

OutOfMemoryErrorを知る

日本Androidの会 日高 正博

発表者紹介

- ▶ 日本Androidの会／関西支部
- ▶ 日高 正博
- ▶ Twitter :@mhidaka 
- ▶ Blog :http://d.hatena.ne.jp/hdk_embedded/



Hatena::Diary [日記] [検索] ブログトップ 記事一覧 記事を書く 管理 ログアウト ヘルプ

hdk_embeddedの日記

<前の5日分 | 新しいエントリ>

2010-05-03

Android 2.1アイコン作成方法 3つのポイント Android | 02:27 | 編集 ★★★★

ADF2010 LT Android開発Tipsの後半です。お待たせしました。

Android Developerの[Icon Design Guidelines](#)、[Android 2.0](#)が原典です。

英語ですがデザインについて大変参考になります。

以下を読んで興味を持ったひとは是非、読んでみてください。

- ランチャアイコンのデザインポイント

Androidピックス

- [BeagleBoardとu-boot](#)
- [BeagleBoardとタッチパネル](#)
- [ジェスチャの使い方](#)
- [開発環境構築の手順](#)
- [Android SDK 1.6アップグレード手順](#)
- [Androidでゲーム](#)
- [記事一覧](#)

目次

1. OutOfMemoryError

- ・概要／経緯

2. CaseStudy

- ・画像の読込／発生タイミング

3. Androidアプリケーション

- ・コンポーネント／ライフサイクル

4. Dalvik VM

- ・メモリ管理／GC／ヒープ

5. まとめ

- ・原因／改善策



OutOfMemory概要



OutOfMemoryErrorを知る

調査を思い立った背景

- Painter系アプリケーション作成時にOutOfMemoryを食らう
「大きい画像を読み込めない」

原因はメモリ不足

- Androidでは、みんな一度は経験したことがあるよね。
- 合いたくないのに、ちょくちょくお目にかかる。
- 特に重い画像データを扱うと頻発してしょうがない。

どんなケースでメモリが無くなる？

- 発生タイミングについて、よく分からない事が多かった



そこそこマニアックに分析、調査

分からないので調べてみた

- なぜOutOfMemoryになるの？
- Androidアプリってどうメモリを管理しているの？
- だれか良い回避策知らない？

仮想マシンDalvik

- **A JIT Compiler for Android's Dalvik VM – Ben Cheng, Bill Buzbee**
- Google I/O 2010でJITのお話がある。
マニアックな方なら知っておいて損はないです。たぶん。

CaseStudy

»» OutOfMemoryの発生

CaseStudy



アプリケーション作成時

- ・ 画像加工、カメラ、写真Viewerなど



大きい画像を扱いたい

- ・ リソースファイル／外部メディア読み込み



どうやる？

- ・ `android.graphics.BitmapFactory`

BitmapFactory

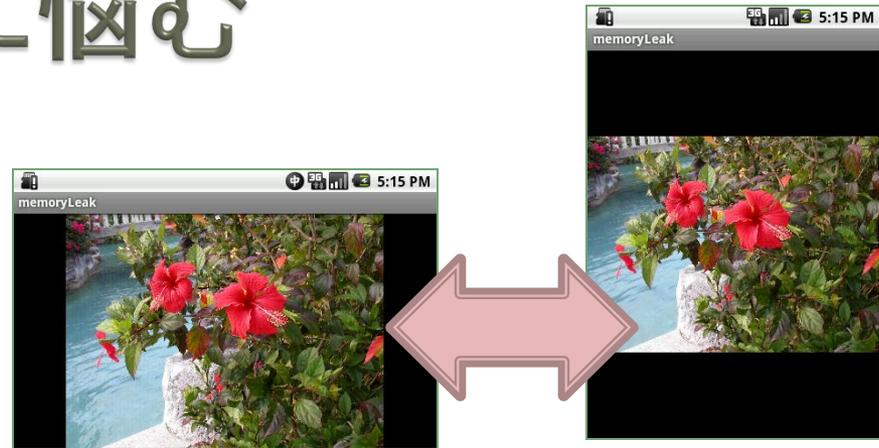
- ▶ public class **BitmapFactory** extends Object
- ▶ android.graphics.BitmapFactory

- ▶ **Class Overview**
 - Creates Bitmap objects from various sources, including files, streams, and byte-arrays.

- ▶ **Public Methods**
 - static Bitmap decodeFile(String pathName, BitmapFactory.Options opts) Decode a file path into a bitmap.
 - static Bitmap decodeResource(Resources res, int id) Synonym for decodeResource(Resources, int, android.graphics.BitmapFactory.Options) will null Options.

OutOfMemoryに悩む

- ▶ 縦横切り替え
 - CTRL+F12連打



```
05-14 17:16:45.035: INFO/ActivityManager(51): Config changed:  
{ scale=1.0 imsi=310/260 loc=en_US touch=3 keys=2/1/2 nav=3/1  
orien=1 layout=18}  
05-14 17:16:45.075: ERROR/dalvikvm-heap(187): 2457600-byte external  
allocation too large for this process.  
05-14 17:16:45.075: ERROR/(187): VM won't let us allocate 2457600  
bytes...省略...  
05-14 17:16:45.204: ERROR/AndroidRuntime(187):  
java.lang.OutOfMemoryError: bitmap size exceeds VM budget
```

Android Application

➤➤ コンポーネントとライフサイクル

Androidアプリケーションおさらい

Activity

- ・ 視覚的なUIを伴うコンポーネント

Service

- ・ 持続的に実行されるコンポーネント

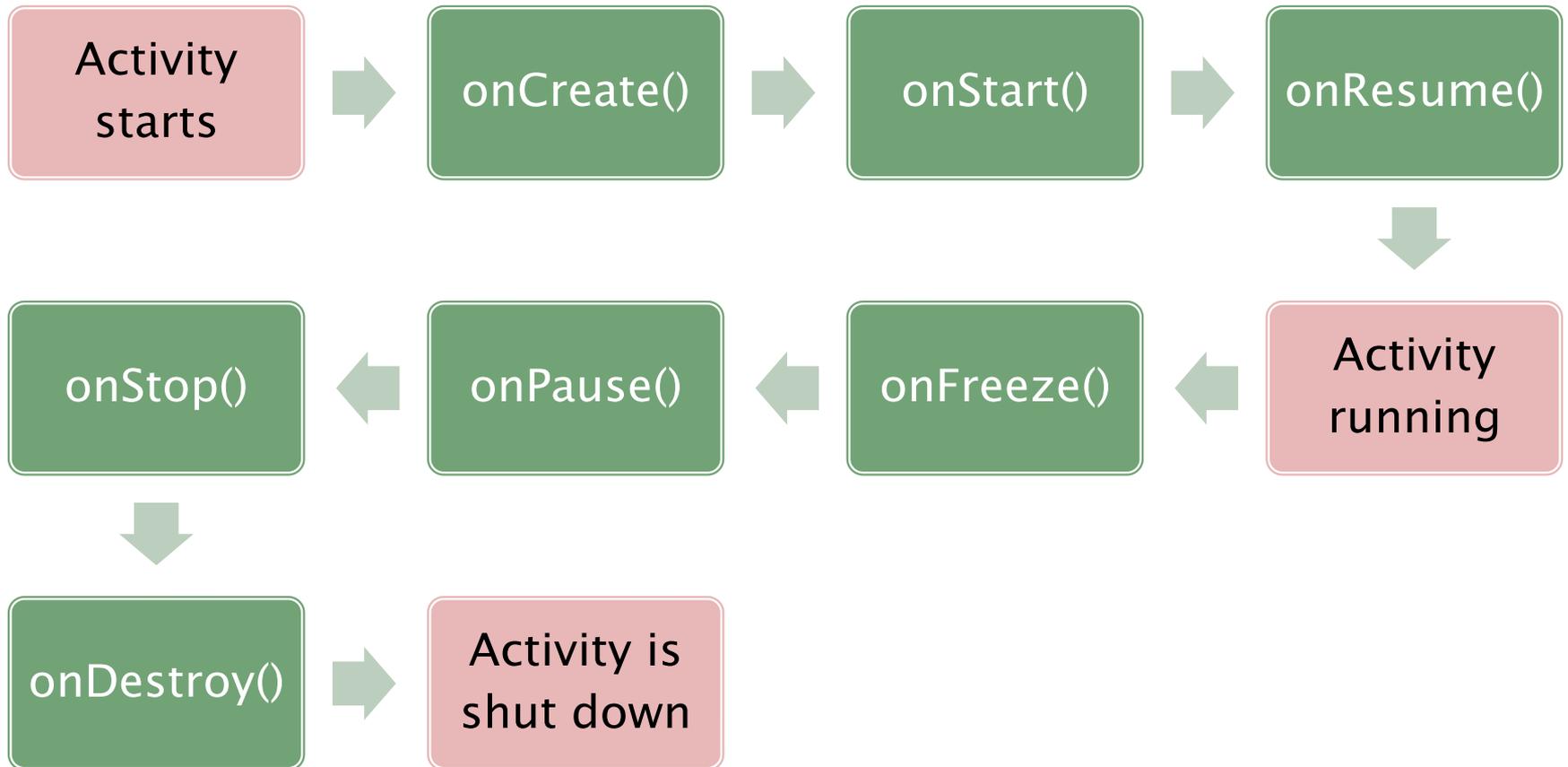
BroadcastReceiver

- ・ システム、他のアプリからの連絡を受け取る

ContentsProvider

- ・ アプリケーション間のデータ交換のため

Activityのライフサイクル



Activity回転時のライフサイクル

- ▶ 一旦、onDestroy()されてから、再生成

システムによるアクティビティ再生成
にOutOfMemory

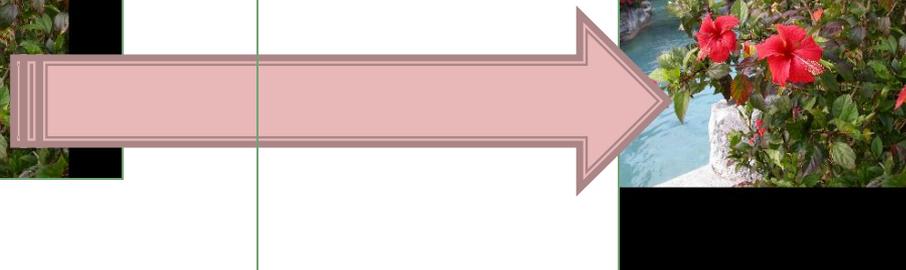
onCreate()

onPause()

onDestroy()

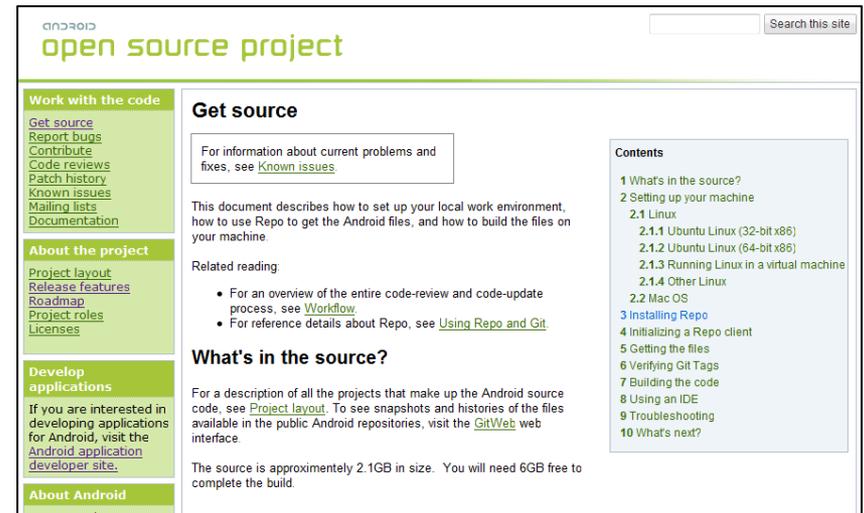
onResume()

onPause()



OutOfMemory調査

- ▶ メモリ確保できていない。
- ▶ 画像読み込みには、どれだけスタックが必要？
- ▶ Android App frameworkのソースコードを追う
 - [Android OpenSource Project](http://source.android.com/download)
<http://source.android.com/download>



The screenshot shows the 'open source project' page for Android. The main heading is 'Get source'. Below it, there is a search bar and a 'Search this site' button. The page is divided into several sections:

- Work with the code:** Includes links for Get source, Report bugs, Contribute, Code reviews, Patch history, Known issues, Mailing lists, and Documentation.
- About the project:** Includes links for Project layout, Release features, Roadmap, Project roles, and Licenses.
- Develop applications:** A section for those interested in developing applications for Android, with a link to the developer site.
- About Android:** A section for general information about Android.
- Get source:** The main section, containing a search box, a link to 'Known issues', and a paragraph explaining how to set up a local work environment using Repo. It also includes 'Related reading' with links to 'Workflow' and 'Using Repo and Git'.
- What's in the source?:** A section providing a description of the projects in the Android source code, with a link to 'Project layout' and a link to the 'GitWeb' interface.
- Contents:** A table of contents listing the sections of the document, such as '1 What's in the source?', '2 Setting up your machine', '2.1 Linux', '2.1.1 Ubuntu Linux (32-bit x86)', '2.1.2 Ubuntu Linux (64-bit x86)', '2.1.3 Running Linux in a virtual machine', '2.1.4 Other Linux', '2.2 Mac OS', '3 Installing Repo', '4 Initializing a Repo client', '5 Getting the files', '6 Verifying Git Tags', '7 Building the code', '8 Using an IDE', '9 Troubleshooting', and '10 What's next?'.

frameworks¥base¥graphics¥java¥android¥graphics¥
BitmapFactory.java

バッファで32KB

▶ decodeResource

– decodeResourceStream

– decodeStream

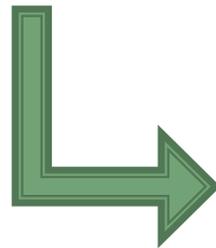
is = new BufferedInputStream(is, 16*1024);

...

tempStorage = new byte[16 * 1024];

...

bm = nativeDecodeStream
(is, tempStorage, outPadding, opts);



JNI – NativeCodeの世界へ

Androidのフレームワーク



Native Code

- ▶ Java Native Interface (JNI)
- ▶ 仮想マシンからネイティブなC/C++の呼び出し
- ▶ Just In Time Compilerの恩恵は？
- ▶ Froyoで導入されるJITが本当に高速なら高速化のためにJNIを使う必要は薄れる。

途中のまとめ

BitmapFactory

- ・ 小さな画像を読み込むときも32KB(16KB x 2個)のバッファを使う(+生データ分も必要だけど)

OutOfMemoryの発生タイミング

- ・ 問題が潜んでいそうなのはActivityのライフサイクル。
onDestroy() <消失>、onResume() <再生成>するポイント。

Dalvik VM

➤➤ 仮想マシンDalvikのメモリ管理

Dalvik VM

レジスタベースの仮想マシン

- ・ Dan Bornsteinおよび Google 社のエンジニアによりAndroidプラットフォームのために設計・開発

低メモリ環境に対して最適化

- ・ システムプロセス Zygoteによる効率的なVMインスタンス作成

VMインスタンスの同時起動

- ・ プロセス間の分離／メモリ管理／スレッドのサポート

Java VM ?

- ・ 動作するバイトコードがJavaバイトコードではない
- ・ 独自のDex形式。厳密にはJVMではない

Wikipedia:dalvik

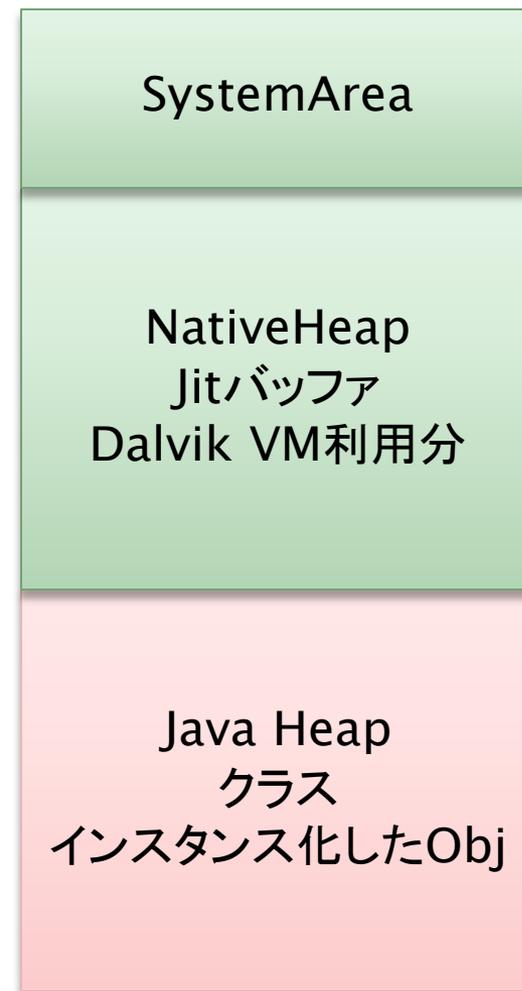
Dalvik VMのメモリ管理

▶ JavaHeap

- アプリ(VM)ごとに管理される
- VM内でスレッドを作った場合はスレッドでも共有されるメモリ空間
- (初期2MB,最大16MB ※システム依存)

▶ NativeHeap

- JITコンパイル展開用バッファや
- DalvikVMsystemの利用



メモリ管理 概念図

Dalvik VMのメモリ管理まとめ

JavaHeapのサイズ

- ・初期サイズが2MB、最大サイズが16MB(※システム依存)

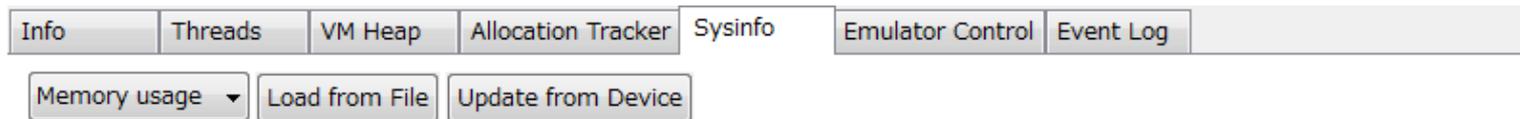
OutOfMemoryErrorの条件

- ・アロケーション時に取得。JavaHeap内で、要求サイズ分の空きが無ければ発生

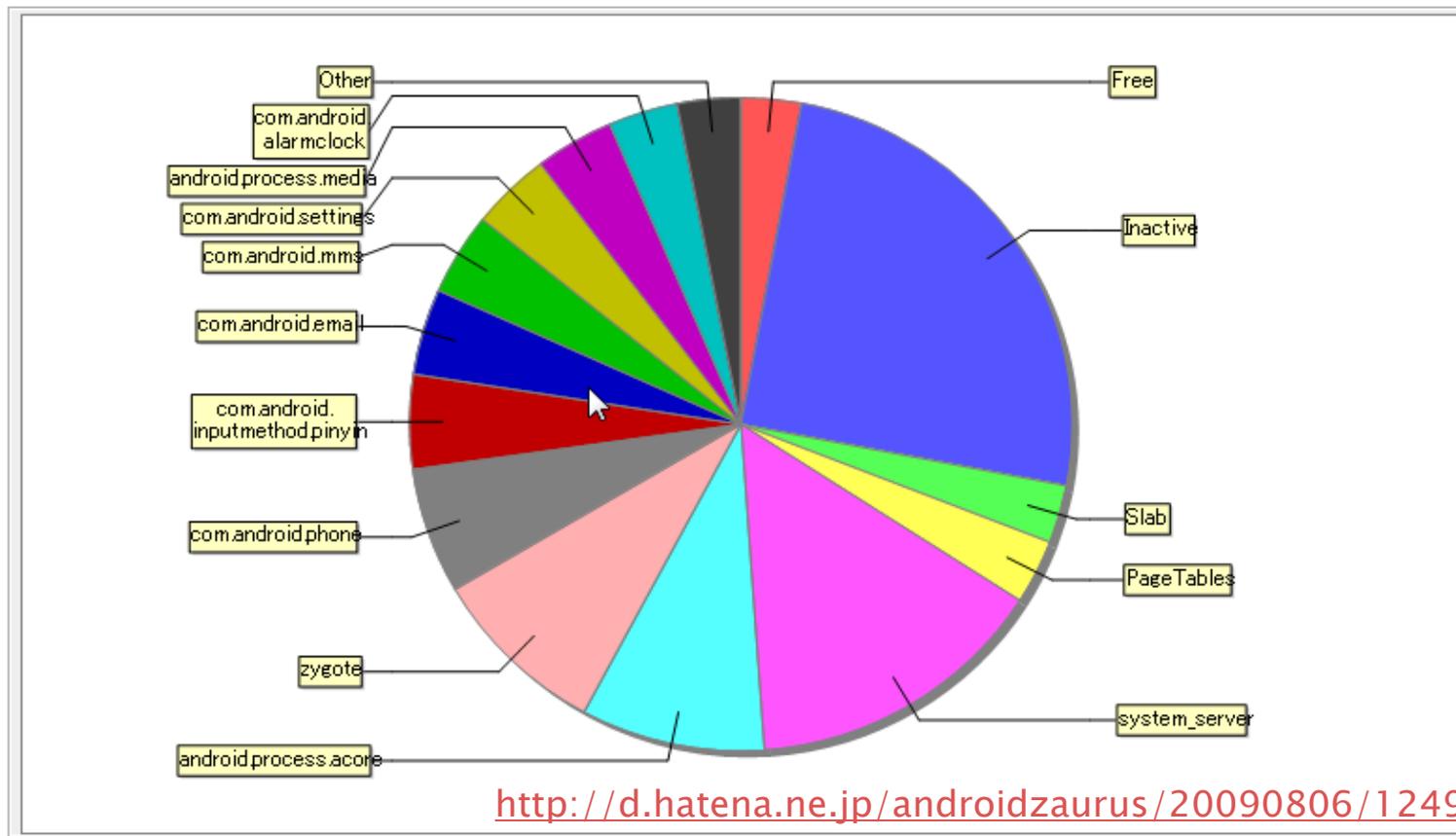
JNIでのメモリリソース消費

- ・オブジェクトはGCの効くJavaHeapを利用。
mallocされたメモリ空間は、NativeHeapが消費される。

メモリ使用量 調査:DDMS (JavaHeap+NativeHeapを把握)



PSS in kB



<http://d.hatena.ne.jp/androidzaurus/20090806/1249548020>

続: BitmapFactory

frameworks\base\core\jni\android\graphics\BitmapFactory.cpp

- ▶ nativeDecodeStream
 - doDecode
 - decoder->decode(stream, bitmap, prefConfig, decodeMode)

external\skia\src\images\SkImageDecoder_libjpeg.cpp

- ▶ SkJPEGImageDecoder::onDecode
 - this->allocPixelRef(bm, NULL)
 - SkBitmap::allocPixels
 - HeapAllocator::allocPixelRef
 - sk_malloc_flags(size.get32(), 0)

external\skia\src\core\SkMemory_stdlib.cpp

- sk_malloc_flags
 - void* p = malloc(size);

まとめ

- ▶▶ Bitmap読み込み時の
OutOfMemoryを防ぐ

調査まとめ

ビットマップオブジェクトは、JavaHeapに格納

- BitmapFactoryで作ったBitmapクラスは当然GCの対象

BitmapデータはNativeHeapに格納

- dlmalloc、mmap()で確保
- mallocはGC対象外。外部メモリ空間の任意の並べ替えは、OSの問題。難しい。

Bitmapデータの解放は？

- ビットマップオブジェクトがデータポインタを手放すとBitmapデータが解放されるようだ
- そんなメソッド提供されているのかな？
- Bitmapクラスにrecycle() / isRecycle()があった！

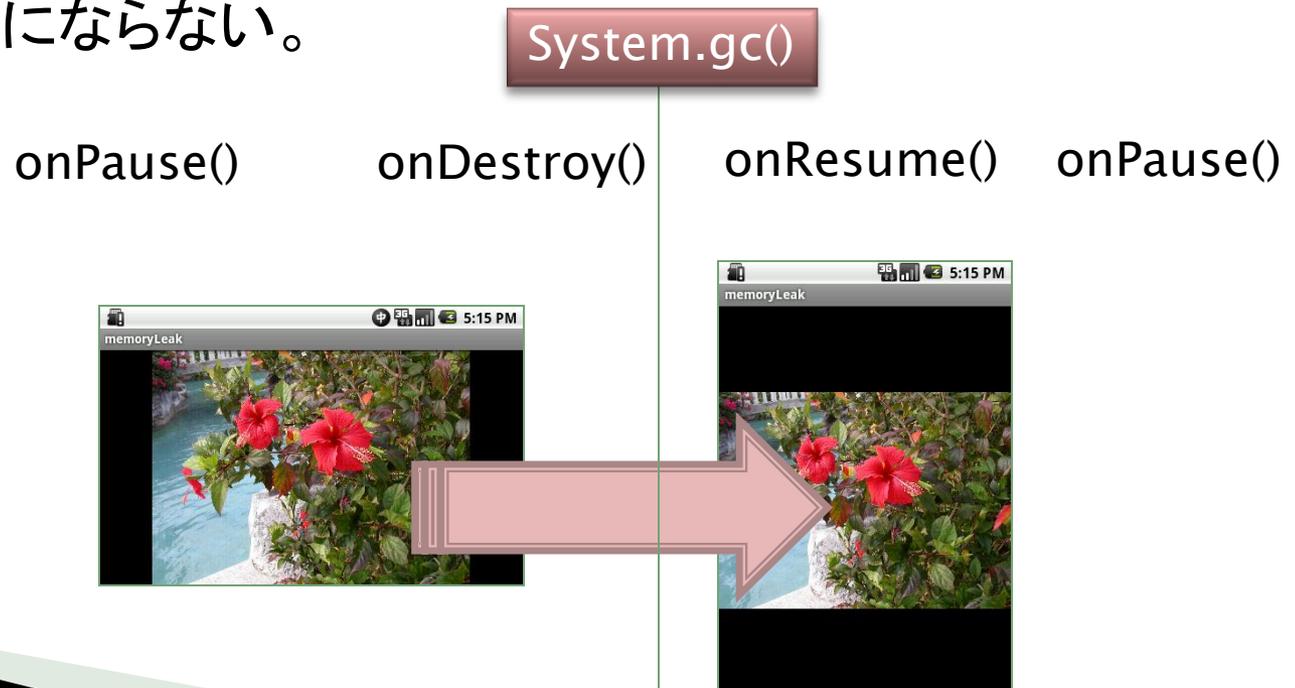
<http://g.oswego.edu/dl/html/malloc.html>

Bitmap.recycle()

- ▶ Free up the memory associated with this bitmap's pixels, and mark the bitmap as "dead", meaning it will throw an exception if `getPixels()` or `setPixels()` is called, and will draw nothing.
- ▶ `isRecycled()` Returns true if this bitmap has been recycled.
- ▶ 「今は使わない」というタイミングで呼んでおくとメモリが不足したときに自動的に解放してくれる

回転時にOutOfMemoryになった理由

- ▶ どうもBitmapのFinalizers (もしくはGC?) にバグがあるっぽい。
- ▶ そのFinalizersはGCが呼ばれたときに動く。
 - 画面回転時にSystem.gc()がCallされたら確かにOOMにならない。



詳細情報募集中...

OutOfMemory回避方法

Bitmapの取り扱い

- Recycle/IsRecycle を活用
- `BitmapFactory.Options.inPurgeable`
- `Bitmap.Config` に `ARGB_8888` をやめて `RGB_565` にするだけでメモリ使用量半分に。アルファブレンディングに利用できる `ARGB_4444` もOK。

メモリ管理一般

- 画像など大きなリソースはNativeHeapを利用。
GCが効かないので足りなくなったら諦める(OSの問題が大きい...)
- JavaHeapは使い終わったglobal変数などの参照を残さない
→ メモリリークしたり、GCの回収が遅れるケースが防げる

▶ ご清聴ありがとうございました