

Writing Real Time Games For Android

Chris Pruett May 2009



Did You See That Awesome Fade?

- Holy Crap!
 - The text was all sliding before that too.
 - How do they do that?!
- Right, this slide is about me, the presenter. Chris Pruett. That's me.
- I'm a Developer Advocate for Android.
 - That means I advocate development for Android. Please make something. Maybe I can help you.
 - I work in Japan. 宜しくお願いします。
- Before that, I wrote code for Lively.
- Prior to working at Google, I made video games.
 - I shipped about 10 titles for lots of platforms: GBA, PS2, PSP, Wii.



Making Games on Android

- Writing games is awesome, and Android is awesome, so writing games on Android must be awesome².
- This theory required testing. So I made a game. Many lessons learned.
- Topics to cover today:
 - Why games? Why Android? I mean, besides awesome².
 - Game engine architecture.
 - Writing Java code that is fast. For serious.
 - Drawing stuff on the screen efficiently.
 - Tips, tricks, and pitfalls.



Who Cares About Games on Mobile Devices?

- Dude, what rock have you been living under?
 - iPhone: 79% of users have downloaded at least one game. (According to a report by Compete, Inc.)
 - There are more than *100 million* **Nintendo DS** devices throughout the world. (According to Nintendo, see http://www.nintendo.co.jp/ir/pdf/ 2009/090507e.pdf)
 - Sony's **Playstation Portable** has just passed *50 million* devices (see: http://www.computerandvideogames.com/article.php?id=208211%3fcid)
 - The Nintendo Game Boy and Game Boy Advance together account for about 200 million devices (see http://www.nintendo.co.jp/ir/library/ historical_data/pdf/consolidated_sales_e0806.pdf).
- Portable games appeal to a huge audience, but traditional phones have not been good game devices.
- Game tech is extremely specific.
 - If your platform can support good video games, other apps should be a walk in the park.



Why Games on Android?

- Traditional PC and console game markets have been come so high-risk that only a few companies can even compete.
- Smaller games on non-traditional platforms are steadily gaining popularity with both traditional gamers and folks new to the medium.
 - See also: Nintendo Wii, iPhone, Flash, XBLA, etc.
 - Lower risk = more interesting and diverse content!
- Android provides an avenue for innovative games across a wide, internet-savvy audience.



Why This Game for Android?

- My goal is three-fold:
 - To produce a fun game for Android.
 - To produce a reusable, open source game engine to allow others to make fun games for Android.
 - To stress test our platform with regards to games; only publically-available code and tools are to be used.
- I went for an orthodox 2D side-scroller.
 - Parallax layers, tile-based worlds, animated sprites, etc.
 - Pushes all the right hardware buttons: input systems, OpenGL ES, sound, etc.
 - Proper game is feasible with one 20% engineer (that's me) for six months and 1 full time artist for four months.
 - Tools are pretty easy to write.
 - Popular and recently under-served genre.



Agenda 2: The Return

- Topics to cover today:
 - Why games? Why Android?
 - Game engine architecture.
 - Writing Java code that is fast.
 - Drawing stuff on the screen efficiently.
 - Tips, tricks, and pitfalls.

Insert Here: Picture of man holding giant gun that is also a chainsaw.



Quick Demo

(video goes here)

Note that this is a work in progress. All bugs are mine.

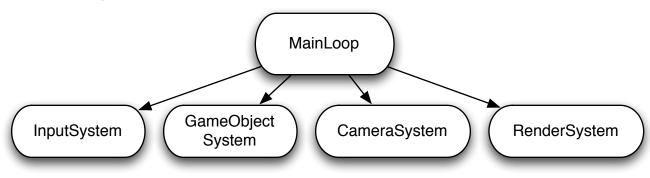


Game Engine Architecture

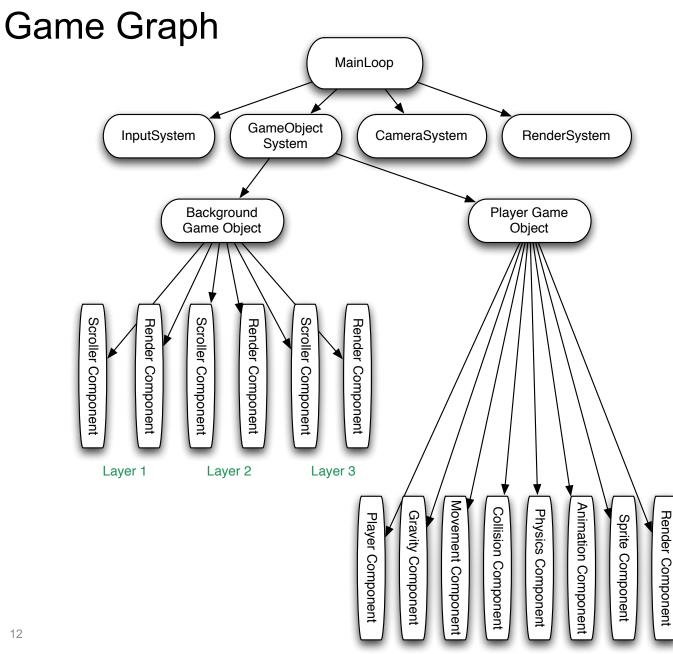
- Lots of approaches, but the basic problems are similar.
- My approach is a "game graph" that can be traversed every frame, taking time and motion events as input and resulting in a list of things to draw to the screen.
 - The root of the graph is the "main loop."
 - Children of the main loop get called once per frame.
 - Children further down the tree might get called once per frame, depending on their parent.
 - "Game objects" are children of a "game manager" node, which only visits children within a certain activity bubble around the camera.
 - Game objects themselves are sub-graphs of "game components," each implementing a single characteristic or feature of the object.



Game Graph









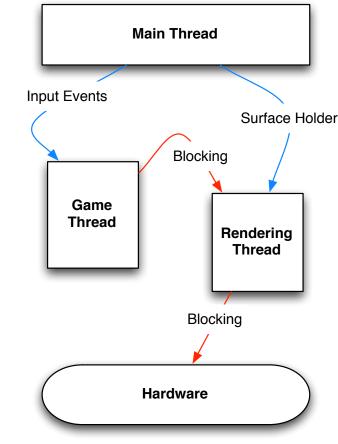
Game Engine Architecture

- At least, that's how I do it.
- Important point: time is passed to each node in the graph so that *framerate independent motion* is possible.
- Second important point: this system collects things to draw in a draw list each frame, *but it doesn't actually draw anything to the screen.*



Game Engine Architecture - Nice Threads, Yo

- I have three threads:
 - The main thread spawned by the Android activity.
 - Responsible for bootstrapping the game and receiving input events.
 - Mostly dormant.
 - The game thread.
 - Handles all non-rendering parts of the game: physics, AI, collision detection, animation, etc.
 - Owns the game graph.
 - The rendering thread.
 - Controlled by a SurfaceHolder.



Google 09

Agenda III: The Series Continues

- Topics to cover today:
 - Why games? Why Android?
 - Game engine architecture.
 - Writing Java code that is fast.
 - Drawing stuff on the screen efficiently.
 - Tips, tricks, and pitfalls.



I Love Coffee, I Love Tea

- I am pretty much a C++ engineer.
 - In fact, I wrote my first line of Java ever for this project.
 - So you should take my advice on the topic of Java-specific optimization with a grain of salt.
 - Still, I have done a lot of optimization work in the last six months, and maybe at a level that most Java apps do not require, so maybe I can offer some useful tidbits.
- Writing real-time games is an exercise in finding the perfect balance between flexibility and performance.
- My (non-language-specific) approach is:
 - Start with the simplest possible implementation, but design for future rework.
 - Choose flexibility over speed every day of the week... until the gameplay is damaged.
 - Profile early and constantly.



Step One: Memory Management

- Never allocate memory. Or release it.
 - Well, never allocate during gameplay.
 - The GC will stop your game for 100 ~ 300 ms. That's death for most real-time games.
- Revised: Allocate as much as possible up front, don't release things until you have natural pause time. Invoke the GC manually when you know it to be safe.
- Use DDMS to track allocations.
 - Hey, Java allocates memory CONSTANTLY. Ugh!
 - Hidden allocations in things like enum.values(), Class.getClassName(), Iterator, HashMap, Arrays.sort() etc etc etc.
 - Some of these are not really avoidable.



Allocation-Related Java Language Contortions

- Treat Java like C++
- Lots of the standard Java utility objects allocate memory.
 - Collections are out, as are iterators.
 - Forget about enums (they are really heavy-weight anyway).
 - Arrays.sort() and similar functions
 - Anything returning a String that needs to be read-only (like Class.getXXX(); man, I miss me some const).
- DDMS is your tool to name and blame.
- Better Java engineers than I might be able to supplement existing frameworks with non-allocating implementations.



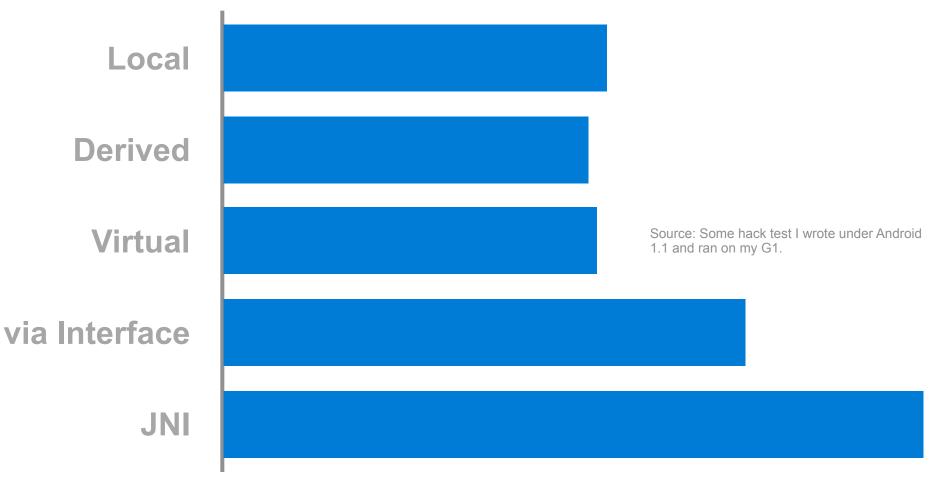
Step Two: Don't Call Functions

- Ok, that's extreme. But function calls are not cheap and you can't rely on inlining.
- Use static functions whenever possible.
- Don't call functions through an interface. 30% slower than regular virtual functions!
- Accessors and Mutators are my bestest friends in C++, but they have no place in your Java inner loop.
- Be wary of JNI functions.
 - In particular: lots of gl.glXX() functions.



Don't Call Functions: A Graph

• Take this with a grain of salt, not a very scientific test.





Step Three: Other Tips

- Use local variables, especially in inner loops.
- Use the final keyword on fields whenever you possibly can.
- Some hardware (like the G1) has no FPU, so avoid float math.
- Always use Log.d() or similar rather than System.out.print(). Printing takes time!
- Use Traceview!

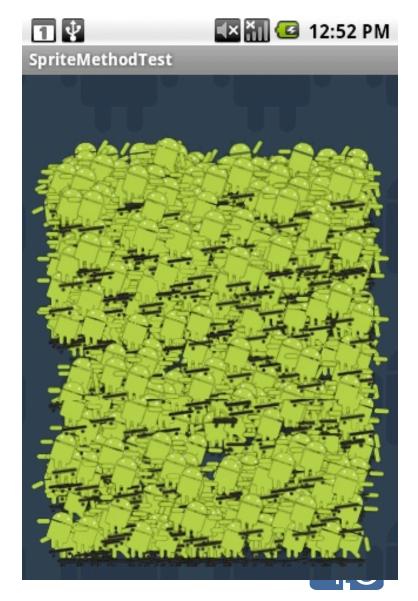
	Traceview: /Users/pruett/andou.trace													
	msec: 3	msec: 3,572.809 max msec: 3,824												324
									_					
	3,560	3,580	3,600	3,620	3,640	3,660	3,680	3,700	3,72	3,740	3,760	3,780	3,800	3,820
[3] main														
[21] Game			ed des	1.0-1		148		and the	eine.		dilloute			1.1.1
[17] Binder Thread #4						1							1	+
[15] Binder Thread #3	-													
[11] Binder Thread #1														
[19] GLThread						210.0								
[13] Binder Thread #2														
Name										Inclusive	Excl %		s+Recur Calls;	
1 com/android/andouku							yv.		47.9%	5380.190	1.9%	217.520	63+744	6.
1 com/android/andouku 2 com/android/andouku	n/MainLoop	update (Fi	.com/andre	oid/andoui	kun/BaseOl)V		47.9% 47.9%	5380.190 5380.156	1.9%	217.520 2.653	63+744 62+0	6. 86.
1 com/android/andouku 2 com/android/andouku 3 com/android/andouku	n/MainLoop n/GLSurfac	eView\$GLT	.com/andri hread.guar	oid/andoui dedRun ()\	kun/BaseOl	bject;)V			47.9% 47.9% 37.7%	5380.190 5380.156 4233.276	1.9% 0.0% 0.3%	217.520 2.653 31.495	63+744 62+0 1+0	6. 86. 4233.
1 com/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb	eViewSGLT jectManage	.com/andre hread.guar r.update (F	oid/andoui dedRun ()V FLcom/and	kun/BaseOl / roid/andou	bject:)V ikun/BaseO	bject;)V		47.9% 47.9% 37.7% 34.5%	5380.190 5380.156	1.9%	217.520 2.653	63+744 62+0	6. 86. 4233. 62.
1 com/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku 5 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameRei	o.update (Fl eViewSGLT jectManage nderer.drav	.com/andre hread.guar r.update (F Frame (Lja	oid/andoui dedRun ()V FLcom/and	kun/BaseOl / roid/andou	bject:)V ikun/BaseO	bject;)V	v	47.9% 47.9% 37.7% 34.5% 30.2%	5380.190 5380.156 4233.276 3869.902	1.9% 0.0% 0.3% 1.3%	217.520 2.653 31.495 145.653	63+744 62+0 1+0 62+0	6. 86. 4233. 62. 15.
1 com/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku 5 com/android/andouku 6 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameRei n/GameRei	eView\$GLT jectManage nderer.drav kground.dr.	.com/andre hread.guar r.update (F vFrame (Lja aw (FF)V	oid/andoui dedRun () FLcom/and avax/micro	kun/BaseOl / roid/andou edition/khr	bject:)V ikun/BaseO ronos/open	bject;)V gles/GL10;)	v	47.9% 47.9% 37.7% 34.5% 30.2% 13.5%	5380.190 5380.156 4233.276 3869.902 3386.536	1.9% 0.0% 0.3% 1.3% 2.8%	217.520 2.653 31.495 145.653 315.181	63+744 62+0 1+0 62+0 225+0	6. 86. 4233. 62. 15. 3.
1 com/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku 5 com/android/andouku 6 com/android/andouku 7 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameRei n/TiledBac n/RenderS	o.update (Fi eView\$GLT jectManage nderer.draw kground.dr. ystem.swap	.com/andri hread.guar r.update (F vFrame (Lja tw (FF)V (Lcom/and	oid/andoui dedRun () FLcom/and avax/micro	kun/BaseOl / roid/andou edition/khr	bject:)V ikun/BaseO ronos/open	bject;)V gles/GL10;)	v	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611	1.9% 0.0% 0.3% 1.3% 2.8% 7.3%	217.520 2.653 31.495 145.653 315.181 814.608	63+744 62+0 1+0 62+0 225+0 450+0	6. 86. 4233. 62. 15. 3. 22.
1 com/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku 5 com/android/andouku 7 com/android/andouku 8 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameRei n/TiledBac n/RenderSy n/Drawable	o.update (R eView\$GLT jectManage nderer.drav kground.dr. ystem.swap eBitmap.dra	.com/andri hread.guar r.update (F wFrame (Lja aw (FF)V (Lcom/and w (FF)V	oid/andoui dedRun () FLcom/and avax/micro	kun/BaseOl / roid/andou edition/khr	bject:)V ikun/BaseO ronos/open	bject;)V gles/GL10;)	v	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5% 11.0%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611 1402.007	1.9% 0.0% 0.3% 1.3% 2.8% 7.3% 0.0%	217.520 2.653 31.495 145.653 315.181 814.608 5.268	63+744 62+0 1+0 62+0 225+0 450+0 63+0	6. 86. 4233. 62. 15. 3. 22. 0.
com/android/andouku com/android/andouku com/android/andouku com/android/andouku com/android/andouku com/android/andouku com/android/andouku com/android/andouku s com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameRei n/TiledBaci n/RenderSi n/Drawable n/FixedSize un/Phased	o.update (F eView\$GLT jectManage nderer.drav kground.dra stem.swap eBitmap.dra Array.sort ObjectMana	.com/andri hread.guar r.update (F vFrame (Lja tw (FF)V (Lcom/and tw (FF)V (Z)V ger.commi	oid/andoui dedRun ()V FLcom/and avax/micro droid/ando itUpdates (kun/BaseOl / roid/andou edition/khr ukun/Gam)V	bject;)V ikun/BaseO ronos/open eRenderer;	bject;)V gles/GL10;))V	v	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5% 11.0% 9.3% 9.3%	\$380.190 \$380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417	1.9% 0.0% 0.3% 1.3% 2.8% 7.3% 0.0% 5.9% 0.0% 0.3%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322	63+744 62+0 1+0 62+0 225+0 450+0 63+0 6075+0 187+0 1614+0	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0.
com/android/andouku com/android/andouku com/android/andouku som/android/andouku som/android/andouku som/android/andouku som/android/andouku som/android/andouku som/android/andouku som/android/andouku som/android/andouku toom/android/andouku toom/android/andouku toom/android/andouku toom/android/andouku	n/MainLoo; n/GLSurfac n/GameOb n/GameRei n/TiledBaci n/RenderSj n/Drawabie n/FixedSize un/FixedSize un/FixedSize	b.update (Fl eViewSGLT jectManage nderer.drav kground.dr. ystem.swap 28itmap.dra 28itmap.dra 28itmap.dra 26itmap.dra 26itmap.dra 26itmap.dra 26itmap.dra 26itmap.dra 26itmap.dra 26itmap.dra 20itmap.d	Lcom/andri hread.guar r.update (F i/Frame (Lja aw (FF)V (Lcom/and w (FF)V (Z)V iger.commi i/Contents ()	oid / andoui dedRun ()V FLcom/and avax/micro droid / ando itUpdates ([Ljava/lang	kun/BaseOl / roid/andou edition/khr ukun/Gam ukun/Gam)V //Object;ILJ	bject;)V ikun/BaseO ronos/open eRenderer; ava/util/Co	ibject;)V gles/GL10;))V mparator;)V	v	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5% 11.0% 9.3% 9.3%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114	1.9% 0.0% 0.3% 1.3% 2.8% 7.3% 0.0% 5.9% 0.0% 0.3% 0.2%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322 22.867	63+744 62+0 1+0 62+0 225+0 450+0 63+0 6075+0 187+0 1614+0 125+0	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 8.
com/android/andouku 10 com/android/andouku 11 com/android/andouku 12 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameRei n/TiledBac n/RenderSy n/Drawabli n/FixedSize un/FixedSiz un/FixedSiz un/FixedSiz	b.update (Fl eViewSGLT jectManage nderer.draw kground.dr. ystem.swap eBitmap.dra Array.sort ObjectMana zeArray.sor zeArray.insi	.com/andri hread.guar r.update (F wFrame (Lja tw (FF)V (Lcom/and w (FF)V (Z)V (Z)V ger.commi (Contents (ertionSort ()	oid / andoui ded Run ()\ FLcom/and avax/micro droid / ando itUpdates ((Ljava/lang Ljava/lang	kun/BaseOl / roid/andou edition/khr ukun/Gam /V /Object:ILj /Object:ILj	bject;)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co	bject;)V gles/GL10;))V mparator;)V omparator;)V	v	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5% 11.0% 9.3% 9.3% 9.3% 9.1%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114 1025.512	1.9% 0.0% 0.3% 1.3% 2.8% 7.3% 0.0% 0.3% 0.0% 0.3% 0.2% 1.6%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322 22.867 184.652	63+744 62+0 1+0 62+0 225+0 450+0 63+0 6075+0 187+0 187+0 1614+0 125+0 1508+0	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 8. 0.
Loom/android/andouku 2 com/android/andouku 2 com/android/andouku 3 com/android/andouku 5 com/android/andouku 6 com/android/andouku 6 com/android/andouku 8 com/android/andouku 9 com/android/andouku 9 com/android/andouku 10 com/android/andouku 11 com/android/andouku 12 com/android/andouku 13 com/android/andouku 13 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameRei n/TiledBac n/RenderSy n/Drawable n/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz	b.update (Fl eView\$GLT jectManage nderer.draw kground.dr. sstem.swap e8itmap.dra Array.sort ObjectMana zeArray.sor zeArray.insi bundCollisio	.com/andri hread.guar r.update (F wFrame (Lja tw (FF)V (Lcom/and tw (FF)V (Z)V (Z)V gger.commi (Contents (rtionSort () nCompone	oid /andoui dedRun ()\ FLcom/and avax/micro droid /ando ltUpdates ([Ljava/lang nt.update	kun/BaseOl / roid/andou edition/khr ukun/Gam)V //Object:IILj /Object:IILj /Dbject:IIL	bject;)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co	bject;)V gles/GL10;))V mparator;)V omparator;)V	v	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5% 11.0% 9.3% 9.3% 9.3% 9.1%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114 1025.512 1020.295	1.9% 0.0% 0.3% 1.3% 7.3% 0.0% 5.9% 0.0% 0.3% 0.2% 1.6% 0.9%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322 22.867 184.652 99.332	63+744 62+0 1+0 62+0 225+0 450+0 63+0 6075+0 187+0 1614+0 125+0 1508+0 248+0	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 8. 0. 4.
Lom/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku 5 com/android/andouku 6 com/android/andouku 7 com/android/andouku 9 com/android/andouku 9 com/android/andouku 10 com/android/andouku 11 com/android/andouku 12 com/android/andouku 13 com/android/andouku 13 com/android/andouku 14 com/android/andouku 14 com/android/andouku	n/MainLoop n/GLSurfac n/GameRei n/TiledBacl n/TiledBacl n/FixedSize un/FixedSize un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz	b.update (Fl eViewSGLT jectManage nderer.draw kground.dr. stem.swap Bitmap.dra Array.sort ObjectMana teArray.sort objectMana teArray.sort objectMana teArray.sort oundCollisio	.com/andri hread.guar r.update (F vFrame (Lja aw (FF)V (Lcom/and w (FF)V (Z)V ger.commi (Contents () ertionSort () nCompone te ()Ljava/li	oid/andoui dedRun ()\ FLcom/and avax/micro droid/ando itUpdates ([Ljava/lang [Ljava/lang nt.update i ang/Object	kun/BaseOl / roid/andou edition/khr ukun/Gam)V //Object:IILj /Object:IILj /Dbject:IIL	bject;)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co	bject;)V gles/GL10;))V mparator;)V omparator;)V	v	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5% 11.0% 9.3% 9.3% 9.3% 9.1% 8.0%	5380.190 5380.156 4233.276 33869.902 33865.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114 1025.512 1020.295 898.311	1.9% 0.0% 0.3% 1.3% 7.3% 0.0% 5.9% 0.0% 0.3% 0.2% 1.6% 0.9% 1.0%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322 22.867 184.652 99.332 107.588	63+744 62+0 1+0 62+0 225+0 450+0 63+0 63+0 1675+0 187+0 1614+0 125+0 1508+0 248+0 6231+0	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 5. 0. 8. 0. 4. 0.
Loom/android/andouku 2 com/android/andouku 2 com/android/andouku 3 com/android/andouku 5 com/android/andouku 6 com/android/andouku 6 com/android/andouku 7 com/android/andouku 9 com/android/andouku 9 com/android/andouku 10 com/android/andouku 10 com/android/andouku 11 com/android/andouku 12 com/android/andouku 12 com/android/andouku 13 com/android/andouku 14 com/android/andouku 15 com/android/andouku 15 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameRein n/TiledBac n/FixedSize un/FixedSize un/FixedSize un/FixedSize un/FixedSize un/FixedSize un/FixedSize un/FixedSize un/FixedSize un/FixedSize un/TObjectF	b.update (Fi eViewSGLT jectManage nderer.draw kground.dr. vstem.swap Bitmap.dra Array.sort cArray.sort ceArray.sort ceArray.insi ceArray.insi ceArray.lnsi ceArra	.com/andri hread.guar r.update (F vFrame (Lja aw (FF)V (Lcom/and w (FF)V (Z)V ger.commi (Contents () ertionSort () nCompone te (Ljava/la ()Ljava/la	oid/andoui dedRun ()\ FLcom/and wax/micro droid/ando itUpdates ([Ljava/lang [Ljava/lang nt.update i ang/Object;	kun/BaseOl / roid/andou edition/khr ukun/Gam)V //Object:IILj /Object:IILj /Dbject:IIL	bject;)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co	bject;)V gles/GL10;))V mparator;)V omparator;)V	v	47.9% 47.9% 37.7% 34.5% 30.2% 11.0% 9.3% 9.3% 9.3% 9.1% 8.0% 7.0%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114 1025.512 1020.295 898.311 790.723	1.9% 0.0% 0.3% 2.8% 7.3% 0.0% 5.9% 0.0% 0.3% 0.2% 1.6% 0.9% 1.0% 3.8%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322 22.867 184.652 99.332 107.588 425.513	63+744 62+0 1+0 62+0 225+0 63+0 6075+0 187+0 187+0 1614+0 125+0 1508+0 248+0 6231+0 6231+0	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 8. 0. 4. 0. 0. 0. 0. 0.
1 com/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku 5 com/android/andouku 6 com/android/andouku 8 com/android/andouku 9 com/android/andouku 10 com/android/andouku 11 com/android/andouku 12 com/android/andouku 12 com/android/andouku 13 com/android/andouku 14 com/android/andouku 5 com/android/andouku 5 com/android/andouku 5 com/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameOb n/FixedBac n/FixedSize un/FixedSize un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/GLSurfa	b.update (Fl eViewSGLT jectManage kground.dr. vstem.swap eBitmap.dra eArray.sort ceArray.sort zeArray.insi oundCollisio tPool.allocat ceViewSEgi	Lcom/andri hread.guar r.update (F krame (Lja tw (FF)V (Lcom/and tw (FF)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z	oid/andoui dedRun ()V FLcom/and avax/micro droid/ando itUpdates ((Ljava/lang nt.update i ang/Object; sp ()Z	kun/BaseOl / roid/andou edition/khr ukun/Gam / //Object:ILj /Object:ILj /Object:ILj /FLcom/and	bject:)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co droid/andoi	Ibject;)V gles/GL10;) W mparator;)V omparator;)u ukun/BaseO	V V bject:	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5% 11.0% 9.3% 9.3% 9.3% 9.1% 9.1% 8.0% 7.0%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114 1025.512 1020.295 898.311 790.723 788.515	1.9% 0.0% 0.3% 1.3% 2.8% 7.3% 0.0% 5.9% 0.3% 0.2% 1.6% 0.9% 1.6% 3.8% 0.2%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322 22.867 184.652 99.332 107.588 425.513 16.848	63+744 62+0 1+0 62+0 225+0 63+0 6075+0 1614+0 125+0 1508+0 248+0 6231+0 225+0	6. 86. 4233. 62. 15. 22. 0. 5. 0. 8. 0. 4. 0. 3.
Com/android/andouku Com/android/andouku Com/android/andouku Com/android/andouku Com/android/andouku Som/android/andouku S	n/MainLoo; n/GLSurfac n/GameOb; n/GameRei n/TiledBac n/FixedSize un/Phased un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/ExcSurfa gles_jn/EC	b.update (Fi eView\$GLT jectManage nderer.draw kground.dr. rstem.swap aBitmap.dra Akrray.sort ObjectMana teArray.sort ObjectMana teArray.insi undCollisio tPool.allocat ceView\$Egi CLImpl.egiSv	.com/andri hread.guar r.update (F wFrame (Lja w (FF)V (Lcom/and w (FF)V (Z)V uger.commi (Contents () ertionSort () nCompone te (Ljava/la Helper.swa wapBuffers	oid/andoui dedRun ()V FLcom/and iwax/micro droid/ando itUpdates ((Ljava/lang (Ljava/lang (Ljava/lang (Ljava/lang object; ap OZ (Ljavax/mi	kun/BaseOl roid/andou edition/khr ukun/Gam V/Object;IILj /Object;IIIL FLcom/and t; croedition/	bject:)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co droid/andou /khronos/eg	bject;)V gles/GL10;))V mparator;)V omparator;) ukun/BaseO gl/EGLDispla	v v bject: sy;Lja:	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 12.5% 11.0% 9.3% 9.3% 9.3% 9.1% 9.1% 9.1% 8.0% 7.0% 6.8%	\$380.190 \$380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1049.067 1047.417 1045.114 1025.512 1020.295 \$98.311 790.723 788.515 763.457	1.9% 0.0% 0.3% 1.3% 2.8% 7.3% 0.0% 5.9% 0.0% 0.3% 0.2% 1.0% 3.8% 0.2% 1.0% 3.8%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322 22.867 184.652 99.332 107.588 425.513 425.513 6.848 763.457	63+744 62+0 1+0 62+0 225+0 63+0 6075+0 187+0 1508+0 1508+0 1508+0 248+0 6231+0 6231+0 6231+0 225+0 225+0	6. 86. 4233. 62. 15. 3. 222. 0. 5. 0. 8. 0. 4. 0. 4. 0. 3. 3.
Com, and roid / and ouku Som, and roid / and ouku Som / and roid / and ouku Som, and r	n/MainLoop n/GLSurfac n/GameOb n/GameOb n/TiledBaci n/TiledBaci n/Drawabla n/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/Backgro un/GbjectF un/GLSurfa (gles_jni/EC un/Collisor	b.update (Fi eView\$GLT jectManage nderer.drav kground.dr. ystem.swap Bitmap.dra Array.sort ObjectMana teArray.sort zeArray.sor zeArray.insi undCollisio zeArray.insi vool.allocate ceView\$Egi iLImp1.egISi System.caa	icom/andri hread.guar r.update (F iframe (Lja tw (FF)V (Lcom/and w (FF)V [Z)V (Contents (Contents	oid/andoui dedRun (Vi FLcom/and ivvax/micro iroid/ando itUpdates ([Ljava/lang Ljava/lang Ljava/lang Jobject; apg/Object; ap (OZ (Ljavax/mi /android//	kun/BaseOl / roid /andou edition/khr ukun/Gam //Object:IILJ //Object:IILJ //Object:IILJ rocoedition/ andoukun/	bject:)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co droid/andor /khronos/eg Vector2;Lcc	bject;)V gles/GL10;))V mparator;)V omparator;) ukun/BaseO gl/EGLDispla	v v bject: sy;Lja:	47.9% 47.9% 37.7% 34.5% 30.2% 13.5% 11.0% 9.3% 9.3% 9.3% 9.1% 8.0% 7.0% 6.8% 6.5%	\$380.190 \$380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1049.067 1047.417 1045.114 1025.512 1020.295 \$98.311 790.723 788.515 763.457 727.203	1.9% 0.0% 0.3% 2.8% 7.3% 0.0% 0.3% 5.9% 0.0% 0.2% 1.6% 0.2% 6.8% 0.2%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 3.953 32.322 22.867 184.652 99.332 107.588 425.513 16.848 763.457 58.773	63+744 62+0 1+0 62+0 225+0 63+0 63+0 1614+0 125+0 1614+0 125+0 248+0 6231+0 6231+0 225+0 225+0 478+0	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 8. 0. 4. 0. 0. 8. 0. 0. 3. 3. 1.
Com/android/andouku Com/android/andouku Com/android/andouku Com/android/andouku Com/android/andouku Som/android/andouku Som/android/andouku Som/android/andouku Som/android/andouku Com/android/andouku Com/android/andouku Com/android/andouku Som/android/andouku S	n/MainLoopn/GLSurfac n/GameOb n/GameOb n/FameOb n/FixedSize n/FixedSize un/FixedSize un/FixedSize un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/Collision un/FixedSiz un/Collision un/FixedSiz	b.update (F) eViewSGLT jectManage ddere.draw kground.dr. sstem.swap BBitmap.dra Array.sort ObjectMana teArray.sort ObjectMana teArray.sort ceArray.sort ceArray.sort ceArray.sort ceArray.sort ceViewSEgi LUmpi.egiSv sSystem.cast tem.update	icom/andri hread.guar r.update (F frame (Li) kw (FF)V (Lcom/and w (FF)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z	oid/andoui dedRun () FLcom/and ivvax/micro droid/ando itUpdates ([Ljava/lang [Ljava/lang [Ljava/lang nt.update i ang/Object ang/Object; pp O2 (Ljavax/mi n/android/a droid/and droid/and	kun/BaseOl / roid /andou edition/khr ukun/Gam)V /Object:ILj /Object:ILj /Object:ILj /Object:ILj /Object:ILj /Coje	bject:)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co java/util/Co iava/util/Co	bject;)V gles/GL10;) VV omparator;)V omparator;) ukun/BaseO gl/EGLDispla m/android;	v v bject: sy;Lja:	47.9% 47.9% 37.7% 34.5% 30.2% 9.3% 9.3% 9.3% 9.1% 8.0% 7.0% 7.0% 6.8% 6.5% 6.2%	\$380.190 \$380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1047.417 1045.114 1025.512 1020.295 \$98.311 790.723 788.515 763.457 727.203 694.820	1.9% 0.0% 0.3% 2.8% 7.3% 0.0% 0.3% 0.2% 0.3% 0.2% 0.9% 1.6% 0.9% 1.6% 0.2% 6.8% 0.2% 6.8% 0.8%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.305 3.953 32.322 22.867 184.652 99.332 107.588 425.513 16.848 763.457 58.773 90.069	63+744 62+0 1+0 62+0 225+0 63+0 6075+0 187+0 1614+0 1508+0 6231+0 6231+0 6231+0 6231+0 6231+0 6231+0 63+0	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 5. 0. 8. 8. 0. 4. 0. 0. 3. 3. 1. 1.
Com/android/andouku 2 com/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku 5 com/android/andouku 7 com/android/andouku 9 com/android/andouku 10 com/android/andouku 10 com/android/andouku 11 com/android/andouku 11 com/android/andouku 12 com/android/andouku 15 com/android/andouku 16 com/android/andouku 10 com/android/andouku	n/MainLoo; n/GLSurfac n/GameOb n/GameOb n/GameOb n/FixedBiz un/FixedSiz un/Fix	5. update (F) EvViewSGLT jectManage(etManage) etManage) eBitmap.dra stem.swap.sort ObjectMana teArray.sort ObjectMana teArray.sort ObjectMana teArray.sort ObjectMana teArray.sort ceViewSEgi Limpl.egiS stystem.cas tem.update omponentu.	Lcom/andru hread.guar , rupdate (f vFrame (Lja w (FF)V (Lcom/and w (FF)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z	oid/andoui dedRun (V Lcom/and ivax/micro droid/ando itUpdates ([Ljava/lang ILjava/lang nt.update (ang/Object; ap ()2 (Ljavax/mi /and roid / adroid/and orn/android /	kun/BaseOl / oroid /andou edition/khr ukun/Gam //Object:ILJ	bject.)V įkun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co java/util/Co droid/andou /khronos/eg Vector2;Lcc eObject;)V in/BaseObje	bject;)V gles/GL10;))V mparator;)V omparator;) ukun/BaseO gl/EGLDispla m/android; ect;)V	/ V bject: ay:Lja: /andc	47.9% 47.9% 37.7% 34.5% 30.2% 9.3% 9.3% 9.3% 9.3% 9.3% 9.1% 8.0% 7.0% 6.5% 6.5% 6.5%	\$380.190 \$380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114 1025.512 1020.295 898.311 790.723 788.515 763.457 727.203 694.820 621.830	1.9% 0.0% 0.3% 2.8% 7.3% 0.0% 5.9% 0.3% 0.2% 1.0% 3.8% 0.2% 6.8% 0.5% 0.5% 0.5%	217.520 2.653 31.495 145.653 315.181 81.4.608 5.268 663.305 3.2.322 22.867 184.652 99.332 107.588 425.513 16.848 763.457 783.477 58.773 90.069	63+744 62+0 1+0 62+0 63+0 63+0 637+0 637+0 1614+0 125+0 1508+0 248+0 6231+0 6231+0 6231+0 6231+0 6231+0 6231+0 63+0 478+0 63+0 478+0 63+0 478+0 63+0 478+0 63+0 478+0 63+0 63+0 63+0 63+0 63+0 63+0 63+0 63	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 5. 0. 5. 0. 4. 0. 0. 3. 3. 1. 1. 1.
Com/android/andouku Com/android/andouku Com/android/andouku Com/android/andouku Com/android/andouku Scom/android/andouku	n/MainLoop n/GLSurfac n/GameOb n/GameOb n/GameOb n/FixedSiz un/PixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/FixedSiz un/GbjectF un/GLSurfa (gles_jn/IEG un/Collisio un/Collisio un/Collisio un/SpriteC un/PixedSi	Jupdate (F) eViewSGLT jectManage jectManage derer. draw kground.dr. system.swap Bitmap.dra Asrray.sort DbjectMana evArray.sort DbjectMana isystem.cas tem.update omponent.L DbjectMana	Lcom/andri hread.guar -rupdate (f kFrame (Lja kw (FF)V (Lcom/and w (FF)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z)V (Z	oid/andoui dedRun (V Ecom/and avax/micro droid/ando itUpdates ((Ljava/lang (Ljava/lang (Ljava/lang (Ljava/lang (Dbject; ng/Object; (Ljavax/mi (Jandroid/) droid/and om/android/ adObjectCo	kun/BaseOl / oroid /andou edition/khr ukun/Gam //Object:ILJ	bject.)V įkun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co java/util/Co droid/andou /khronos/eg Vector2;Lcc eObject;)V in/BaseObje	bject;)V gles/GL10;))V mparator;)V omparator;) ukun/BaseO gl/EGLDispla m/android; ect;)V	/ V bject: ay:Lja: /andc	47.9% 47.9% 37.7% 34.5% 13.5% 12.5% 11.0% 9.3% 9.3% 9.3% 9.1% 8.0% 7.0% 6.8% 6.5% 6.5% 5.3%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114 1025.512 1020.295 898.311 790.723 788.515 763.457 727.203 694.820 621.830 595.972	1.9% 0.0% 0.3% 2.8% 7.3% 0.0% 0.2% 0.2% 0.2% 0.9% 0.2% 0.9% 0.2% 0.8% 0.2% 0.8% 0.2% 0.8% 0.8% 0.8% 0.8%	217.520 2.653 31.495 145.653 315.181 814.608 5.268 663.306 663.306 663.306 3.953 32.322 22.867 184.652 99.332 22.867 184.652 99.332 107.588 763.457 58.773 90.069 114.228	63+744 62+0 1+0 62+0 225+0 6075+0 63+0 6075+0 187+0 1614+0 125+0 125+0 125+0 125+0 6231+0 72550 6231+0 72550 6231+0 72550 72500 72500 72500 72500 725500 725500 725500 725000 725000 725000 725000 7250000000000	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 8. 0. 4. 0. 0. 3. 3. 1. 11. 11. 0. 0.
Com/android/andouku 2 com/android/andouku 2 com/android/andouku 3 com/android/andouku 4 com/android/andouku 5 com/android/andouku 7 com/android/andouku 9 com/android/andouku 10 com/android/andouku 10 com/android/andouku 11 com/android/andouku 11 com/android/andouku 12 com/android/andouku 15 com/android/andouku 16 com/android/andouku 10 com/android/andouku	n/MainLoo; n/GLSurfac n/GLAME n/GameOb n/GameOb n/FixedBize un/FixedSize un/FixedSize un/FixedSiz un/FixedSize un/FixedSiz un/FixedSiz un/FixedSiz un/Fobject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/Colject un/FixedSiz un/FixedSiz un/Colject un/FixedSiz	b.update (F) b.update (F) evtManagetextMana	Lcom/andri hread.guar ,rupdate (f %Frame (Lja ww (FF)V (Lcom/ane) (Lcom/ane) (Contents () rrtionSort () ncCompone te (Ljava/la Helper.swa vapBuffers tRay (Lcom (FLcom/ane) pdate (FLc (FLcom/ane) pdate (FLc Wrax/DS)	oid/andoui dedRun (V Eccom/andi ivax/micro iroid/ando itUpdates ([Ljava/lang ILjava/lang ILjava/lang ILjava/lang ILjava/lang ILjava/lang ILjava/lang ILjava/lang ILjava/minoid/ idoobjectCo FFFFFIV	kun/BaseOl / oroid / andou edition/khr ukun/Gam //Object:IILj //Object:IIL //Object:IIL //Diject:IIL //Diject:IIL //Diject:IIL croedItion/ andoukun/ andoukun/ andoukun/ andoukun/ andoukun/	bject.)V ikun/BaseO ronos/open eRenderer; ava/util/Co java/util/Co java/util/Co droid/andou /khronos/eş Vector2;Lcc eObject;V in/BaseObject compare (Lj	bject;)V gles/GL10;))V mparator;)V omparator;) ukun/BaseO gl/EGLDispla m/android; ect;)V	/ V bject: ay:Lja: /andc	47.9% 47.9% 37.7% 34.5% 30.2% 9.3% 9.3% 9.3% 9.3% 9.3% 9.1% 8.0% 7.0% 6.5% 6.5% 6.2% 5.5%	5380.190 5380.156 4233.276 3869.902 3386.536 1510.611 1402.007 1238.422 1049.067 1047.417 1045.114 1025.512 1020.295 898.311 790.723 788.515 763.457 727.203 694.820 621.830 595.972	1.9% 0.0% 0.3% 2.8% 7.3% 0.0% 5.9% 0.3% 0.2% 1.0% 3.8% 0.2% 6.8% 0.5% 0.5% 0.5%	217.520 2.653 31.495 145.653 315.181 81.4.608 5.268 663.305 3.2.322 22.867 184.652 99.332 107.588 425.513 16.848 763.457 783.477 58.773 90.069	63+744 62+0 1+0 62+0 63+0 63+0 637+0 637+0 1614+0 125+0 1508+0 248+0 6231+0 6231+0 6231+0 6231+0 6231+0 6231+0 63+0 478+0 63+0 478+0 63+0 478+0 63+0 478+0 63+0 478+0 63+0 63+0 63+0 63+0 63+0 63+0 63+0 63	6. 86. 4233. 62. 15. 3. 22. 0. 5. 0. 5. 0. 5. 0. 4. 0. 0. 3. 3. 1. 1. 1.



Agenda Part 4: Even More Agenda

- Topics to cover today:
 - Why games? Why Android?
 - Game engine architecture.
 - Writing Java code that is fast.
 - Drawing stuff on the screen efficiently.
 - Tips, tricks, and pitfalls.

Screenshot from SpriteMethodTest, 1000 sprites, OpenGL DrawTexture extension. Runs at around ~10 fps on the G1.

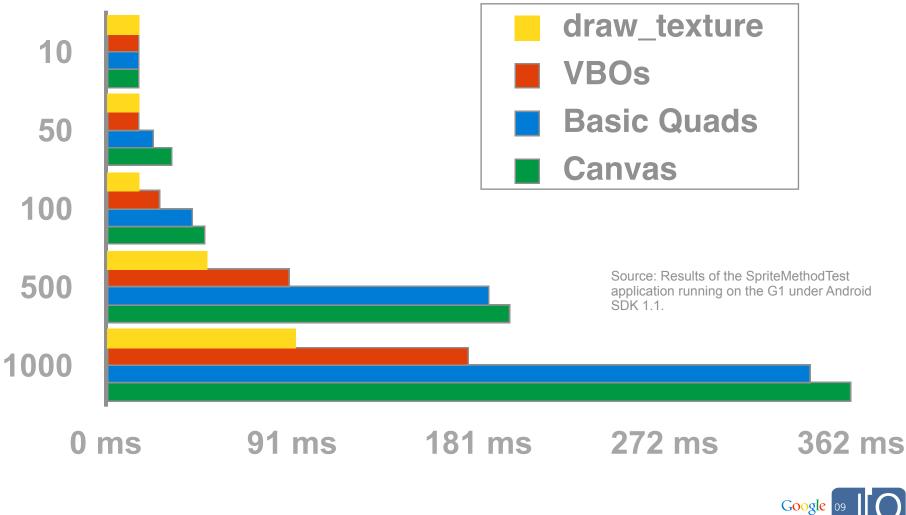


Android Drawing Methods

- Canvas:
 - CPU-based 2D drawing. Used for most of the Android UI.
 - Fast for a small number of blits. (~ 10 sprites < 16 ms in my tests)
 - Very straightforward and easy to use.
- OpenGL ES
 - 2D and 3D drawing.
 - Hardware accelerated on some platforms (like the G1).
 - Scales to much more complex scenes than Canvas.
 - Various 2D drawing methods:
 - Quads with orthographic projection
 - VBO quads (on supported platforms)
 - draw_texture extension (on supported platforms)
 - Only OpenGL ES 1.0 is guaranteed.



OpenGL vs Canvas for 2D drawing (G1)



Which Method is Best?

- Clearly, OpenGL ES + the draw_texture extension is fastest for 2D drawing on the G1.
 - But that extension isn't guaranteed to be supported on all platforms. You MUST check glGetString(GL10.GL_EXTENSIONS) before using it.
- However, Canvas isn't bad if...
 - You have very few things to draw every frame, or
 - You don't have to draw every frame (puzzle games, etc).
- SpriteMethodTest provides a framework for swapping between drawing methods (and timing them) based on my game code.
 - <u>http://code.google.com/p/apps-for-android/</u>



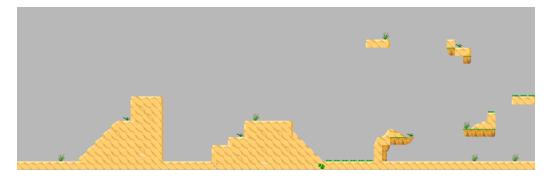
- Replica Island uses three background layers: one large static image, one mid ground tile-map, and the foreground tile map.
- The tile map layers are regular grids of 32x32 tiles. That means 150 tiles to draw per layer on any given frame (worst case).
- More layers would be nice, but drawing the background is the single most expensive operation in the Replica Island code.
 - Actually, the single static image is quick. It's just one 512x512 texture.
 - It's the tile maps that eat frame time, either on the CPU or the GPU, depending on the drawing method.



Layered Parallax Backgrounds

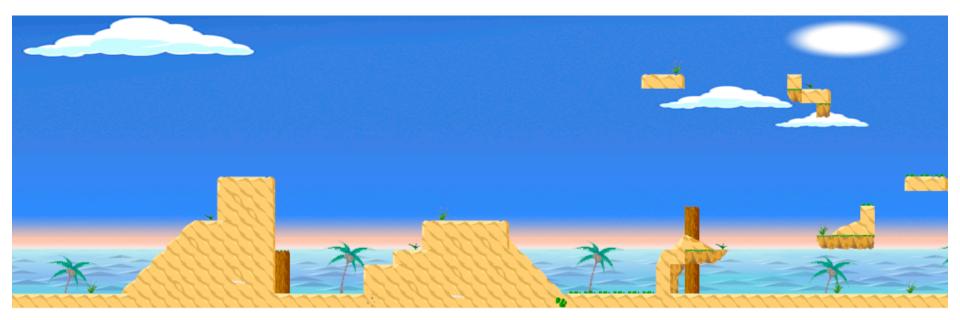








Layered Backgrounds Composited





- First idea: use a single atlas texture as the tile map, use draw_texture to draw, and adjust cropping of the texture to select individual tiles.
 - Only one glBindTexture() call needed, which is good.
 - Otherwise, this was a terrible idea. glTexParameteriv() is expensive.
 - Remember kids, state change is costly on fixed-function hardware!





- Second idea: draw the tiles individually with draw_texture extension using a bunch of really small textures.
 - This actually works pretty well once some easy optimizations are made (RLE the tile map, etc).
 - But it's still a lot of calls to OpenGL. 400 calls to glDrawTexfOES() in the worst case, plus lots of superfluous glBindTexture calls.
 - Average case is good: less than 16 ms for the hardware to draw everything. But the actual submission of tiles to OpenGL is variable depending on the sparseness of the layer: 2 - 10 ms.
 - Worst case is bad: 9 13 ms to make GL calls and 19 23 ms to draw. Way unacceptable.



- Third idea: Make "meta tiles" out of vertex arrays, uv them to the atlas texture, and project them using orthographic projection.
 - Initial results were in line with the "basic vert quads" test in SpriteMethodTest: terrible.
 - Switching to VBOs sped things up a lot.
 - Lots of advantages to this approach: only one glBindTexture() call, very few total calls to other GL commands (only four meta tiles to draw per layer per frame).
 - Worst case situation (two layers with no empty tiles) is much faster than the draw_texture approach. CPU spends only 3 - 5 ms submitting to GL and drawing takes less than 16 ms.
 - Average case is the same! This makes it slightly slower than draw_texture in the average case (most maps are very sparse).



- Last ditch idea: pre-render the tile map and cut it up into meta tile textures, which can be drawn with VBO quads or draw_texture.
 - I haven't actually implemented this yet.
 - Level size will be restricted by total amount of VRAM if I use this method (unless I dynamically load textures).
 - High main memory cost too.
 - But, given all available information, drawing this way should be blazing fast.
 - I'm close enough to 60hz now that this probably won't be necessary.
- Future improvements to Android's GL interface (or the G1 GL driver) might render these optimizations unnecessary.



Requiem for Agenda #5

- Topics to cover today:
 - Why games? Why Android?
 - Game engine architecture.
 - Writing Java code that is fast.
 - Drawing stuff on the screen efficiently.
 - Tips, tricks, and pitfalls.



Performance Tips

- Touching the screen causes your app to be flooded with MotionEvents.
 - This will kill your framerate.
 - Sleep in the onTouchEvent callback to slow the flood. 16 ms is a good place to start.
- The mechanics of pausing and resuming are complicated when it comes to OpenGL, as the contents of VRAM are not always maintained. GLSurfaceView solves this for you.

- GLSurfaceView handles the state machine correctly.

- ATITC texture compression (supported on the G1) can be a big win if you are bus-bound.
- Android 1.0 1.5 failed to throw errors if you try to use VBOs with indirect buffers. Result: unpredictable crashes.
 - Solution is to just use direct buffers.



Game Design Tips

- Keep your application as small as possible. 2 ~ 3 mb is ideal.
- Now is the time for competent games. Customers are hungry for them and the platform is capable!
- As of right now there are already Android devices without a hardware keyboard (HTC Magic) or trackball (Samsung i7500). Don't rely on these input devices for game play (or support them all).
- The key to success Android Market is *quality*. Making a high-quality game is the way to be considered for the Featured Apps section too. Polish polish polish!
- You have always-on internet at your disposal. Use it!





