

# Developing 24/7 Flash apps

**Matt Pollitt - [mattpollitt.co.uk](http://mattpollitt.co.uk)**

**Juan Delgado - [zarate.tv](http://zarate.tv)**

**[ustwo.co.uk](http://ustwo.co.uk)**



us-hole™



**The us-hole™ is ustwo™'s window to Shoreditch.  
An AIR 2 application running on a MacMini  
that displays any crazy content we might come up with.**





STING

SECOM

SECOM

12

STING

STING

SECOM

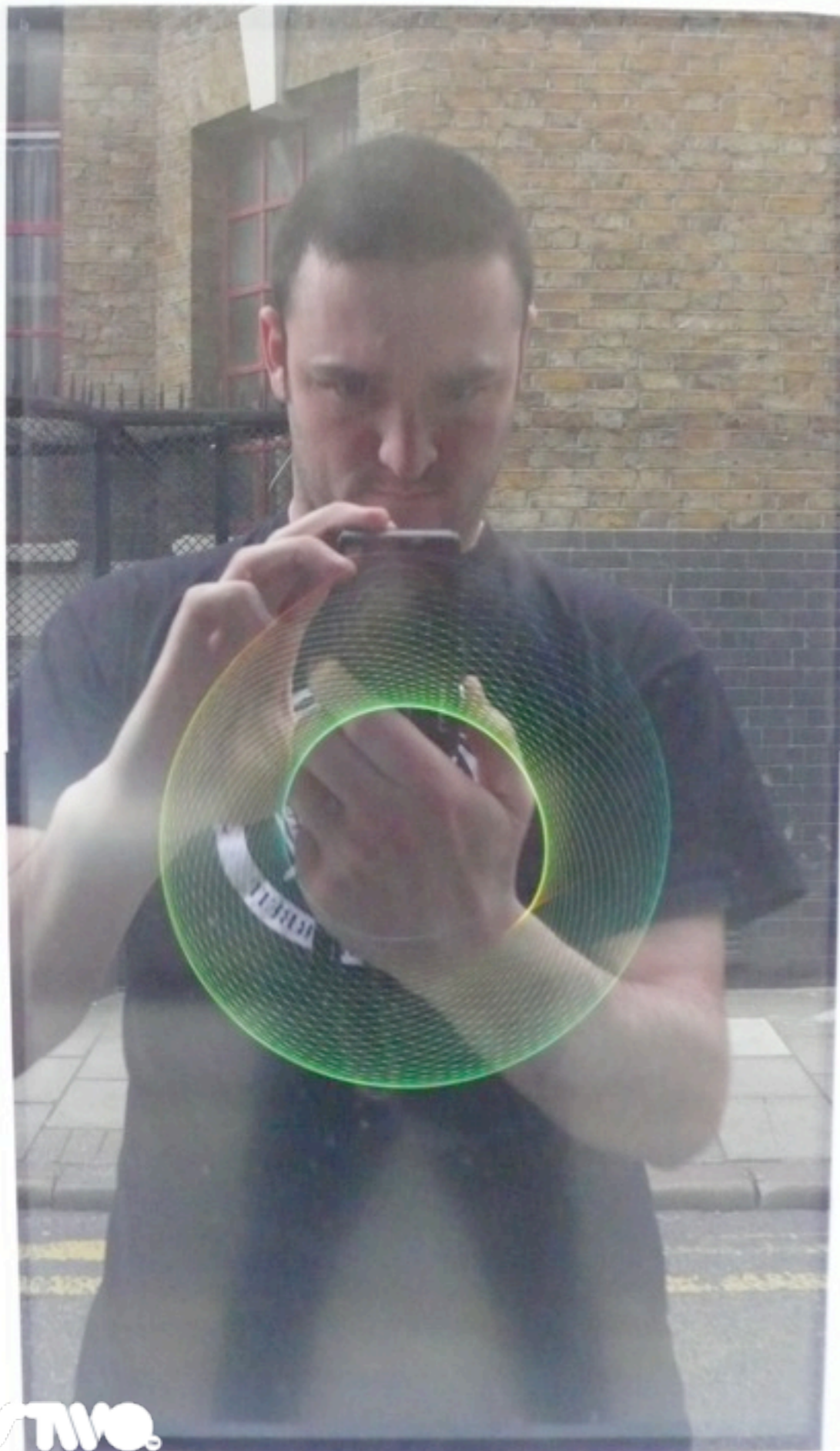
SECOM



STING



STIVO



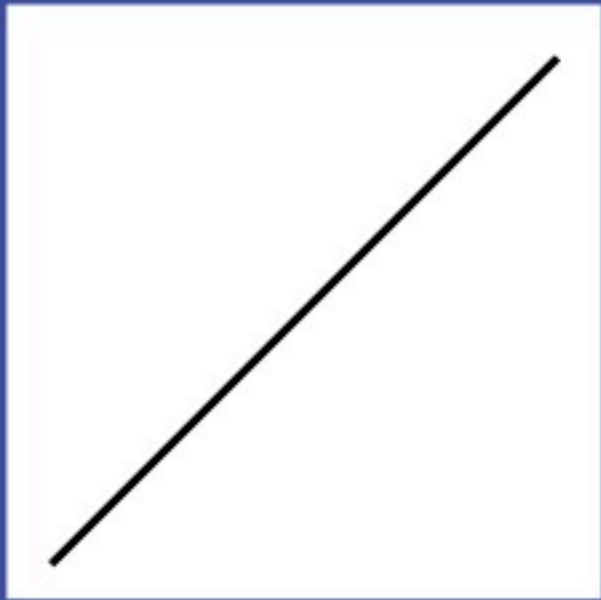
**24/7**

**B/S**

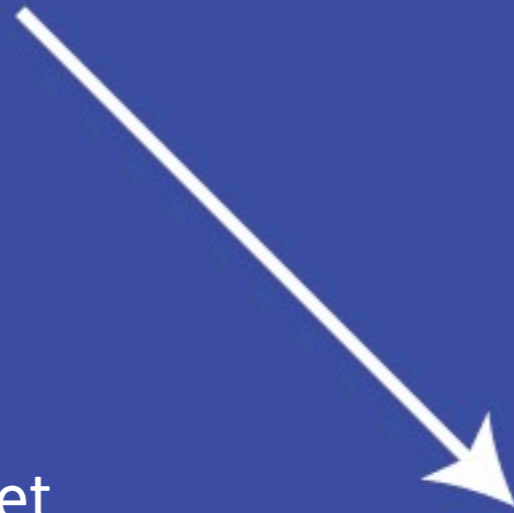
23.9/7

(that's more likely)

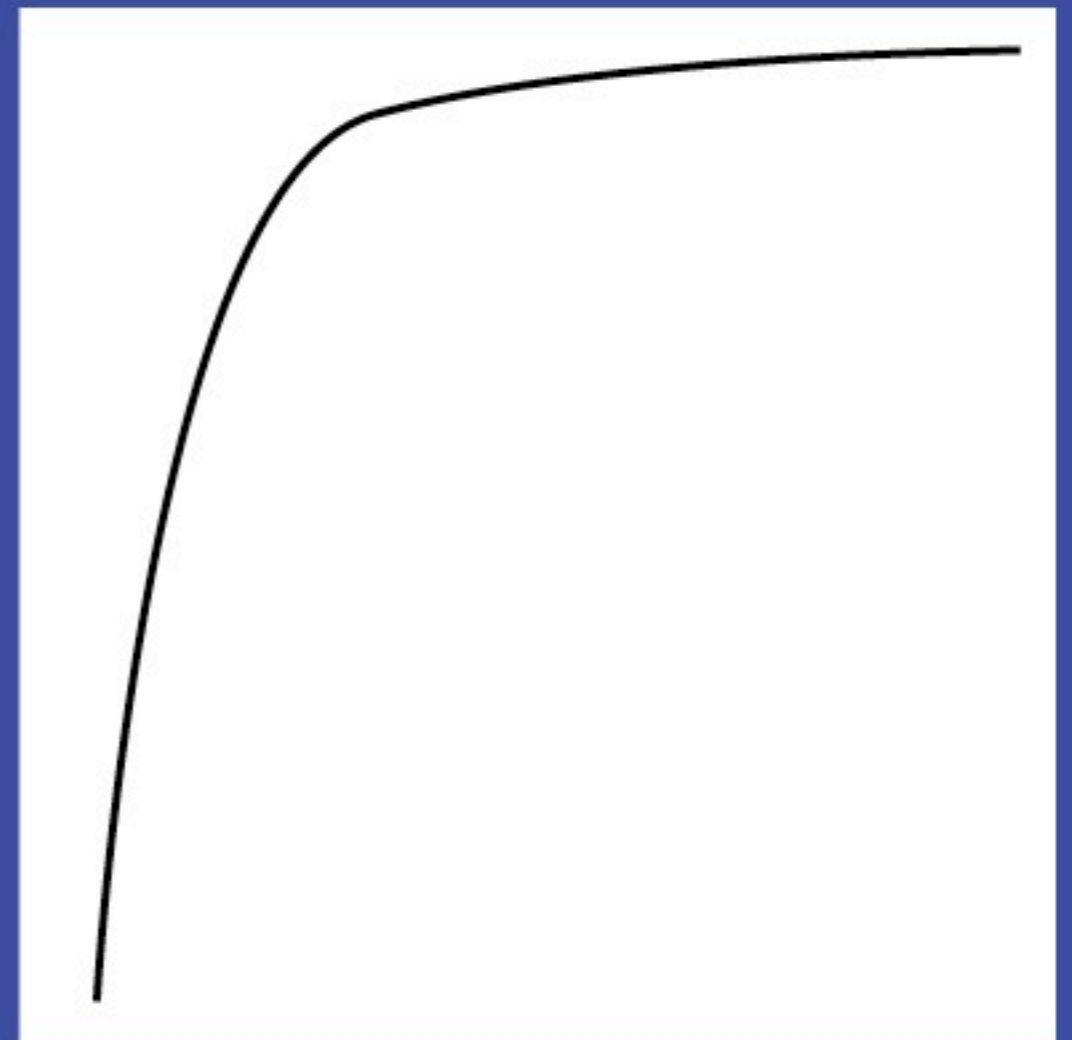
uptime



effort



uptime



effort

It's near impossible to get 100% uptime and also remember that going over 90% requires much more effort than getting to 90%.

Set a target uptime based on your time and budget.

# FAIL

Be ready to FAIL, shit happens all the time. There will be errors you just cannot foresee.

Use the new 10.1 API for uncaught errors:

```
stage.loaderInfo.uncaughtErrorEvents.addEventListener(UncaughtErrorEvent.UNCAUGHT_ERROR, onError);
```

# RELEASE

Release as much resources as you can because when running 24/7 every detail counts. Explicitly remove all listeners, use `unloadAndStop`, etc.

You can't be too strict with yourself.

# MONITOR

Monitor your application. We decided to use a watchdog application in haXe + Neko that checks every minute whether the us-hole™ is running or not. If it is not running, it will be launched.

# UPDATE

Put in place a good updating mechanism. We decided *not* to use AIR's update framework because:

- \* We wanted to delegate to an app that wasn't AIR based in case AIR runtime crashed and
- \* We didn't have a public URL for the updates.

# us-hole™ updates

geeks



dev

manual  
merge

master

automatic  
update

Updates to the us-hole™ are automatic whenever there's an update to the master branch in our GIT repository.

us-hole™

# LOG

Log as much events as possible, there's not enough information you can log. When there's the need of doing crash investigation, the more information you have, the better.

Just be careful of not coming up with huge log files, the us-hole™ has a 2MB limit on log files, then creates a new one.

# WARN

Set up automatic warnings when errors occur. You can set up a private Twitter account that developers subscribe to and automatically post errors. You can also use Growl or send emails (using AS3 SMTP's libraries).

Please note that most of these systems require an internet connection available (let alone that Twitter might go down any minute) so do not exclusively rely in one of them, ALWAYS log locally to a file.

# REMOTE

Many 24/7 applications are for kiosks and are physically located far away from the developer so try to install a VNC server (for screen sharing) or at least SSH access (for diagnosis).

# CONTROL

Keep the environment your application is running under your control as much as you can. Disable:

- \* System updates.
- \* Java updates.
- \* AIR runtime updates.
- \* Error reporting.
- \* Etc.

Many of those updates require user interaction (when 24/7 apps run mostly unattended) and might not be backwards compatible with your application.

There's a slighter longer version of this talk online in [Running Flash apps 24/7](#).

Read [Build tools with haXe](#) and [XAPI](#) if you are Flash dev interested in creating little native applications to interact with the system (think file management, running GIT, etc.).

Contact us for any doubts or comments:

Matt Pollitt: [matt@ustwo.co.uk](mailto:matt@ustwo.co.uk)  
Juan Delgado: [juan@ustwo.co.uk](mailto:juan@ustwo.co.uk)

Thanks :)

