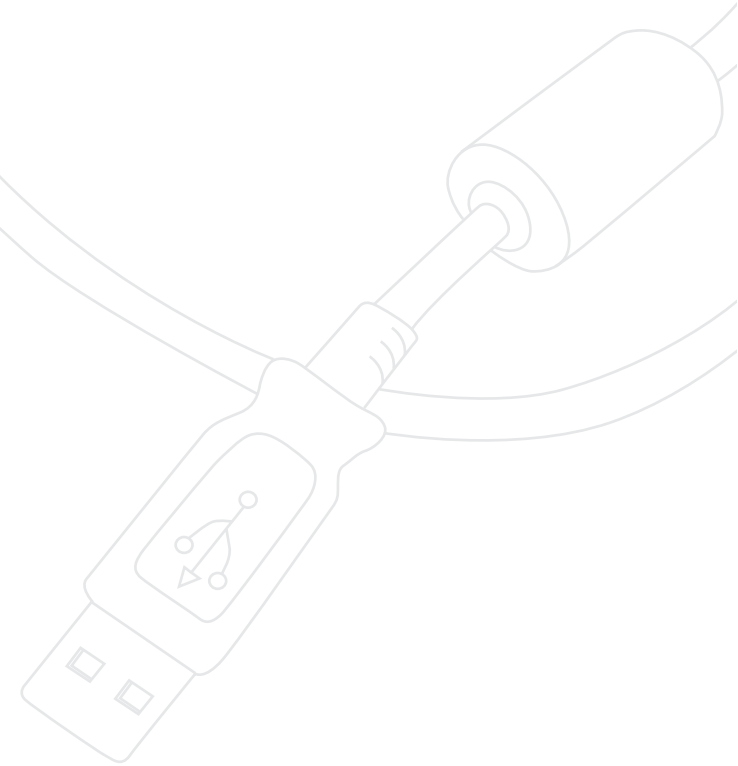




**Laurent Deketelaere**

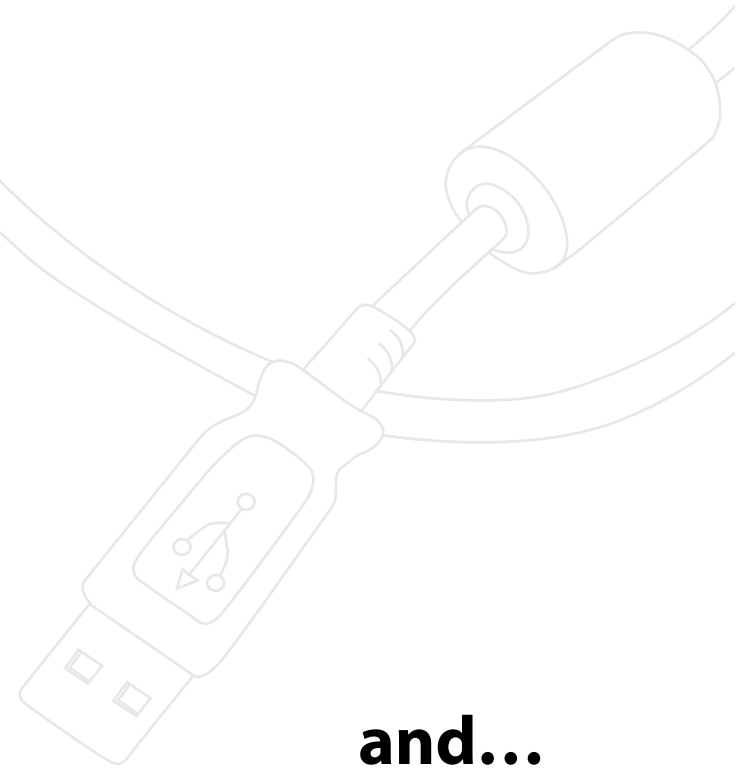
# Bonjour



# PRÉSENTATION

## Un environnement

- **Eclipse**  
environnement de développement.
- **FDT**  
plugin d'édition Actionscript 2/3.
- **MTASC**  
compilateur open source
- **Flash Inspector**  
Logguer externe
- extra : SWFMILL, SVN

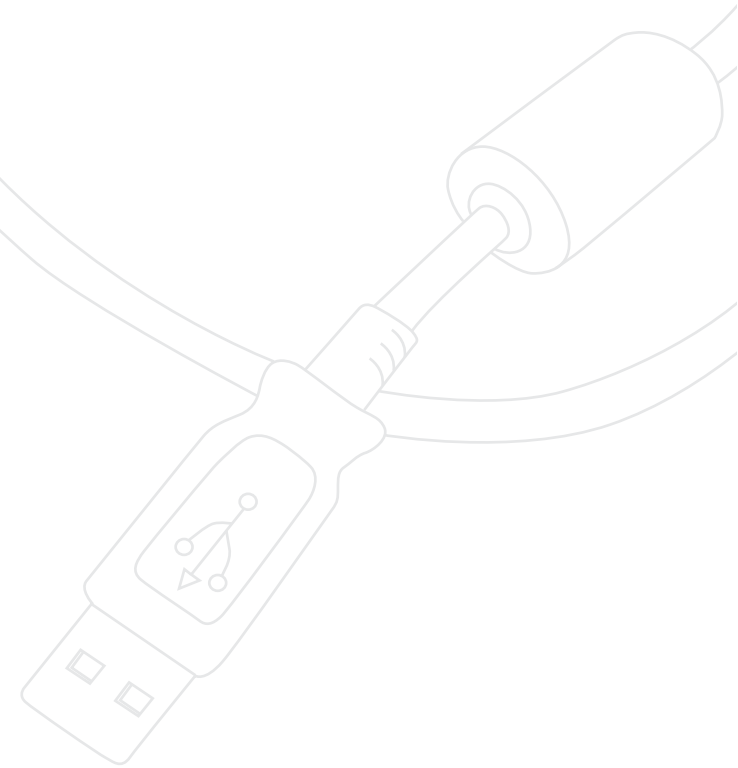


**and...**

# PRÉSENTATION

## **pixLib**

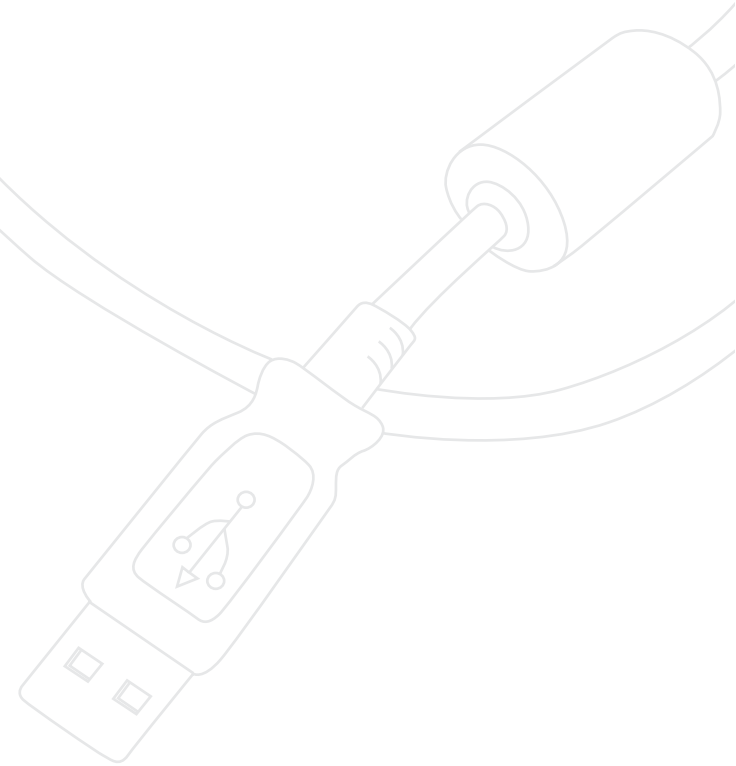
- une librairie de bas niveau.
- Comble les manques de l'actionscrip 2.
- De l'actionscrip 3 avant l'heure.



# PRÉSENTATION

## But de la semaine

Découvrir le framework pixLib, en deployant un projet.



## Logger == Afficher les messages d'erreurs

```
8 class pixlib.Log
9 {
10 public function Log()
11 {
12     // -- débuggage
13     Logger.LOG("Quelle est la couleur du cheval blanc de Napoléon?" + "blanc",
14     // -- information
15     Logger.LOG("Version : " + System.capabilities.version, LogLevel.INFO);
16     // -- attention
17     Logger.LOG("ATTENTION!", LogLevel.WARN);
18     // -- erreur
19     Logger.LOG("ERREUR!!", LogLevel.ERROR);
20     // -- erreur fatale
21     Logger.LOG("C'est mort :p", LogLevel.FATAL);
22 }
23 }
```



<http://blog.geturl.net/post/2006/01/02/12-pixlib-comment-loguer>

## Logger + Channel == faire le tri dans les infos

```
11 class pixlib.Debug
12 {
13     public static var isOn : Boolean = true;
14     public static var channel : LogChannel = new LogChannel( ClassUtils.getFullyQualifiedClassName(
15
16     private function Debug()
17     {
18     }
19
20     public static function DEBUG( o ) : Void
21     {
22         if (Debug.isOn) logger.LOG( o, LogLevel.DEBUG, Debug.channel );
23     }
24
25     public static function INFO( o ) : Void
26     {
27         if (Debug.isOn) logger.LOG( o, LogLevel.INFO, Debug.channel );
28     }
29
30     public static function WARN( o ) : Void
31     {
32         if (Debug.isOn) logger.LOG( o, LogLevel.WARN, Debug.channel );
33     }
34
35     public static function ERROR( o ) : Void
36     {
37         if (Debug.isOn) logger.LOG( o, LogLevel.ERROR, Debug.channel );
38     }
39
40     public static function FATAL( o ) : Void
41     {
42         if (Debug.isOn) logger.LOG( o, LogLevel.FATAL, Debug.channel );
43     }
44 }
```

<http://blog.geturl.net/post/2006/01/07/15-pixlib-import-combourellogchannel>

## LibStack == le multi-chargement

```
63 private function _loadAssets() : Void
64 {
65     var lib : LibStack = new LibStack();
66
67     lib.enqueue(new ConfigLoader(), "xml de configuration", "xml/config.xml");
68     lib.enqueue(new GraphicLib(_view, 1, false), "image", "picts/google.jpg");
69     lib.enqueue(new GraphicLib(_view, 2, false), "swf", "swf/carre.swf");
70     lib.enqueue(new GraphicLib(_view, 3, false), "swf2", "swf/carre2.swf");
71     lib.enqueue(new GraphicLib(_view, 4, false), "swf3", "swf/carre3.swf");
72
73     lib.addEventListener(LibStack.onLoadProgressEVENT, this, _onLoadProgress);
74     lib.addEventListener(LibStack.onLoadCompleteEVENT, this, _onLoaded);
75
76     lib.execute();
77
78 }
```

<http://blog.geturl.net/post/2006/05/08/47-pixlib-libstack-ou-le-multi-chargement>



## Métronome et temporisation

```
import com.bourre.transitions.FPSBeacon;
import com.bourre.transitions.IFrameListener;

class net.geturl.project.TimeManagerUI
    implements IFrameListener
{
    public function TimeManagerUI ( Void )
    {
        // --
    }

    public function start( Void ) : Void
    {
        FPSBeacon.getInstance().addFrameListener( this );
    }

    public function stop( Void ) : Void
    {
        FPSBeacon.getInstance().removeFrameListener( this );
    }

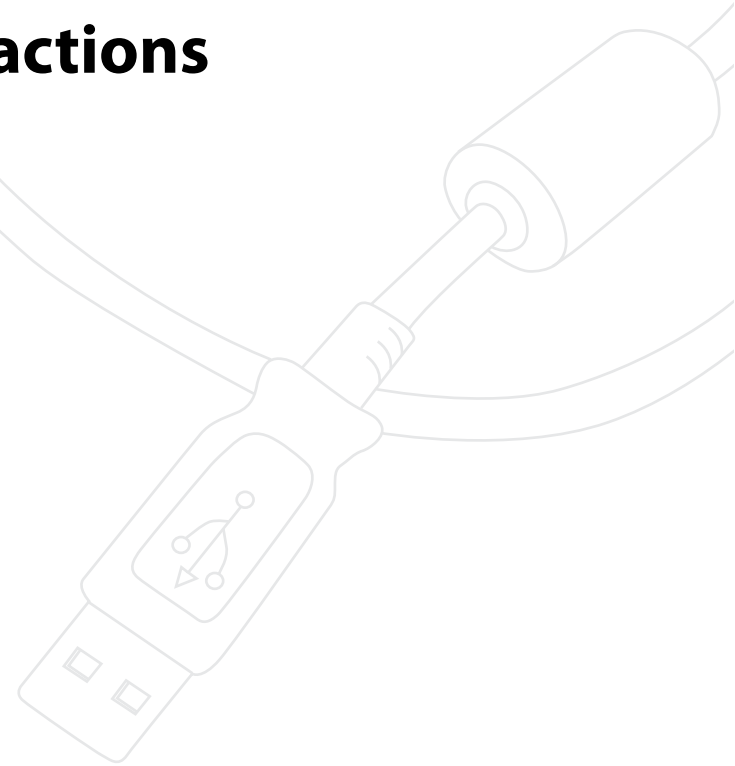
    public function onEnterFrame() : Void
    {
        // actions
    }
}
```

<http://blog.geturl.net/post/2006/11/13/65-pixlib-metronome-et-temporisation>

# ORIENTÉE OBJET

**Application == Ecosystème**

**Ecosystème == Individus + Interactions**



# ORIENTÉE OBJET

## **Encapsulation == Intimité**

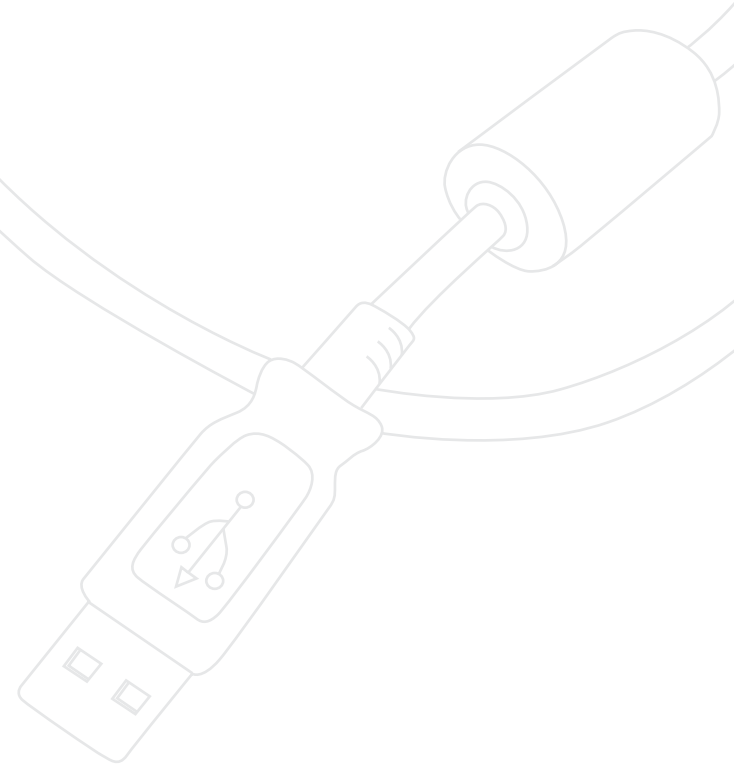
Je montrer QUE ce dont vous avez besoin.

## **Héritage == extends == Famille**

Je ressemble à mes parents.

## **Polymorphisme == interface == Contrat**

Je dois respecter mes contracts.



# ORIENTÉE OBJET

## Encapsulation == Intimité

Je montrer QUE ce dont vous avez besoin.

```
7 class Encapsulation
8 {
9     private var _nPercent : Number;
10
11     public function getPercent() : Number
12     {
13         return _nPercent;
14     }
15
16     public function setPercent(nPercent : Number) : Void
17     {
18         _nPercent = Math.min(Math.max(0, nPercent), 100);
19     }
20 }
21
```

# ORIENTÉE OBJECT

## Héritage == extends == Famille

Comme mes parents, mais en mieux.

```
5 class poe.Heritage
6   extends Date
7 {
8   public function Heritage(year : Number, month : Number, date : N
9   {
10    super(year, month, date, hour, min, sec, ms);
11  }
12
13  public function timediff(d : Date) : Number
14  {
15    return Math.abs(this.getTime() - d.getTime());
```

# ORIENTÉE OBJET

## Polymorphisme == interface == Contrat

Je dois respecter mes contracts.

```
5 interface poo.polymorphisme.IArea
6 {
7     public function getSurface() : Number;
8 }
```

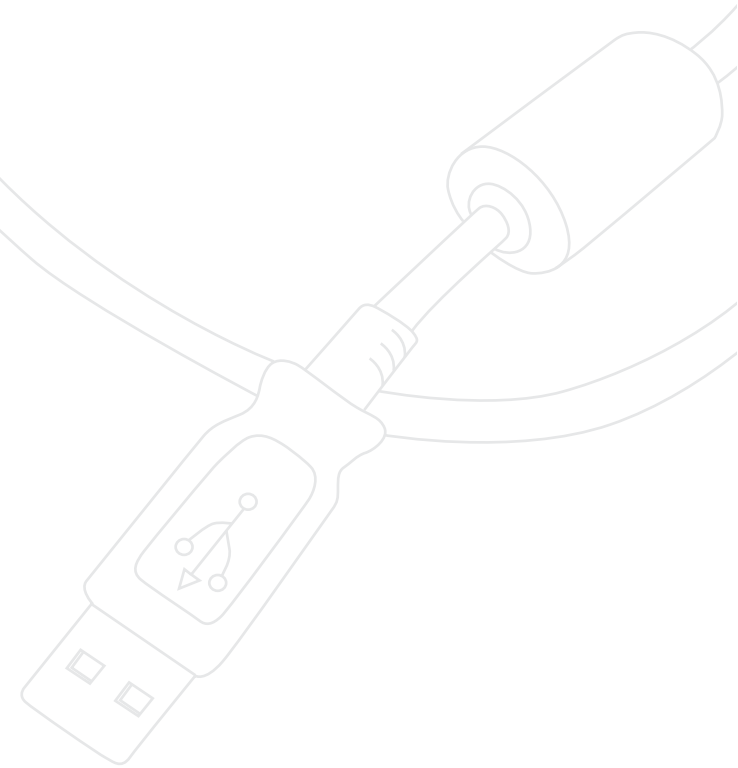
```
6 class poo.polymorphisme.Cercle
7     implements IArea
8 {
9     private var _nRayon : Number;
10
11     public function Cercle( nRayon : Number )
12     {
13         _nRayon = nRayon;
14     }
15
16     public function getSurface() : Number
17     {
18         return Math.PI * _nRayon * _nRayon;
19     }
20 }
```

```
6 class poo.polymorphisme.Carre
7     implements IArea
8 {
9     private var _nCote : Number;
10
11     public function Carre( nCote : Number )
12     {
13         _nCote = nCote;
14     }
15
16     public function getSurface() : Number
17     {
18         return _nCote * _nCote;
19     }
20 }
```

# DESIGN PATTERNS

**Design Patterns** == recettes de cuisine

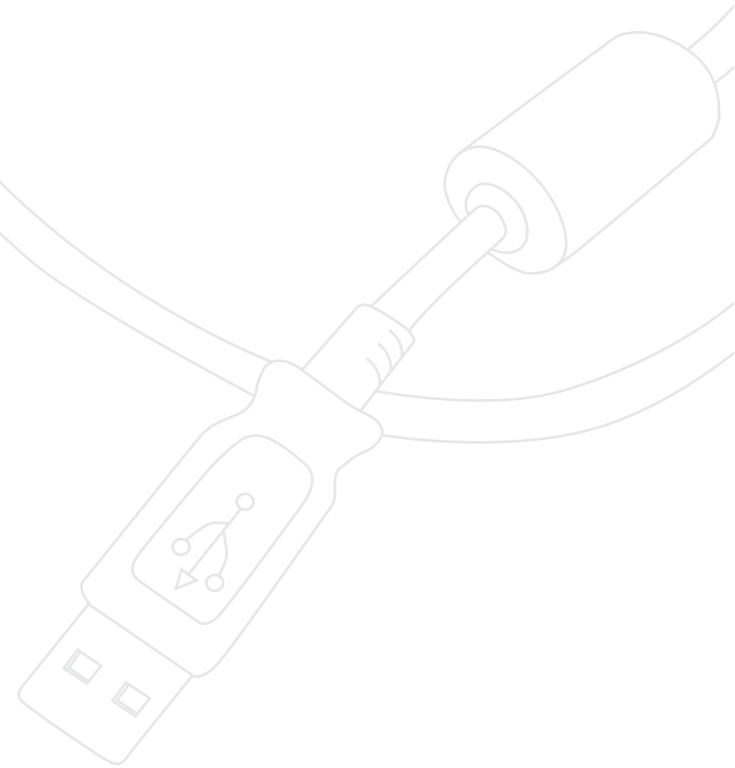
- Une recette c'est fait pour être adapté.
- méthode éprouvé.



# DESIGN PATTERNS

## Singleton == Une seule instance

```
6 class design_patterns.Singleton
7 {
8     private static var _oI : Singleton;
9
10    /**
11     * @return singleton instance of Singleton
12     */
13    public static function getInstance() : Singleton
14    {
15        if (!_oI) _oI = new Singleton();
16        return _oI;
17    }
18
19    private function Singleton()
20    {
21    }
22 }
```

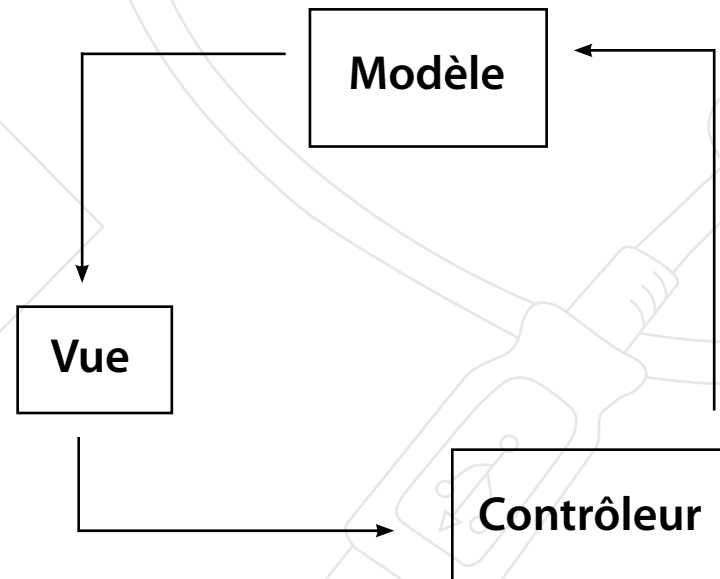




# DESIGN PATTERNS

## **MVC == Modèle - Vue - Contrôleur**

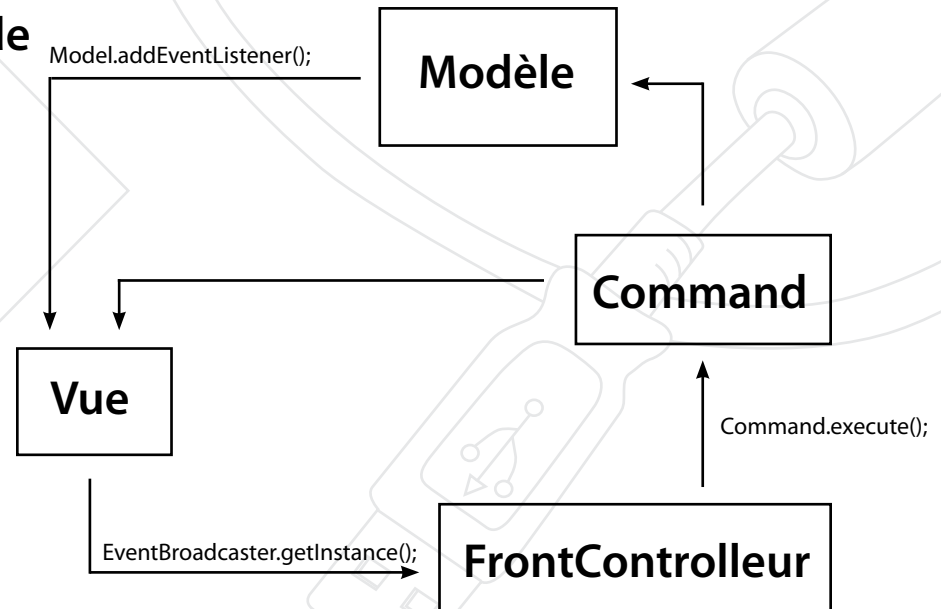
- **Modèle == Savoir.**
- **Vue == Paraître.**
- **Contrôleur == agir.**



# DESIGN PATTERNS

**MVC + FT == Une communication globale.**

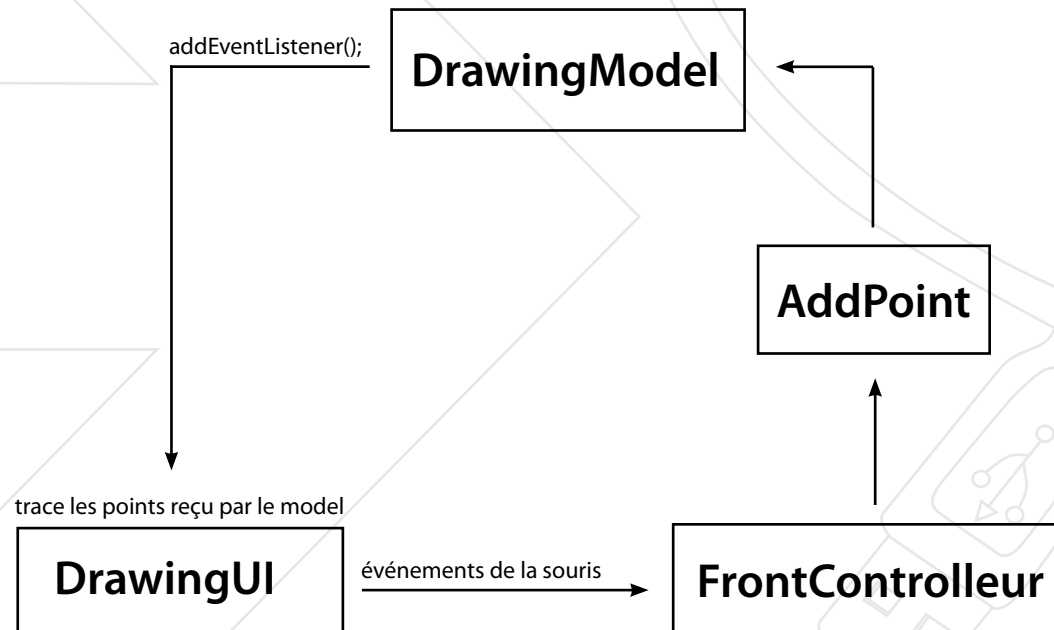
- Le FrontController == bureau de post
- Un événement == Une commande
- Un événement == Une adresse



<http://blog.geturl.net/post/2006/10/27/63-pixlib-vues-modeles-controleur>

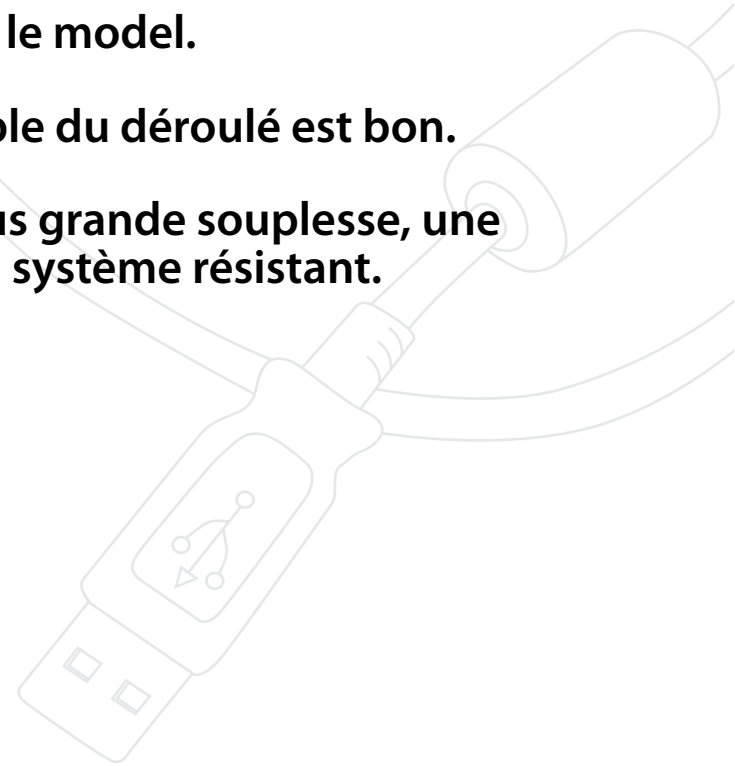
# DRAWING

## MVC + FT implémentation dans une application de dessin.



## Avantages

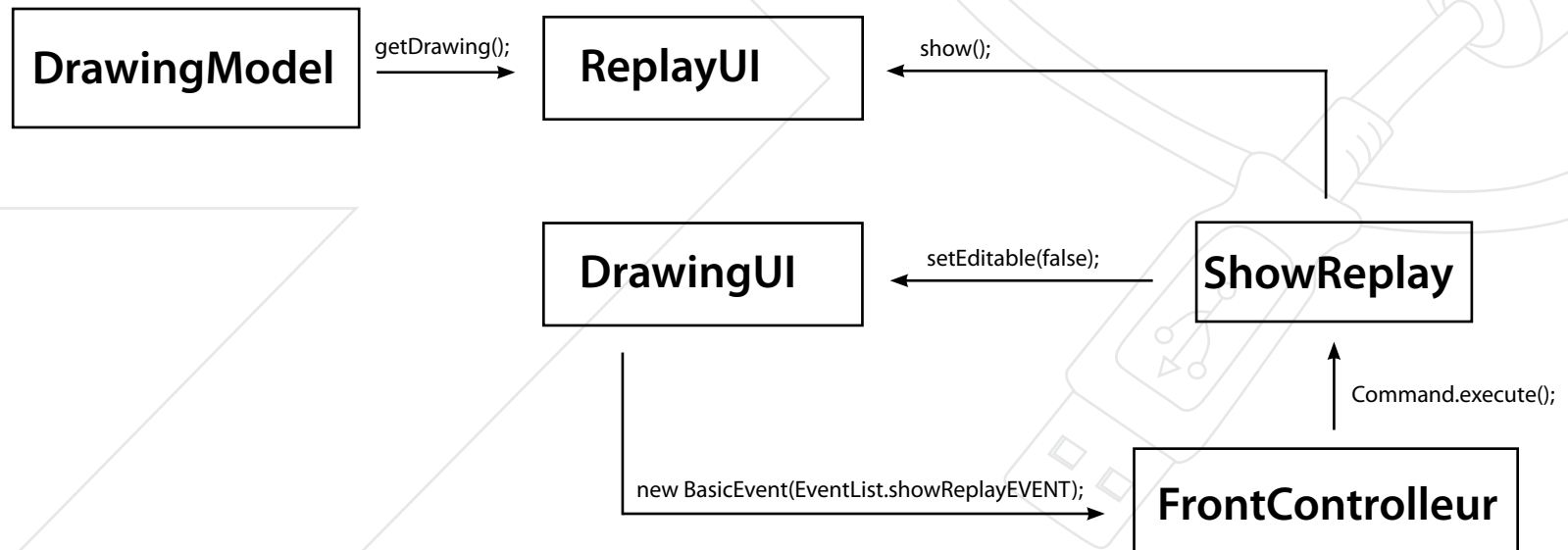
- On trace uniquement les points enregistré par le model.
- Si les points s'affiche cela signifie que l'ensemble du déroulé est bon.
- Le fait de séparer ainsi le code apporte une plus grande souplesse, une facilité pour le débogage et assure d'avoir un système résistant.



# DRAWING

## Ajouter un replay - ReplayUI

- On récupère les points sauvé dans le modèle.
- On retrace chaque points avec un ecart temps de 10milisec.



# LINKS

- ECLIPSE  
<http://www.eclipse.org/>
- FDT  
<http://fdt.powerflasher.com/>
- MTASC  
<http://www.mtasc.org/>
- SWFMILL  
<http://swfmill.org/>
- SVN  
<http://tortoisesvn.tigris.org/>
- FLASH INSPECTOR  
<http://www.luminicbox.com/blog/default.aspx?page=post&id=2>
- PIXLIB  
<http://www.pixlib.org/>  
[http://www.elimak.com/pixlib/Get\\_Started\\_Pixlib\\_FD.zip](http://www.elimak.com/pixlib/Get_Started_Pixlib_FD.zip)
- PIXIOC  
<http://osflash.org/projects/pixlib/pixioc>
- KAIROS  
<http://code.google.com/p/kairos/>
- PIXSWX  
<http://code.google.com/p/pixswx/>
- LOWRA  
<http://www.lowra.org/>