



Test Your Robotlegs Project

高見龍





9th Flash/AS Study Group

高見龍

a.k.a Eddie or Aquarianboy

- Live and work in Taipei, Taiwan.
- Serving in my own little tiny company.
- Flash / AS3 / Ruby / Rails / Python programming for living.
- A little bit Objective-C for personal interests.
- Technical Education and Consultant.
- PTT Flash BM (since 2007/4).
- Adobe Certificated Flash Developer (Since 2006/7).
- Linux Professional Institute Certification (Since 2005/3).





Previous on Robotlegs...

Where we start..



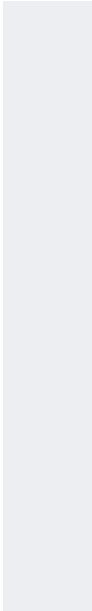


Every Robotlegs project begins
with a Context.

Use `startup()` to provide your injection rules, map your mediators to views and commands to events.

Dependency





A dependency is just a requirement
to use another object

Injection





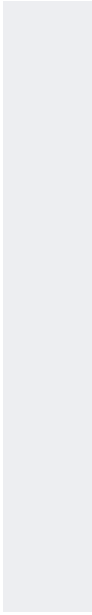
Injection is cool, but it isn't magic!



Don't inject base types.



The injector must be declared as public.



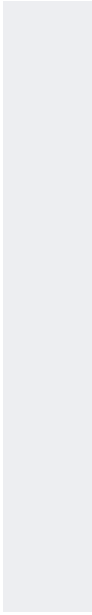
or they will be silently ignored.

Mediator





Robotlegs mediators are bridges,
not view controllers.



Don't imagine it as part of your
view layer.



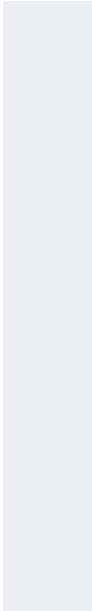
Adding view logic to mediators gets messy very quickly.



Mediator just do translation jobs!

Command



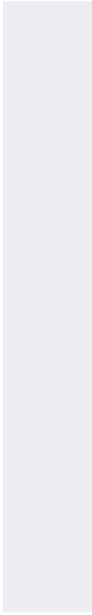


A Command is a concise single-purpose controller object.



A command is just a class which has a public `execute()` method.

The advantage to extending the Command class is that many handy injections are already made for you.



Commands should do their thing
and then die.

Model and Service

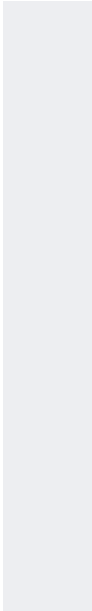


A service doesn't store data. A model doesn't communicate with the outside world to retrieve data.

If they don't dispatch events to the shared event dispatcher , you don't need to extend Actor.

Architecture

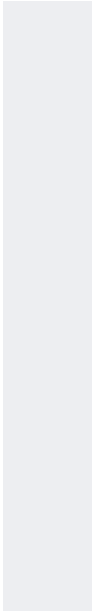




you should always let your model
design drive your service design,
and never the other way around.

if only the view classes care about this logic, it belongs in your view layer.

if other parts of the application might need to be checked or informed, it's controller code.



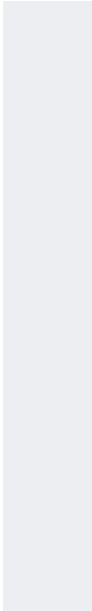
if you've already forgotten
these things I just mentioned..

you can visit:

www.eddie.com.tw/slides

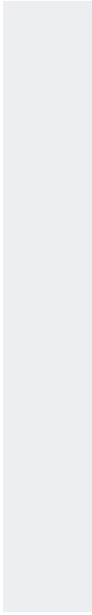
TONIGHT!



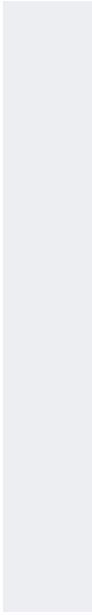


Tonight, we're going to
talk about Testing.

Before we start..



In your company, who does these tests?



and, how do they do tests?



How can codes are testable?

Test-first development is hard —
it's hard because it forward-shifts
your confusion.



You might heard about the “DDD”



“Deadline-Driven Development”



or “SCDD”



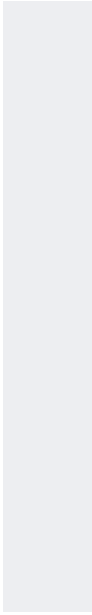
“Stupid-Client-Driven Development”

TDD

Twitter-Driven Development



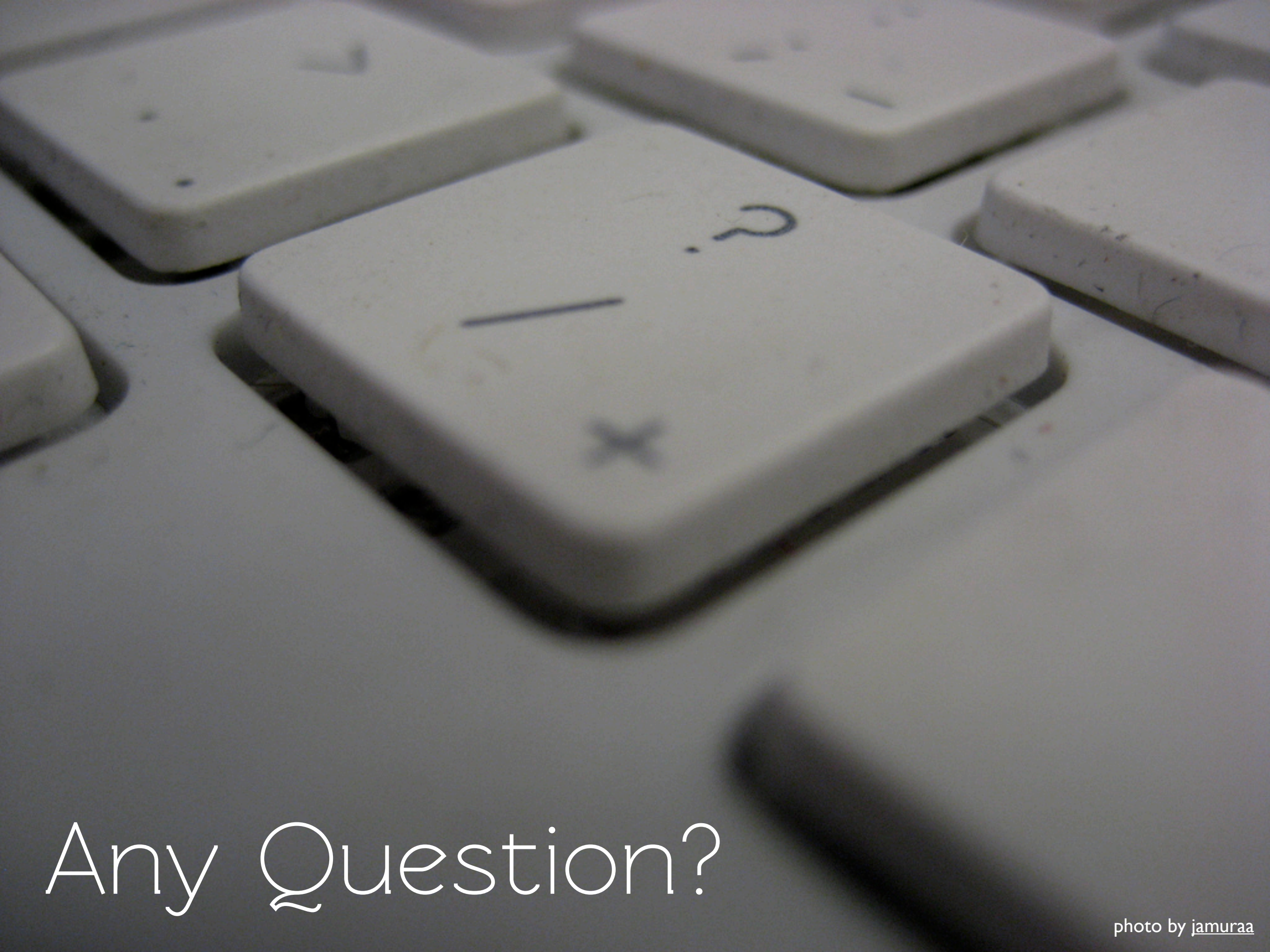
Test-Driven Development



Test != Debug

Live Coding Time!

and it's REALLY Live Coding!



Any Question?

Contacts

高見龍

- Website <http://www.eddie.com.tw>
- Blog <http://blog.eddie.com.tw>
- Plurk <http://www.plurk.com/aquarianboy>
- Facebook <http://www.facebook.com/eddiekao>
- Google Plus <http://www.eddie.com.tw/+>
- Twitter <https://twitter.com/#!/eddiekao>
- Email eddie@digik.com.tw
- Mobile +886-928-617-687