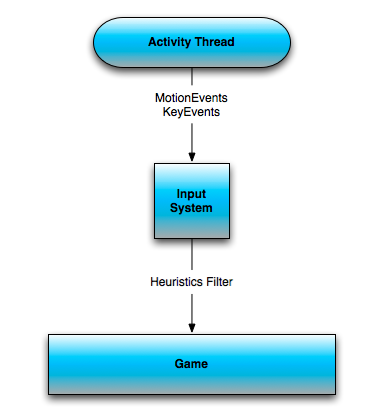
## MONDAY, MAY 3, 2010

### Control Configuration and Abstraction

The #1 thing that I've learned since shipping Replica Island is that users want configurable controls.  I mean, I might have guessed that some devices would have one sort of controller and not another, but I didn't anticipate the number of people who prefer a specific control configuration even when others are available.  Users with trackballs and directional pads asked for configurable keyboard settings, and when I added orientation sensor-based movement for devices without other controls (I'm looking at you, Xperia), many users who could already play the game chose to switch to tilt controls too.  I've made four updates so far and all of them have had to do with the input system; in the first three I added more and more configuration options, and in the most recent (v1.3) I rewrote the core input framework to improve non-standard control configurations.

When I started writing Replica Island, the only device available was the G1.  About half way through development I switched to an HTC Magic, and at the very end of development I switched to a Nexus One. The game was entirely designed around HTC's trackball-on-the-right design.  Fairly late in development, devices sporting directional pads (like the Motorola Cliq, and more importantly, the Droid) started to hit the market, so I added some support for d-pad controls.  I didn't really think anybody was going to use the keyboard to play, so I only added a few key-based controls to support the ODROID.

The input system started out like this:

[](http://replicaisland.net/img/blog/input_original.png)

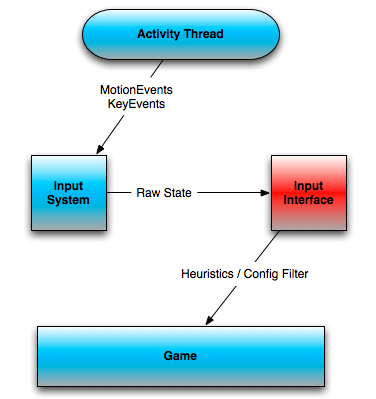
MotionEvents from touch and trackball motion, as well as KeyEvents, were passed to the InputSystem (via the GameThread, for reasons I'd rather not discuss), which recorded them in some internal structures.  The goal here was to abstract the Android events from the game interface.  The game wants to be able to say things like "is the jump button pressed," or "was the jump button just pressed since the last frame," or "how long has it been since the last time the button was pressed."  It's a query-based interface, rather than the message-based interface that the Android framework provides.  So the initial role of the InputSystem was to record Android events so that they could be queried in the future.

The trackball was tricky to get right.  I want to allow the player to flick the trackball in a direction and have the character get an impulse in that direction scaled by the magnitude of the flick.  But the Android motion events come in at a fixed frequency and fixed magnitude, so in order to build a vector describing recent motion, I needed to maintain some history between motion events.  My first implementation, which survived for the entire course of development, was to cache a history of 10 motion events and calculate the average direction of motion across all of them to find the flick direction and magnitude.  After a specific timeout had passed with no new events, the cache was cleared and averaging would begin again with the next event.

This worked ok as a way to calculate a motion vector, but it had problems.  The biggest issue was that there was no way for a user to move slowly; even if the user rolled the ball slowly (thus causing motion events to come less frequently), as long as he rolled fast enough to make the internal event timeout, the events would get averaged together and would come out looking the same as a fast flick.  So users who tried to move with precision or in small steps often found themselves rocketing across the level.

When I went to add d-pad support, I just treated the pad as a different source of motion events.  I treated each keydown as a roll of a specific magnitude in a specific direction, and fed that into the same cache system I used for motion events.  This worked, sort of: it allowed me to pipe the directional pad through the trackball interface (which connected directly to the game) pretty easily, but it didn't feel good.  The problem with this approach was that directional pad events don't need any averaging; in fact, you want exactly the most recent state to be represented, as the player can release a key at any time (the trackball, unlike other kinds of input, never goes "up", and thus required a history).  So directional pad support in Replica Island, in the first few versions, sucked.

Add in configurable control options and very quickly my simple G1-centric input system grew into a mess that didn't work very well.  So, for the most recent version, I rewrote the whole thing.  Now the structure looks like this:

[](http://replicaisland.net/img/blog/input_new.png)

The main change here is to separate input recording (necessary for querying) from game-specific filtering and control configuration switching.  The [InputSystem](http://code.google.com/p/replicaisland/source/browse/trunk/src/com/replica/replicaisland/InputSystem.java) is now generic; it just records input events from the keyboard, touch panel, orientation sensor, and trackball, and provides an interface for the current state (as defined by the most recently received events) to be queried.  A new system, [InputGameInterface](http://code.google.com/p/replicaisland/source/browse/trunk/src/com/replica/replicaisland/InputGameInterface.java), reads the hardware state from InputSystem, applies heuristics and filters, and presents fake buttons for the game to use.  This way the game can ask for the "directional pad" and get input from a trackball, orientation sensor, keyboard, directional pad, or whatever, already filtered and normalized.  I put all of the filtering code for the trackball into this class, and I can now pass directional pad input directly to the game without tying it to the trackball.

Speaking of the trackball, I changed my approach to filtering.  Now I accumulate trackball events that occur within a very short cutoff, and average them after a slightly longer cutoff.  Instead of turning the trackball input "off" after a fixed duration, I make it decay until it reaches zero.  This lets the user make small, precise movements, and still get a big motion from a large flick (as in the latter case, events occur in rapid succession and accumulate).  This method also gave me an obvious spot to stick a sensitivity factor for the trackball, which several users of devices with optical trackpads (HTC Desire, Samsung Moment, etc) had requested.

The new system probably needs a bit more tuning, but I played the game through with it and it feels pretty good.  The code is about 100x cleaner now, and InputSystem is something that others can easily reuse without any other dependencies.

POSTED BY [CHRIS](http://www.blogger.com/profile/16079744366139636224" \o "author profile)AT [1:55 PM](http://replicaisland.blogspot.de/2010/05/control-configuration-and-abstraction.html" \o "permanent link)

LABELS: [CODE DESIGN](http://replicaisland.blogspot.de/search/label/code%20design), [TARGETING ANDROID](http://replicaisland.blogspot.de/search/label/targeting%20Android)

#### 15 COMMENTS:

1. http://img2.blogblog.com/img/b36-rounded.png

**[Andrex](http://www.blogger.com/profile/14462912880548662710)**[May 3, 2010 at 3:08 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1272924523881" \l "c9024976703668890369)

Interesting. I would have thought that d-pad support would have been fairly trivial. But I guess that's just because I've been debugging purely in the emulator and not an actual device. I'd always been puzzled about how to do trackball movement "right", initially I would probably just done the opposite of what you did: map the trackball to d-pad specific code. :P  
  
Great to see you're continuing to support RI and the RI engine post-release. Can we expect to see Replica Island 2 at I/O? ;)

[Reply](javascript:;" \t "_self)

1. http://img2.blogblog.com/img/b36-rounded.png

**[Chris](http://www.blogger.com/profile/16079744366139636224)**[May 3, 2010 at 4:47 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1272930425176" \l "c1571568651242889535)

The issue is that I want the game to have a single entry point for movement controls. I want the game to see a single "directional pad", even if the device has no real dpad. Only, in the initial implementation, the game was looking for a single trackball, so on devices that didn't have one, I emulated it.   
  
Now the game uses virtual buttons exposed by InputGameInterface, and the source of those buttons is unknown to the game engine.  
  
No Replica 2 right away, I'm afraid. Still lots to do on Replica 1!

[Reply](javascript:;" \t "_self)

1. http://img1.blogblog.com/img/anon36.png

**[Mike Leahy](http://www.typhon4android.org/)**[May 3, 2010 at 5:44 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1272933878932" \l "c5397722102810481644)

Yep, a very time consuming task. This is a large focus (control config & abstraction) with the middleware, TyphonRT, I'm finally releasing shortly. It's taking a bit of time to resolve things as TyphonRT works across the desktop and Android. When factoring in a configuration system useful across the Android ecosystem including virtual control overlays and desktop configurations it's quite the task.  
  
Without getting overly wordy I take a similar approach in defining generic input actions controls 1D (buttons), 2D & 3D. This defines an "InputSet". The game engine components (entity system et al) of TyphonRT don't care about the physical input source.   
  
Lots more glue and cool stuff involved as everything is XML configurable w/ resource string substitution on Android for action control descriptions, there is the concept of an "InputCalibration" which is loaded from an XML file with defaults and device family/model name filtering. "InputCalibration" provides for device specific scaling. An example is the N1 trackball vs optical trackball. For 3D cases scaling to ensure that rotation of a game entity when using touch input swiping across the G1 screen is the same as the N1, etc.  
  
TyphonRT also provides a neutral event system and works with JInput, AWT, and Android. There are input filter components that can handle additional tasks such as tracking last time a button is pressed and even doing specific filtering for touch screens that are deficient (N1) in order to make virtual control overlays usable.   
  
Seeing as I support all sorts of game controllers with axis and dpad input via JInput you bet I'm hankering for proper bluetooth support in Android as I've already got a controller event system that just needs to be hooked up.  
  
Ok.. It's getting wordy / beyond wordy... heh.. Basically one can open a Pandora's box of sorts with a configurable config system w/ nice abstraction. Making it work across desktop and the entire Android ecosystem is no easy task.

[Reply](javascript:;" \t "_self)

1. http://img1.blogblog.com/img/anon36.png

**Anonymous**[May 8, 2010 at 3:59 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1273359557425" \l "c6399620576946363785)

Im having trouble understanding how to use opengl with android. I know theres a lot of documentation for opengl and a lot of documentation for android, but I can't put the two together. I read on the google code section that you are planning to work on a skeleton which will be like a framework for games. Is that still planned?

[Reply](javascript:;)

1. http://img2.blogblog.com/img/b36-rounded.png

**[Chris](http://www.blogger.com/profile/16079744366139636224)**[May 8, 2010 at 5:09 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1273363796827" \l "c2914094243762401276)

Anonymous,  
  
You can get OpenGL ES up and running on Android in just a few lines of code. See the API demos that come with the SDK. You don't need a whole game engine.

[Reply](javascript:;" \t "_self)

1. http://img2.blogblog.com/img/b36-rounded.png

**[Gilead](http://www.blogger.com/profile/13407008816951592471)**[May 13, 2010 at 1:27 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1273782422912" \l "c7871250031453741382)

Mike, you can connect BT controllers like JS1 to Android right now, see my story over @ StackOverflow :) http://stackoverflow.com/questions/2660968/

[Reply](javascript:;" \t "_self)

1. http://img2.blogblog.com/img/b36-rounded.png

**[Chuck](http://www.blogger.com/profile/14789510707956337658)**[May 19, 2010 at 10:51 AM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1274291518412" \l "c7541384026803866964)

Chris,  
I am fairly new to Java and wanted to begin learning the language to develop applications on Android. After watching your video on the Android Dev website I am intrigued with game development. In your opinion, what should someone like myself learn in order to get into game development for Android? Is Java enough? Do I need to learn about Physics and other mathematics topics? Your opinion is appreciated. Thanks!

[Reply](javascript:;)

1. http://img2.blogblog.com/img/b36-rounded.png

**[Chris](http://www.blogger.com/profile/16079744366139636224)**[May 19, 2010 at 11:03 AM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1274292219502" \l "c4487083189187744838)

@Chuck  
  
If you're just starting out with Java and game development, I think it's a fine learning tool. Just target something simple: a tetris clone, or sudoku, or whatever; something that you can accomplish while learning the language and platform. Once you have a more solid foundation to build upon you can increase the complexity of your projects.  
  
I think game design and programming language are pretty much orthogonal issues, so my advice is get comfortable with whatever language first (Java, sure, whatever) and then figure out what kind of games you want to make with it. When you hit a wall related to the language (which, if you're just starting, won't be for a long while), consider another approach at that time.

[Reply](javascript:;" \t "_self)

1. http://img2.blogblog.com/img/b36-rounded.png

**[Hasoan](http://www.blogger.com/profile/16239826672262783121)**[May 21, 2010 at 3:56 AM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1274439408059" \l "c4435626508460934593)

Just wanted to let you know that my samsung moment optical pad does not work at all wit the 1.3 update, which was kind of funny since the update specifically mentions fixing it... anyway, I'm running android 2.1 if that might make a difference. Sorry if this is the wrong place for bugs.

[Reply](javascript:;)

1. http://img2.blogblog.com/img/b36-rounded.png

**[hearn](http://www.blogger.com/profile/02123376927554957565)**[May 24, 2010 at 2:42 AM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1274694147901" \l "c1236881211484799083)

One issue I had with the game on the N1 is that with the touch-sensitive buttons it's very easy to get too aggressive with the trackball and accidentally press the home or back buttons. I don't think there's a good solution to this, chalk it up to fragmentation ;)  
  
BTW what's up with all the workarounds and general noise in the app description and changelog for specific devices. If Cliq is "broken" and needs a "safe mode" can't you detect that at startup and handle it automatically?

[Reply](javascript:;" \t "_self)

1. http://img1.blogblog.com/img/anon36.png

**Christian**[May 25, 2010 at 8:54 AM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1274802883601" \l "c7425296448312132927)

Hello Chris, nice game. I’ve being playing it a little but mostly studying the source for a while. I’m trying to do a game as well. I’ve in mind a different concept but I’m planning it to have the same kind of 2d graphics. I’m trying to guess some stuff from opengl out of replica. I’m new at this though, and it’s a hard task to understand what all the code does. I’m sorry if this is not the right place to ask for this but could u post the stripped replica island code used as the FPS demo at Google's IO? What I’m basically trying to do right now is find out how you got opengl to work on the glsurfaceview ( .. android:id="@+id/glsurfaceview" ..), so that it can have other layers, or even be in a linearlayout to only use part of the screen). Could you please help me in this. I’ve being researching for a week now and every opengl example I’ve found uses the complete screen and not the XML. I’ve also found many similar questions regarding this on the web with no answers.

[Reply](javascript:;" \t "_self)

1. http://img1.blogblog.com/img/anon36.png

**[Thomas Sheppard](http://androided.wordpress.com/)**[May 25, 2010 at 6:29 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1274837355629" \l "c6064883808576928157)

Hey Chris, Sorry to pull the comments off topic but I had a question and didn't know where to ask it. I wanted to work on a level builder add on and noticed the level layouts seem to be defined in bin files in the raw resources directory. Is there an easy way of reading in these files? I wanted to gain an understanding of them to then be able to build them.  
  
Thanks for the awesome code and posts.  
  
Tom

[Reply](javascript:;)

1. http://img2.blogblog.com/img/b36-rounded.png

**[Chris](http://www.blogger.com/profile/16079744366139636224)**[May 26, 2010 at 11:09 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1274940546435" \l "c7813633265434832334)

@Hasoan  
  
Ok, thanks for the report. Bug reports should probably go here:  
  
http://code.google.com/p/replicaisland/issues/list  
  
I'll double-check the Moment. The last time I tested on that device was before their 2.1 upgrade.  
  
@hearn  
  
I actually have code in the game to ignore misclicks on the back and menu keys (I ignore clicks to those buttons within 400 ms of a trackball roll), but there's nothing I can do about the home key. Sigh.  
  
The Cliq is not the only broken device--there are a number of devices that use the same broken Qualcomm graphics driver. It's just that the Cliq is (apparently, based on the volume of complaints) the device that sold the most in this category. I actually do detect the Cliq at runtime and turn safe mode on, but it seems that there's maybe another model (the XT?) that I've missed in that test.  
  
@ Christian  
  
You might want to look at SpriteMethodTest--it's much simpler but functionally very similar to Replica Island:  
  
http://code.google.com/p/apps-for-android/source/browse/#svn/trunk/SpriteMethodTest  
  
The short answer to your question is: just make a layout with a GLSurfaceView in it in XML, give it an id the way you do any Layout element, then find it using that id in your Activity's onCreate() method and set a renderer. Pretty simple.  
  
@Thomas Sheppard   
  
There's a thread about level building going on over at the development community list:  
  
http://groups.google.com/group/replica-island-coding-community

[Reply](javascript:;" \t "_self)

1. http://img2.blogblog.com/img/b36-rounded.png

[**Oren**](http://www.blogger.com/profile/02685726922532859012)[July 6, 2010 at 10:24 AM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1278437081589#c1734682573779356311)

You wrote that the input interface code can be reused. I'm new to Android, and I've been wondering about this aspect. There is no jar to include - from what I saw, only way for code reuse is intents. Am I missing something? should I copy the InputInterface code to my project?

[Reply](javascript:;)

1. http://img1.blogblog.com/img/anon36.png

**Anonymous**[December 10, 2010 at 3:44 PM](http://replicaisland.blogspot.com/2010/05/control-configuration-and-abstraction.html?showComment=1292024652027#c6467289971144187799)

I'd like to see it possible to map the volume, search, back, and menu buttons, since they aren't used in the game. Because these buttons do other actions as well as what they are assigned to, they are impossible to use in the game. Also, I think the pause button should be reassignable.  
  
This would make the game more playable on devices that lack both a keyboard and directional pad, such as the upcoming Nexus S. You could use the volume buttons for left/right control, the back button for jump, and the menu button for attack. I think that would work quite well.

[Reply](javascript:;)