Android and Bitmaps: How hard could it be?

Maksim Lin
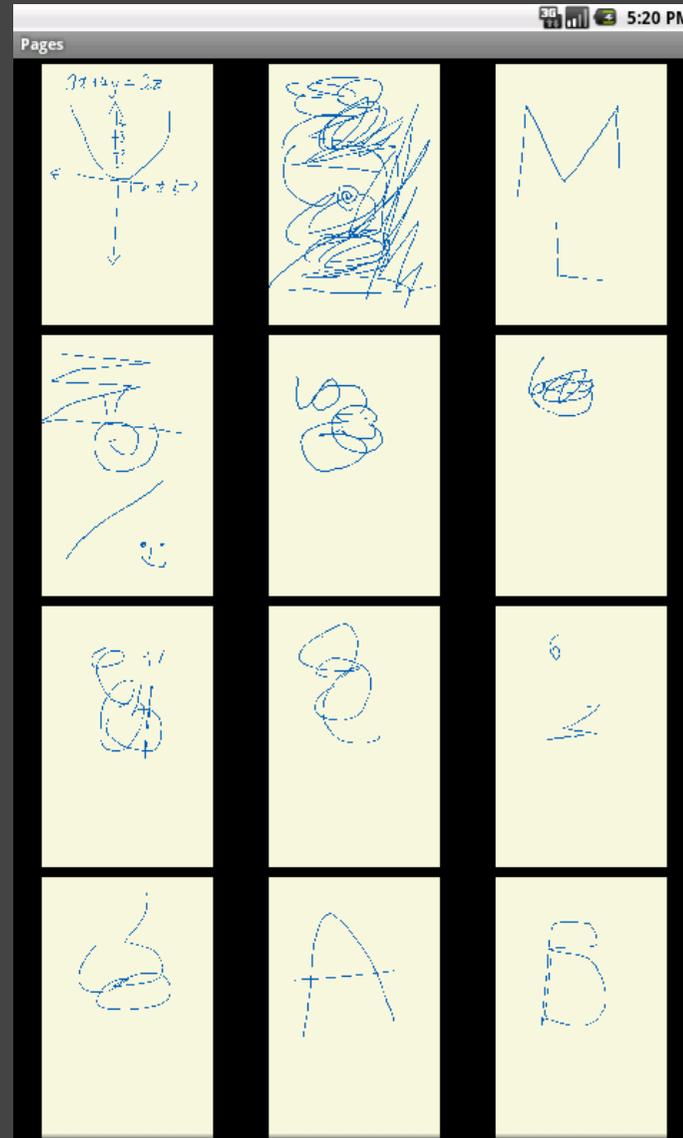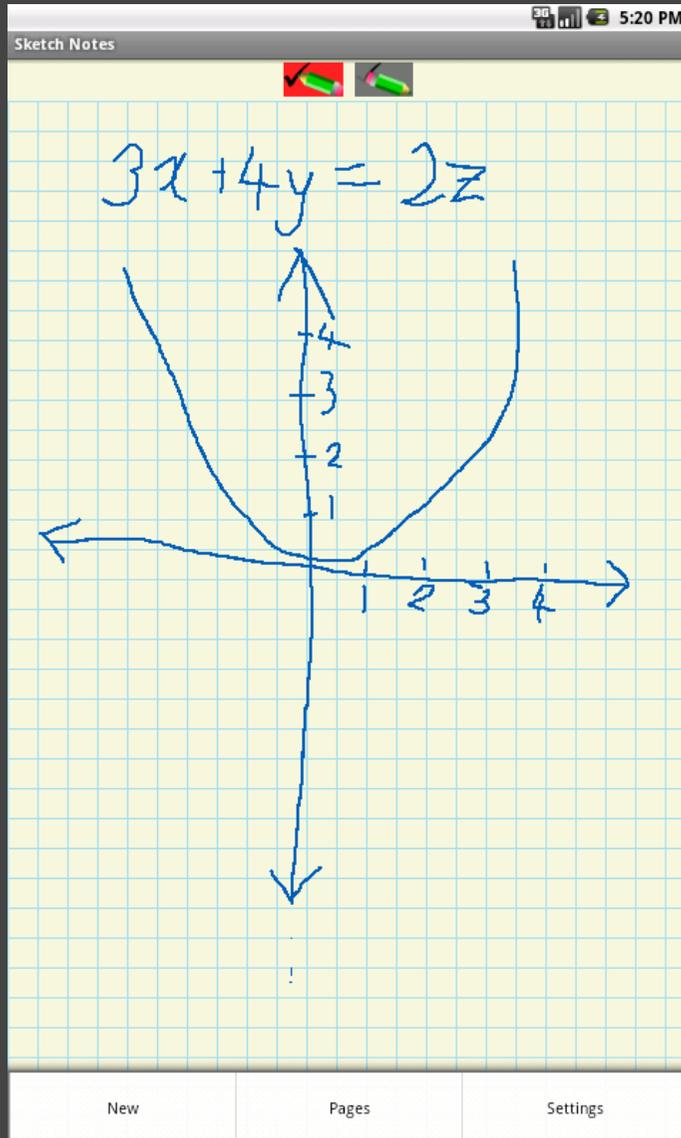Manichord Mobile Solutions

Intro

I'm an old time Java dev (serverside and J2ME), but a Android apprentice...

I'll talk about my experiences recently writing and ***debugging*** my "apprentice piece" Android App.

Specifically what I learnt about debugging memory issues with Bitmaps

My App uses Bitmaps *alot*

Bitmaps

- Bitmaps make life just that little bit more interesting...
- Up until Honeycomb Bitmap objects "data" memory is allocated in native code, "outside" Dalvik VM object map **BUT** still count towards your applications heap quota
- The heap quota is not very big...
- Android 2.1 = 16MB, 2.2 = 24MB, 2.3 = 32MB
- On Optus v9 Tablet: 732x480x4 ~ 1.4MB per "screen" !!
- Because its **native** allocated memory, the SDKs Allocation Tracker tool is not very helpful :-(

The Dreaded OOM !

java.lang.OutOfMemoryError:
bitmap size exceeds VM budget

- Thats the helpful exception you get when you run out of heap when trying to create a new Bitmap without enough heap space...

"All this has happened before, and all this will happen again."

Old School Debugging

So after a bit of Googling you come up with this:

```
public static String getMemUsageString() {
  int usedNativeKbs = (int)
      (Debug.getNativeHeapAllocatedSize() / 1024L);

  return String.format("SKNotes Memory Used: %d KB",
                              usedNativeKbs);
}
```

 And then lots of Log.d() ...

 theres lots of other interesting methods on Debug...

To reason why…

- So why is this happening in the first place?
- Answer in **bug 8488** in AOSP bug tracker if you have time to read through it...
- basically GC it seems does not get around to calling the native code in the Bitmaps finaliser before your app hits the out of memory condition
- But should we really follow the advice in that bug report of lots of call to System.gc() ?!?!?
- **NO!**
- A good blog post: http://goo.gl/PpQua elaborates further (all the way down into native c++ code!) to basically point out the need for calling:
  `mybitmap.recycle()`
  due to async nature of GC wehter or not you call .gc()

One more helpful hint…

Use the BitmapFactoryOptions.inSampleSize !
e.g.

```
static {
  private static BitmapFactory.Options
    mBitmapFactoryOpts =
                new BitmapFactory.Options();
  mBitmapFactoryOpts.inSampleSize = THUMB_SCALE_FACTOR;
}

...

//Decode with inSampleSize
Bitmap loadedBM =
    BitmapFactory.decodeFile(filePath, mBitmapFactoryOpts);
```

Lessons Learnt

stackoverflow is your friend!

Google is your friend!

d.android.com is your friend!

References

Sketch Notes App in Android Market :-)

Sketch Notes App source code available on Github: http://github.com/maks/Sketch-Notes

The Android Developers Blog: http://android-developers.blogspot.com/

Romain Guy blog: http://www.curious-creature.org/

Questions?