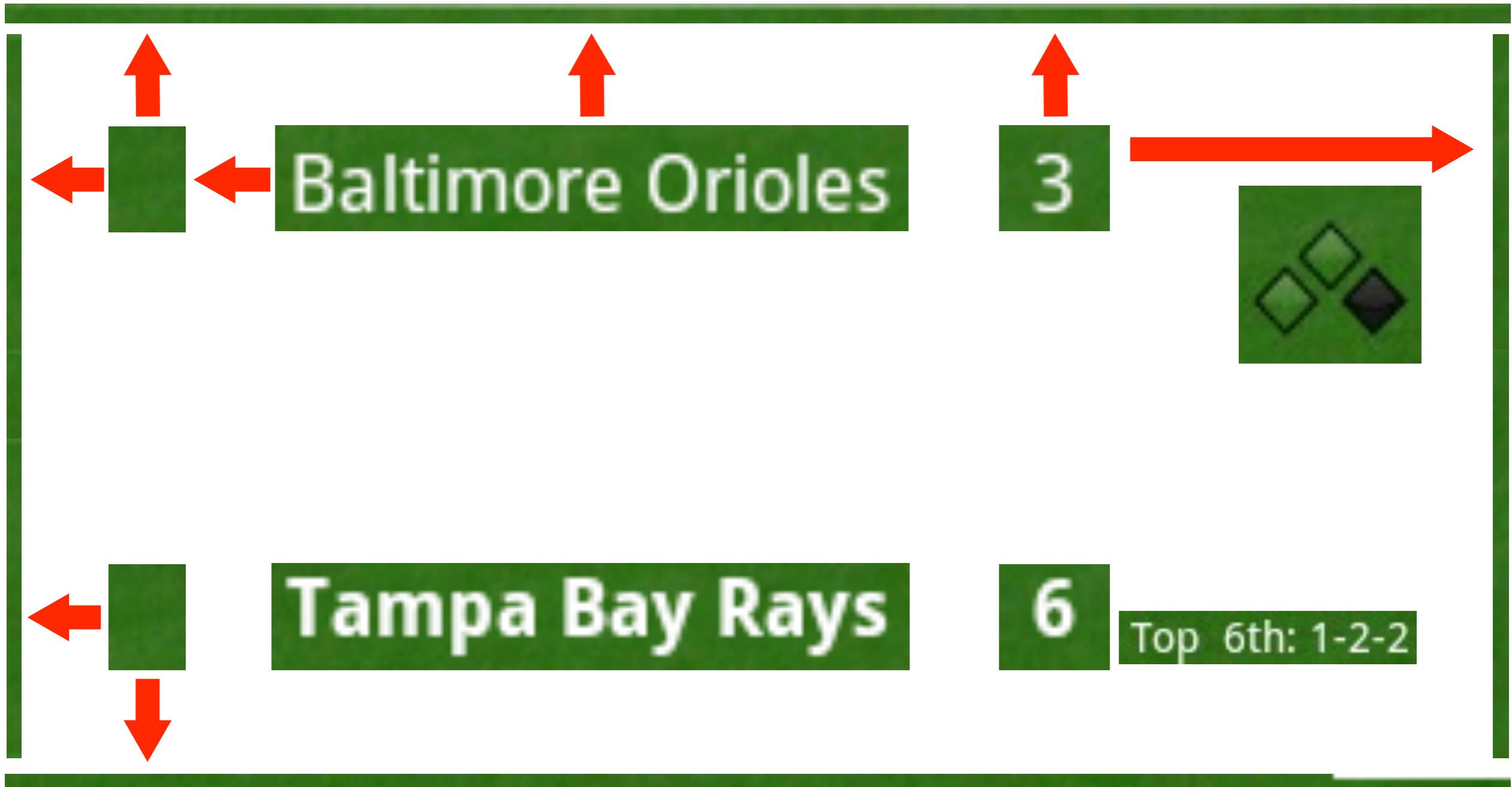
Fixing the Problem



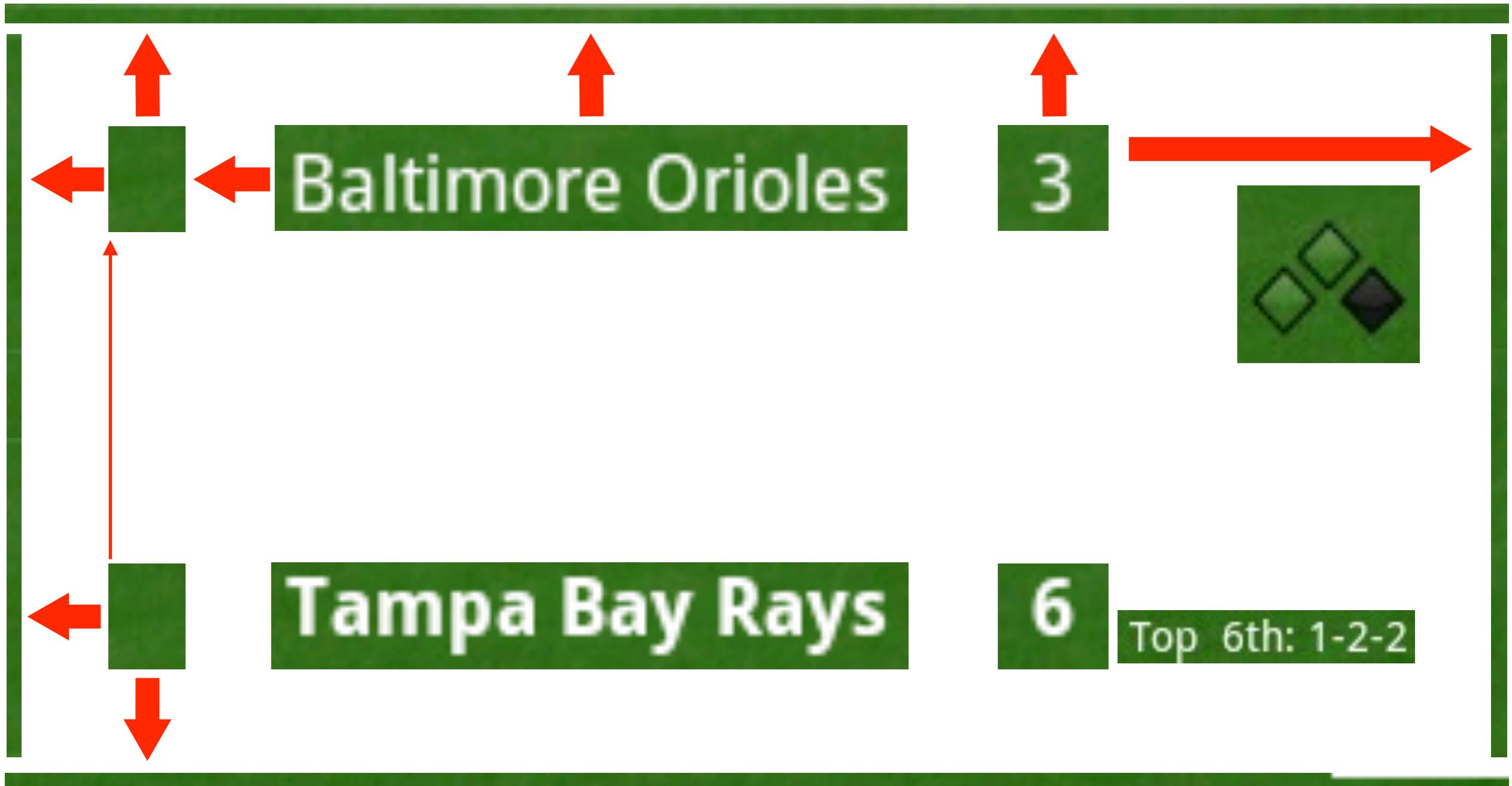
Fixing the Problem



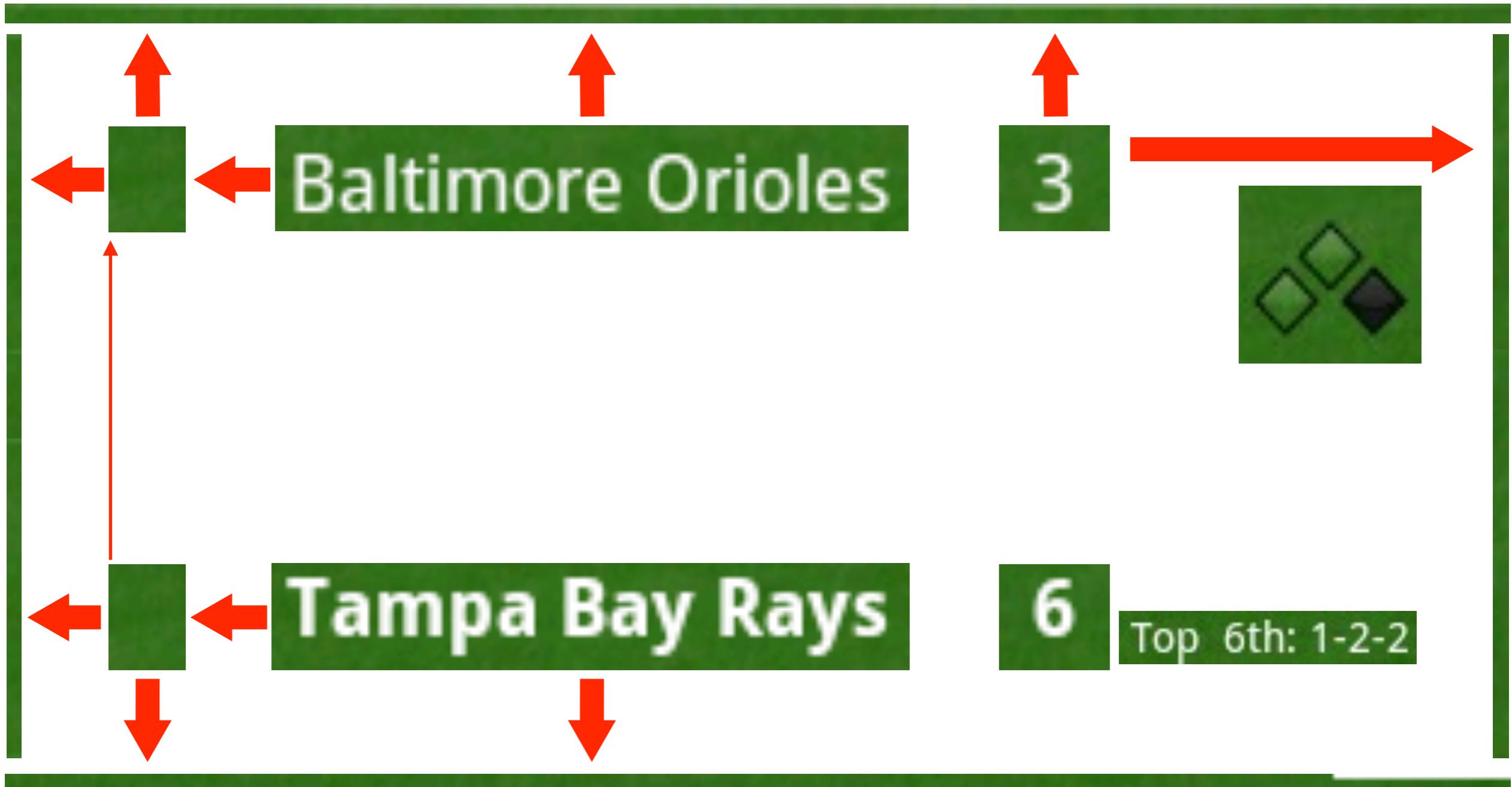
Fixing the Problem



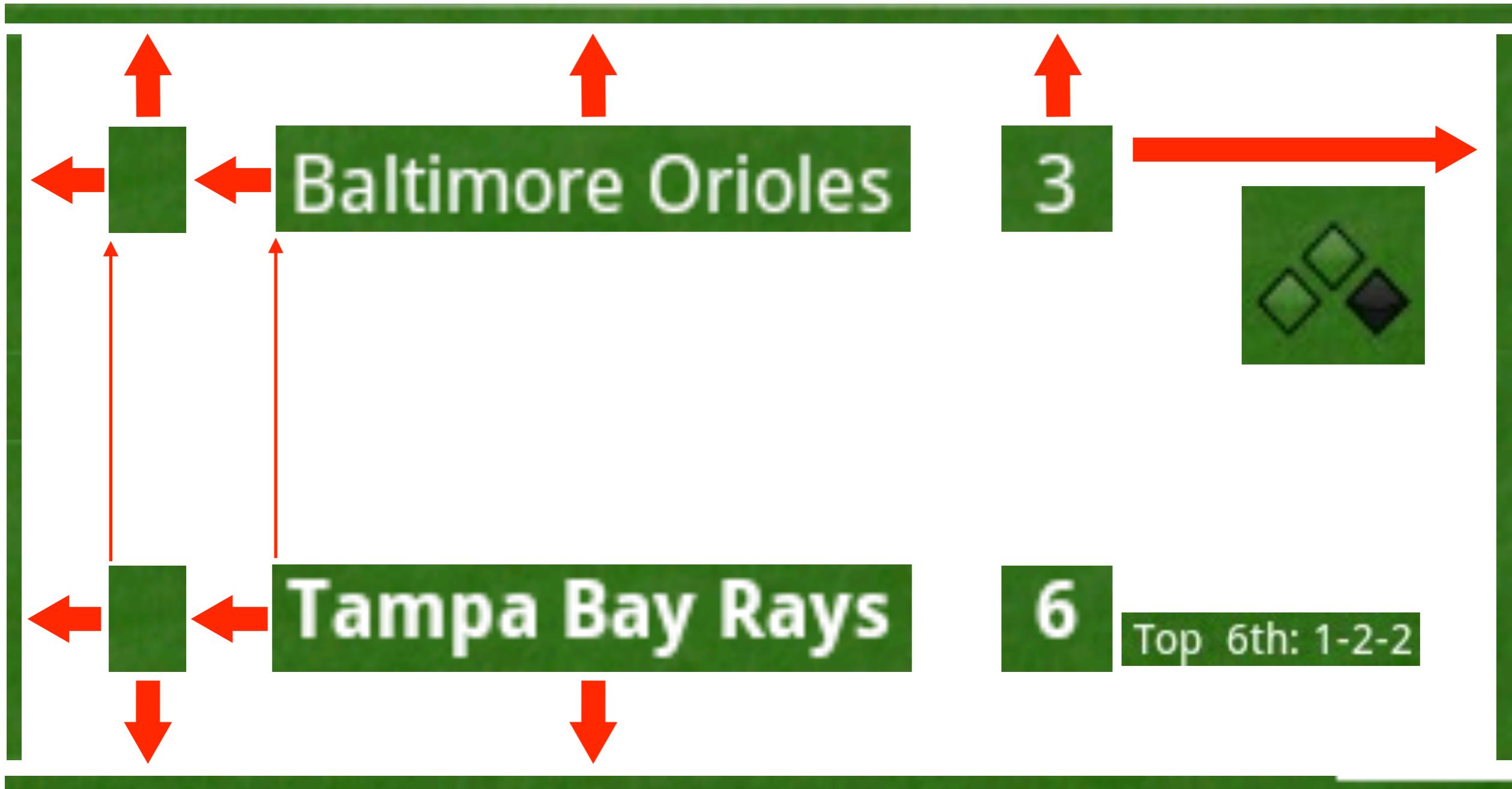
Fixing the Problem



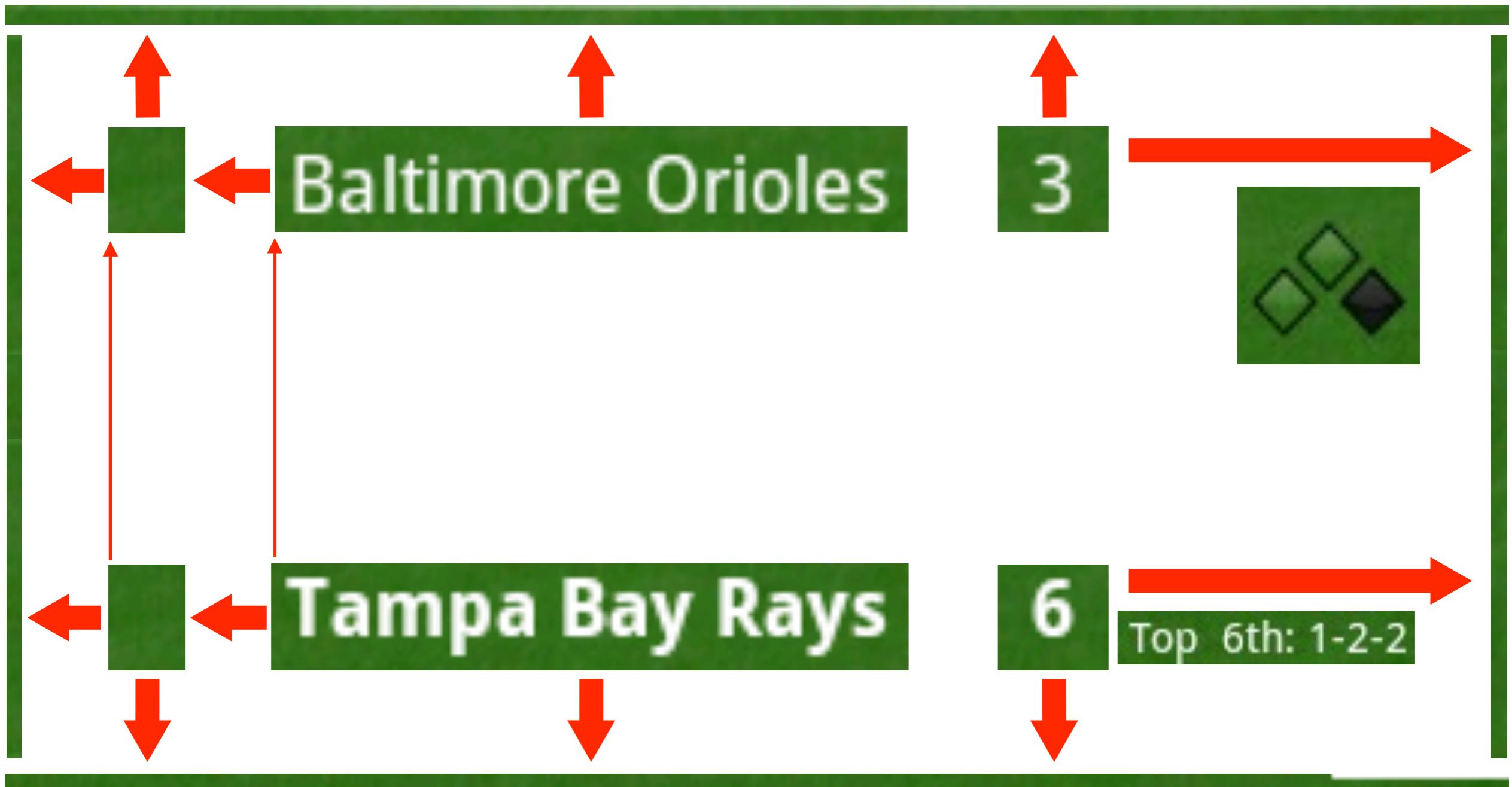
Fixing the Problem



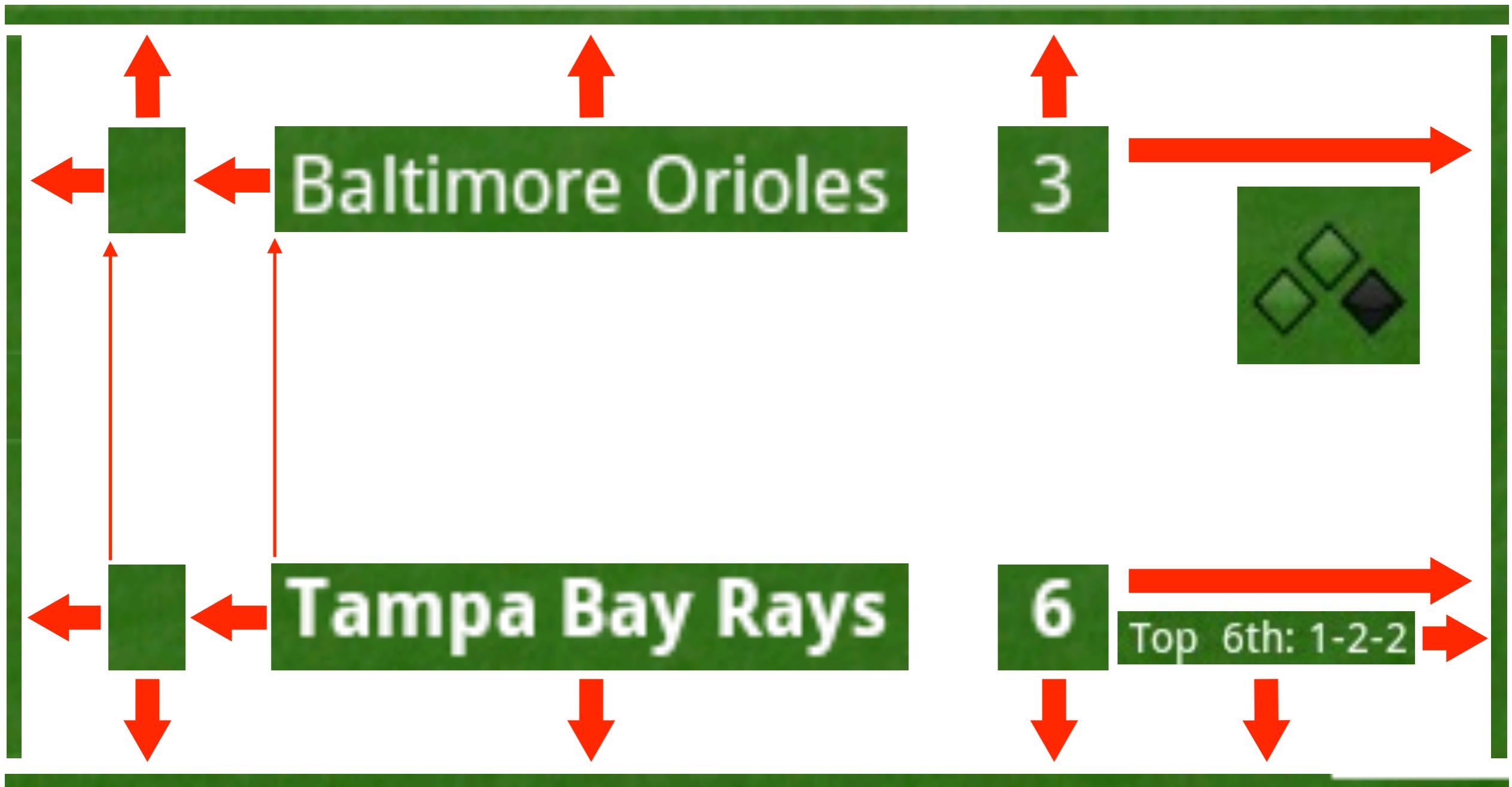
Fixing the Problem



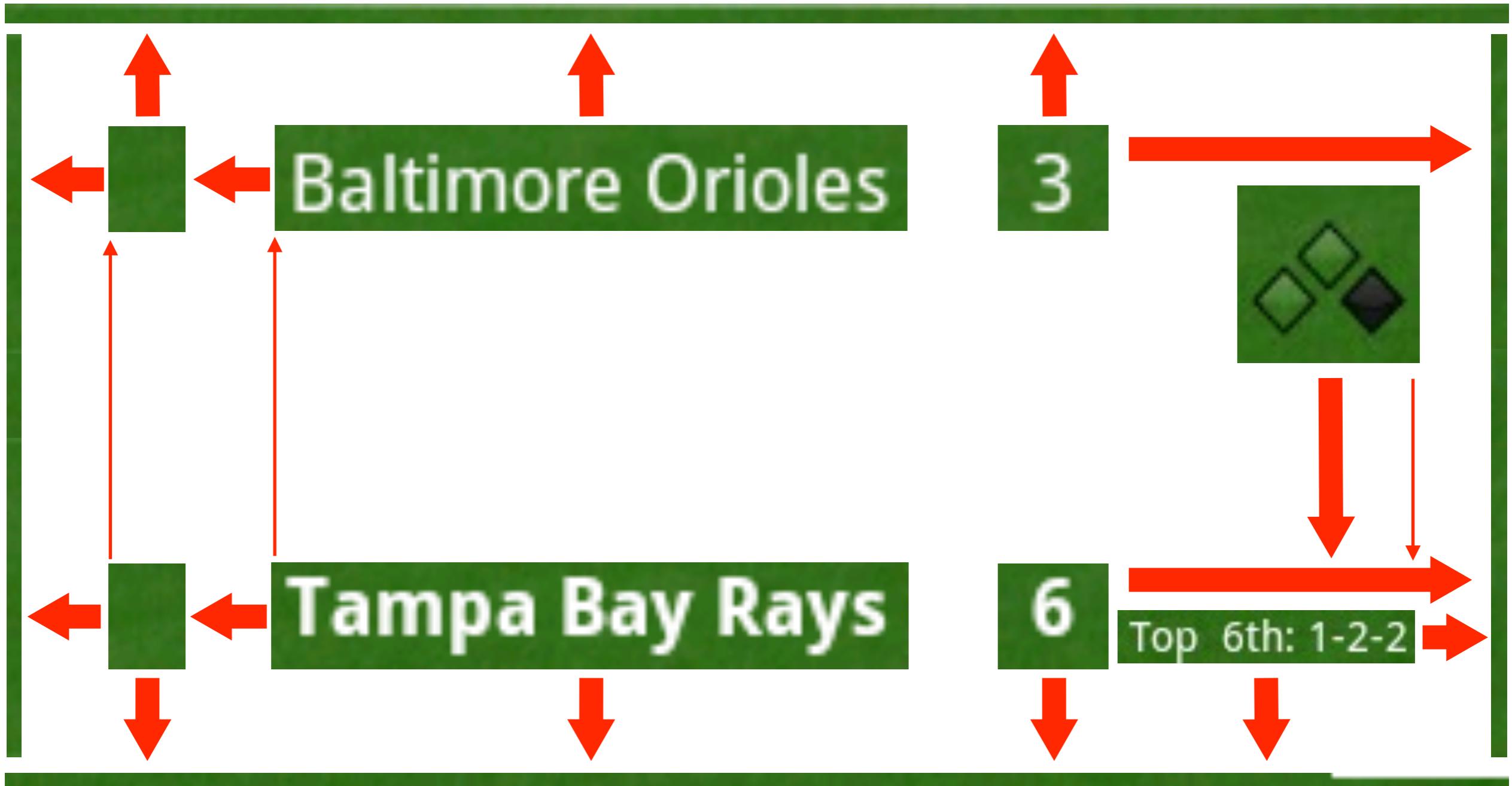
Fixing the Problem



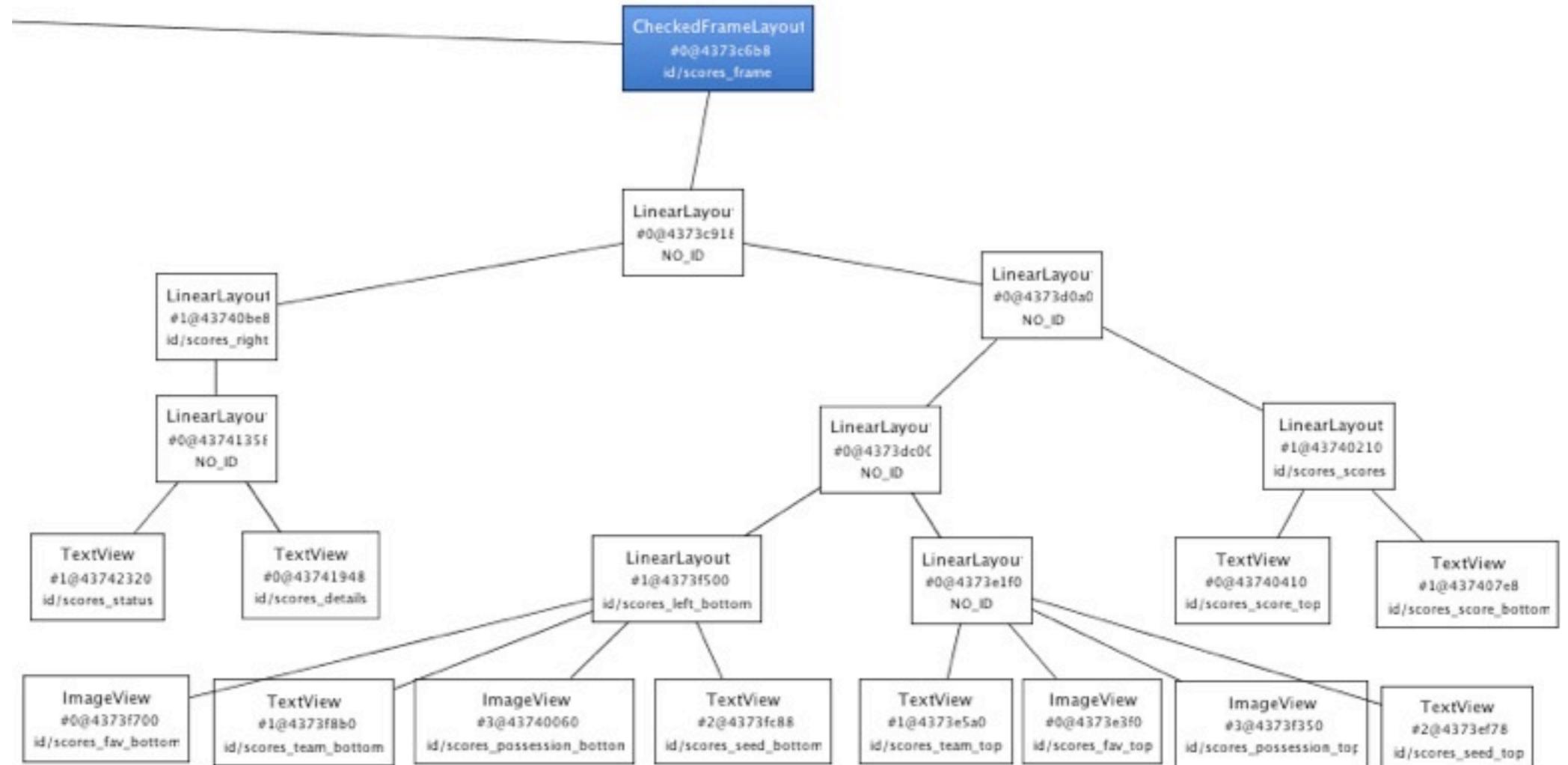
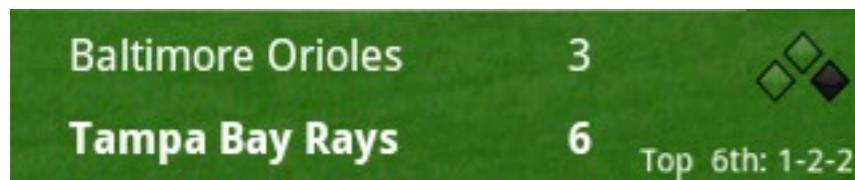
Fixing the Problem



Fixing the Problem



Fixing the problem



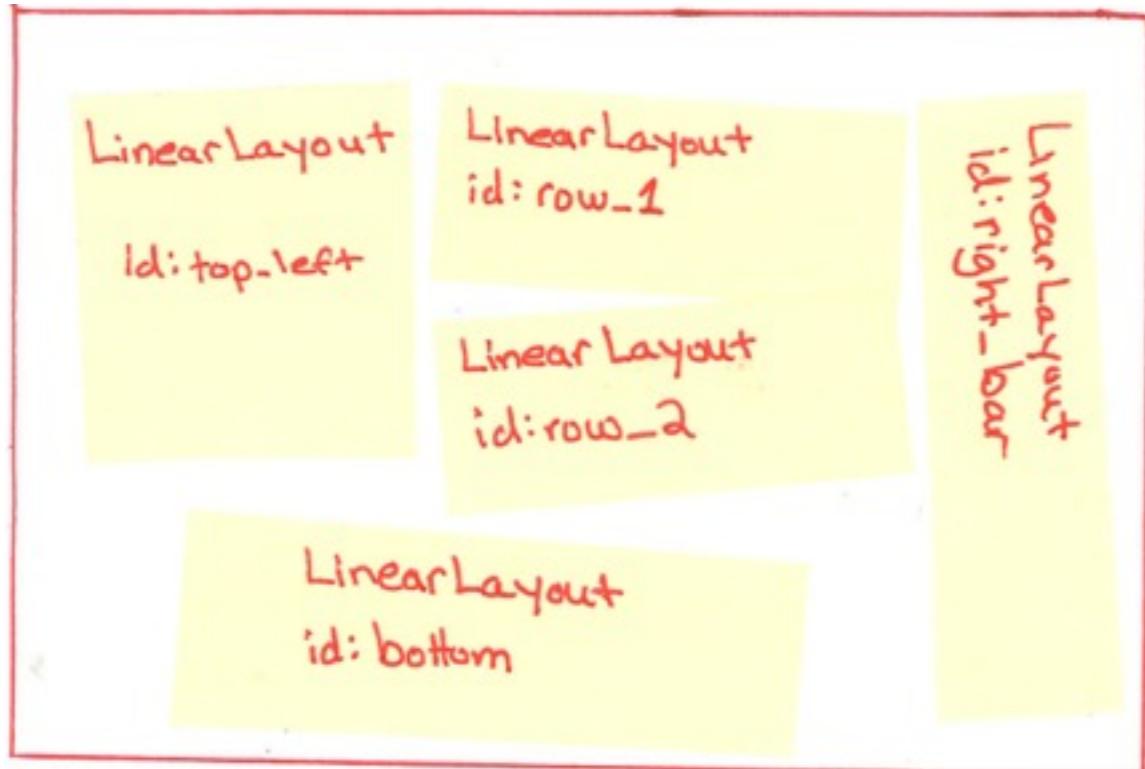
Fixing the problem



If it looks the same, who cares?

- Flatten for aerodynamic performance
- RelativeLayouts adapt to their environment
- RelativeLayouts force you to think

Key Learnings



- Layouts are code too and will evolve iteratively
- LinearLayouts are simple to visualize and understand, but can result in overly complex, less flexible hierarchies
- Rationalization needs to happen to keep the layout comprehensible and efficient



Traveler's Tales: Squeaky wheels

What's that noise?

- Same process id, similar object count, similar sizes

```
D/dalvikvm( 1825): GC freed 7 objects / 313984 bytes in 121ms
D/dalvikvm( 1825): GC freed 6 objects / 316016 bytes in 115ms
D/dalvikvm( 1825): GC freed 7 objects / 318080 bytes in 120ms
D/dalvikvm( 1825): GC freed 6 objects / 320112 bytes in 123ms
D/dalvikvm( 1825): GC freed 7 objects / 322176 bytes in 122ms
D/dalvikvm( 1825): GC freed 7 objects / 324312 bytes in 120ms
D/dalvikvm( 1825): GC freed 20 objects / 326920 bytes in 115ms
D/dalvikvm( 1825): GC freed 6 objects / 328272 bytes in 120ms
D/dalvikvm( 1825): GC freed 7 objects / 330336 bytes in 118ms
D/dalvikvm( 1825): GC freed 6 objects / 332368 bytes in 116ms
D/dalvikvm( 1825): GC freed 7 objects / 334432 bytes in 119ms
D/dalvikvm( 1825): GC freed 6 objects / 336464 bytes in 115ms
D/dalvikvm( 1825): GC freed 7 objects / 338528 bytes in 119ms
D/dalvikvm( 1825): GC freed 6 objects / 340560 bytes in 120ms
D/dalvikvm( 1825): GC freed 7 objects / 342624 bytes in 114ms
D/dalvikvm( 1825): GC freed 6 objects / 344656 bytes in 120ms
D/dalvikvm( 1825): GC freed 7 objects / 346720 bytes in 115ms
D/dalvikvm( 1825): GC freed 6 objects / 348752 bytes in 138ms
```

Allocation Tracker

The screenshot shows the Allocation Tracker interface with the following components:

- Top Navigation Bar:** A horizontal bar with tabs: Info, Threads, VM Heap, Allocation Tracker (which is selected and highlighted in blue), Sysinfo, Emulator Control, and Event Log.
- Buttons:** Two buttons below the tabs: "Start Tracking" and "Get Allocations".
- Allocation Table:** A large table with columns: Allocation Size, Allocated Class, Thread Id, Allocated in (with two sub-columns), and Allocated in (with two sub-columns). The table is currently empty.
- Class Hierarchy Table:** A table with columns: Class, Method, File, Line, and Native. The table is currently empty.
- Horizontal Scroll Bar:** A scroll bar located at the bottom of the class hierarchy table.

Allocation Tracking

The screenshot shows the Allocation Tracker tab of the DDMS tool. The top part displays a table of allocations:

Allocation Size	Allocated Class	Thread Id	Allocated in	Allocated in
95032	byte[]	17	com.example.android.util.WebService	fetchUrl
94904	byte[]	17	com.example.android.util.WebService	fetchUrl
94776	byte[]	17	com.example.android.util.WebService	fetchUrl
94648	byte[]	17	com.example.android.util.WebService	fetchUrl
94520	byte[]	17	com.example.android.util.WebService	fetchUrl
94392	byte[]	17	com.example.android.util.WebService	fetchUrl
94264	byte[]	17	com.example.android.util.WebService	fetchUrl
94136	byte[]	17	com.example.android.util.WebService	fetchUrl
94008	byte[]	17	com.example.android.util.WebService	fetchUrl
93880	byte[]	17	com.example.android.util.WebService	fetchUrl
93752	byte[]	17	com.example.android.util.WebService	fetchUrl
93624	byte[]	17	com.example.android.util.WebService	fetchUrl

The bottom part shows a table of tracked classes:

Class	Method	File	Line	Native
com.example.android.util.WebService	fetchUrl	WebService.java	50	false
com.example.android.util.WebService	fetchRemoteList	WebService.java	17	false
com.example.android.data.ItemDao	findItems	ItemDao.java	72	false
com.example.android.ItemList	getItems	ItemList.java	180	false
com.example.android.ItemList	access\$7	ItemList.java	175	false
com.example.android.ItemList\$5	run	ItemList.java	242	false
java.util.Timer\$TimerImpl	run	Timer.java	293	false

Using the right tool

- Problem caused by old code
 - Used for speed and convenience, over-stayed its welcome

```
public static byte[] fetchUri(String url) {  
    ...  
    InputStream in = openUrl(url);  
    byte[] response = new byte[0];  
    byte[] readChunk = new byte[384];  
    int byteSize = 0;  
    while ( (byteSize = in.read(readChunk)) != -1) {  
        byte[] alreadyRead = response;  
        response = new byte[alreadyRead.length + byteSize];  
        mergeArrays(response, alreadyRead, readChunk);  
    }  
}
```

Fixing the problem

- Rewrite code to fit the problem
- Anticipate larger read sizes, read in pieces, zipper together only at the end

```
public static byte[] fetchUri(String url) {  
    ...  
    InputStream in = openUrl(url);  
    ArrayList <byte[]> responsePieces = new ArrayList <byte[]>();  
    byte[] readChunk = new byte[2048];  
    int biteSize = 0;  
    int responseSize = 0;  
    while ( (biteSize = in.read(readChunk)) != -1) {  
        responsePieces.add(readChunk);  
        responseSize += biteSize;  
        readChunk = new byte[2048];  
    }  
    byte[] fullResponse = new byte[responseSize];  
    mergeArrays(fullResponse, responseSize);  
}
```

- Result: 90% decrease in read times

Key Learnings

- Take care with reused or prototype code
- Log messages if code is used beyond tolerances



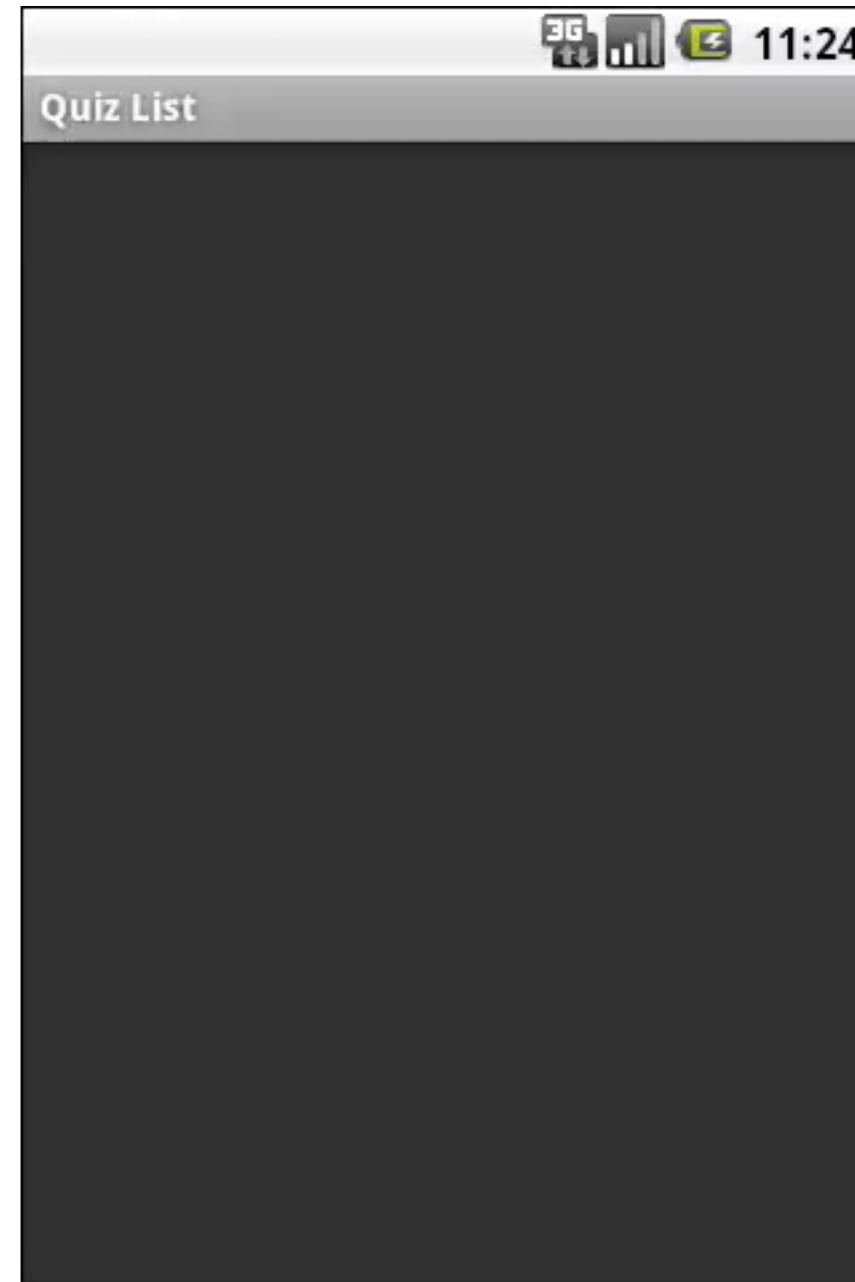
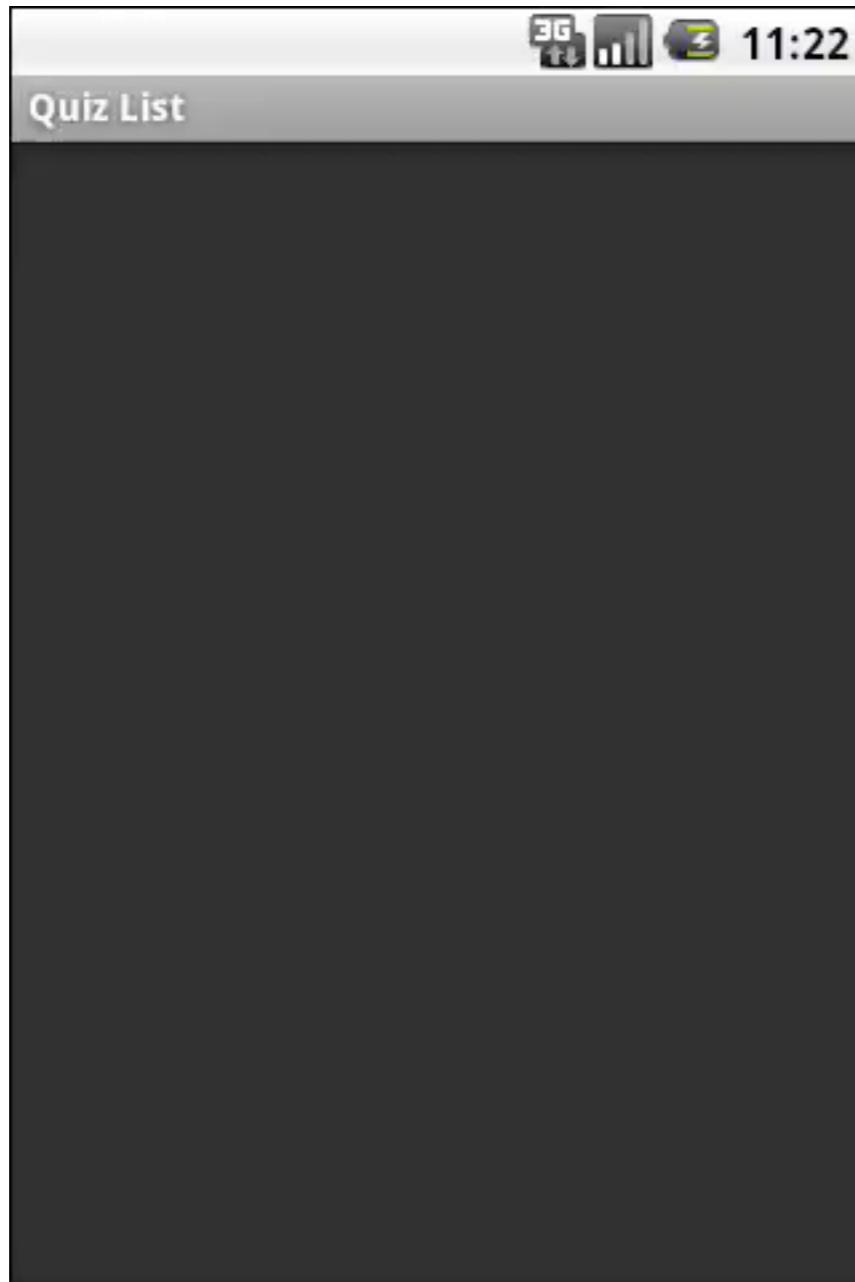
Pop quiz!

What's gone wrong?

```
E/AndroidRuntime( 1004): Uncaught handler: thread main exiting due to uncaught exception
E/AndroidRuntime( 1004): java.lang.StackOverflowError
E/AndroidRuntime( 1004):     at android.text.Styled.measureText(Styled.java:371)
E/AndroidRuntime( 1004):     at android.text.Layout.measureText(Layout.java:1600)
E/AndroidRuntime( 1004):     at android.text.StaticLayout.generate(StaticLayout.java:721)
E/AndroidRuntime( 1004):     at android.text.StaticLayout.<init>(StaticLayout.java:97)
E/AndroidRuntime( 1004):     at android.text.StaticLayout.<init>(StaticLayout.java:54)
E/AndroidRuntime( 1004):     at android.text.StaticLayout.<init>(StaticLayout.java:45)
E/AndroidRuntime( 1004):     at android.widget.TextView.makeNewLayout(TextView.java:4757)
E/AndroidRuntime( 1004):     at android.widget.TextView.onMeasure(TextView.java:4984)
E/AndroidRuntime( 1004):     at android.view.View.measure(View.java:7115)
E/AndroidRuntime( 1004):     at android.view.ViewGroup.measureChildWithMargins(ViewGroup.java:2875)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureChildBeforeLayout(LinearLayout.java:888)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureHorizontal(LinearLayout.java:619)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.onMeasure(LinearLayout.java:280)
E/AndroidRuntime( 1004):     at android.view.View.measure(View.java:7115)
E/AndroidRuntime( 1004):     at android.view.ViewGroup.measureChildWithMargins(ViewGroup.java:2875)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureChildBeforeLayout(LinearLayout.java:888)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureHorizontal(LinearLayout.java:619)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.onMeasure(LinearLayout.java:280)
E/AndroidRuntime( 1004):     at android.view.View.measure(View.java:7115)
E/AndroidRuntime( 1004):     at android.view.ViewGroup.measureChildWithMargins(ViewGroup.java:2875)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureChildBeforeLayout(LinearLayout.java:888)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureHorizontal(LinearLayout.java:619)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.onMeasure(LinearLayout.java:280)
E/AndroidRuntime( 1004):     at android.view.View.measure(View.java:7115)
E/AndroidRuntime( 1004):     at android.view.ViewGroup.measureChildWithMargins(ViewGroup.java:2875)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureChildBeforeLayout(LinearLayout.java:888)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureHorizontal(LinearLayout.java:619)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.onMeasure(LinearLayout.java:280)
E/AndroidRuntime( 1004):     at android.view.View.measure(View.java:7115)
E/AndroidRuntime( 1004):     at android.view.ViewGroup.measureChildWithMargins(ViewGroup.java:2875)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureChildBeforeLayout(LinearLayout.java:888)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureHorizontal(LinearLayout.java:619)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.onMeasure(LinearLayout.java:280)
E/AndroidRuntime( 1004):     at android.view.View.measure(View.java:7115)
E/AndroidRuntime( 1004):     at android.view.ViewGroup.measureChildWithMargins(ViewGroup.java:2875)
E/AndroidRuntime( 1004):     at android.widget.LinearLayout.measureChildBeforeLayout(LinearLayout.java:888)
```

What's gone wrong?

What change caused this slow down?



Q&A

09

