

IHM - ERGONOMIE - PROTOTYPAGE

ESIEE 2017

Jour 2 - Conception, API Swing

Nicolas MORTIER / Guillaume KUOCH

MBDA France

nicolas.mortier@mbda-systems.com / guillaume.kuoch@mbda-systems.com

AGENDA DU JOUR

Concepts et patterns du développement d'IHM

L'API Java Swing

L'API Java 2D

CONCEPTS ET PATTERNS



"LE" PATTERN

Modèle

Vue

Contrôleur



EN FAIT PLEIN DE VARIANTES

Modèle Vue Adapteur

Modèle Vue Présentation

Modèle Vue Vue Modèle

Présentation Abstraction Contrôle

...



L'IMPORTANT EST DE DÉCOUPER

L'accès aux données

Le modèle de données

La présentation des données



D'AUTRES PATTERNS UTILES

Singleton, très décrié, mais parfois pratique

Observateur

Décorateur

THREADING

Tous* les frameworks d'IHM utilisent une boucle d'événements!

Il ne faut pas la bloquer!

Ni faire n'importe quoi avec!

* Vérifié avec Swing, JavaFX, MFC, QT



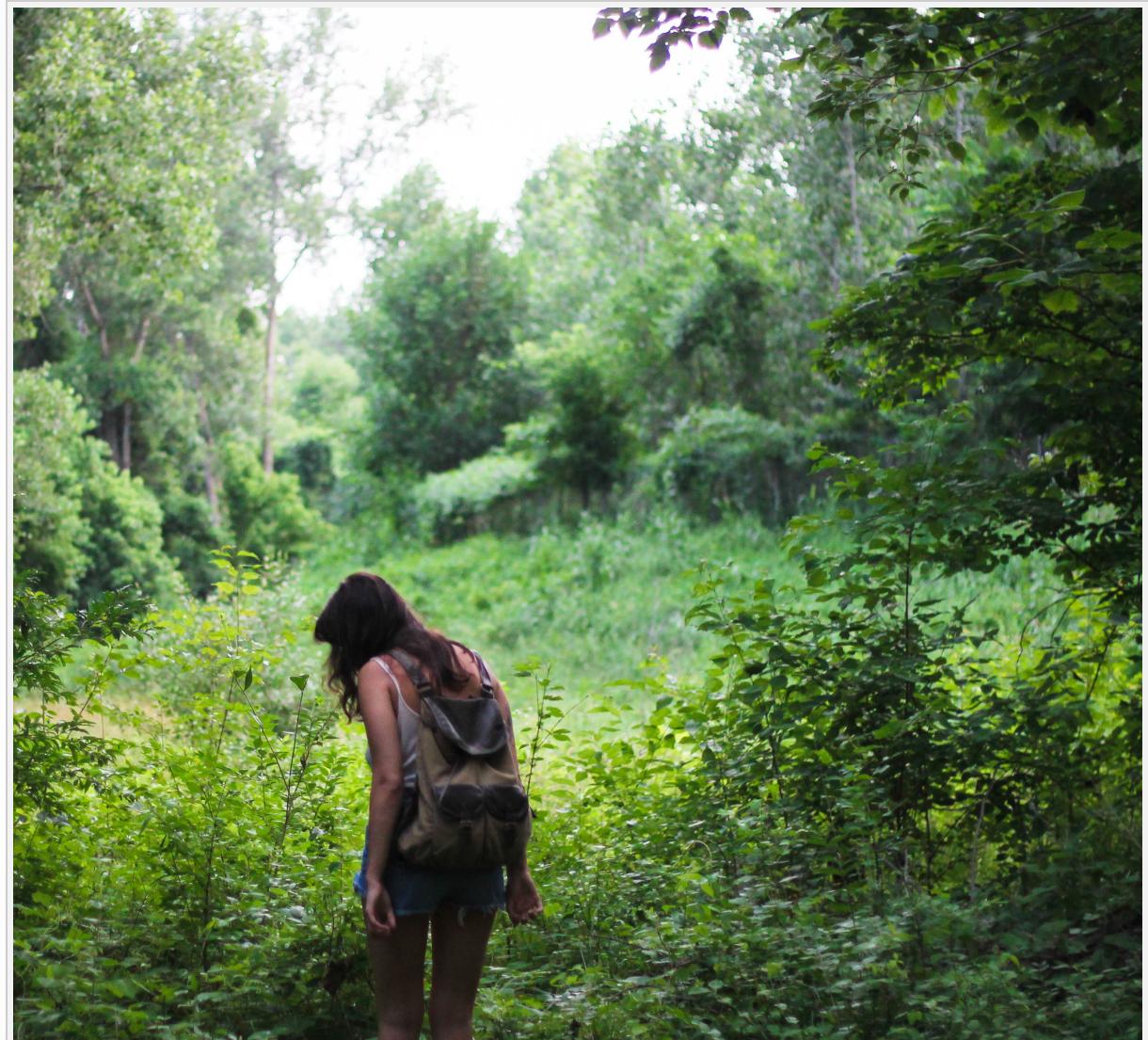
TESTS, UN POINT DIFFICILE

Les tests unitaires ne suffisent pas...

Penser aux tests automatisés

Les tests « scriptés » sont coûteux...

Penser aux tests exploratoires



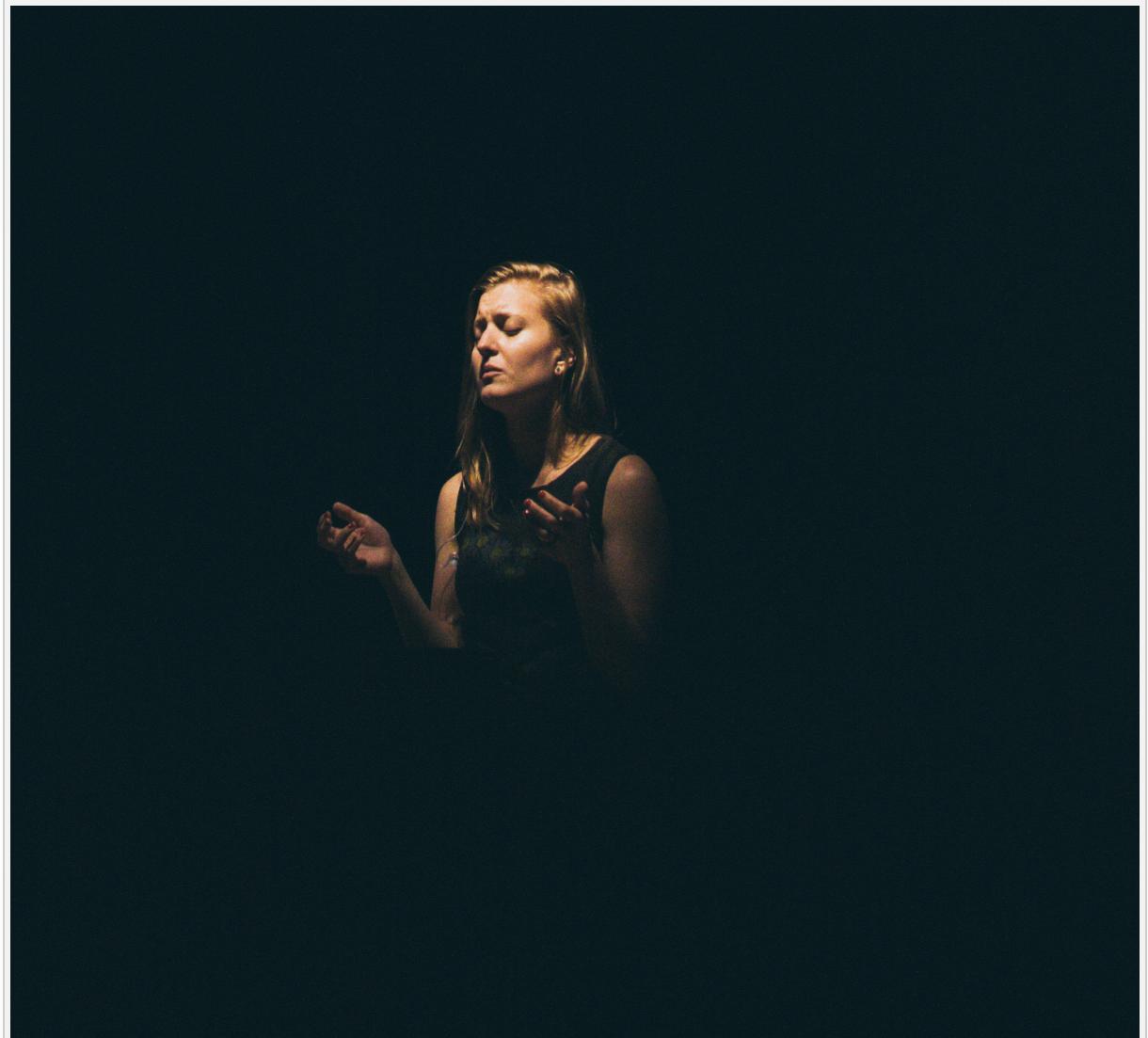
ET LES SPÉCS?

Point très difficile

Souvent en langage naturel, elles sont interprétables

Pour l'IHM il faut éviter de mélanger le quoi et le comment

Conserver uniquement ce que **doit** faire l'application



L'API JAVA SWING



POURQUOI SWING ?

“

Today's problems come from yesterday's solutions

– Brian Goetz @BrianGoetz

UN PEU D'HISTOIRE

Swing n'est plus tout jeune (JDK 1.2 - 1998)

Arrivé en complément d'AWT amène des composants légers

Il a été (et reste?) très utilisé donc vous risquez de le rencontrer...

Les bugs sont connus, les tutoriaux et bibliothèques nombreux

Depuis quelques années JavaFX essaie de le remplacer

JAVA FX

Premières versions catastrophiques

Depuis le JDK 8, stable... ou presque

Oracle ne fournit plus d'exécutable de
SceneBuilder

Petite appréhension pour se lancer sur
de gros projets



TOUTEFOIS...

Si vous avez trop fait de Swing

Si vous n'aimez pas du tout ça

Si vous n'avez pas peur d'en savoir plus
que les profs

**Vous pouvez faire le projet en Java FX,
Angular, C#, Qt, ...**



CONTAINERS ET LAYOUTS

Au premier niveau: **JFrame & JDialog** sont des containers

Les autres containers les plus utilisés: **JPanel, JScrollPane et JTabbedPane**

<http://docs.oracle.com/javase/tutorial/uiswing/components/toplevel.html>

A un container il faut définir un **Layout** qui organisera les composants à afficher

<http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

EXAMPLE SIMPLE

```
JFrame frame = new JFrame("Title");  
  
frame.setLayout(new BorderLayout());  
  
frame.add( new JButton("Click me!", BorderLayout.CENTER);
```

Un peu plus de détails dans le TP01.

COMPOSANTS

L'API Swing vient avec tout le lot classique de composants:

Button, Label, Textfield, ...

Penser à mettre un **J** devant....

[http://docs.oracle.com/javase/tutorial/uiswing/components/componentlist.html\[\]](http://docs.oracle.com/javase/tutorial/uiswing/components/componentlist.html)

[http://da2i.univ-lille1.fr/doc/tutorial-java/ui/features/components.html\[\]](http://da2i.univ-lille1.fr/doc/tutorial-java/ui/features/components.html)

LISTENERS

Omniprésents dans Swing

```
XXXListener lst = new Xlistener();  
componentY.addXListener(lst);  
componentY.removeListener(lst);
```

Les méthodes présentes dans le listener (callbacks) sont appelées lorsqu'un type d'événement survient.

<http://docs.oracle.com/javase/tutorial/uiswing/events/api.html>

On va mettre en pratique avec le TP02.

EVENT DISPATCH THREAD

“

In general Swing is not thread safe. All swing components and related classes, unless otherwise documented, must be accessed on the event dispatch thread.

– Javadoc

La méthode `SwingUtilities.invokeLater()` est votre allié.

```
SwingUtilities.invokeLater( () -> doSomethingOnEDT() );
```

UN PEU DE CODE

```
public static void main(String[] args)
{
    JFrame f = new JFrame();
    f.setLayout( new BorderLayout() );
    f.add( new JPanel(), BorderLayout.CENTER );
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setVisible(true);
}
```

Ce code n'est pas correct! Pourquoi?

L'API JAVA2D



JAVA2D C'EST QUOI?

« The Java 2D API provides two dimensional graphics, text, and imaging capabilities for Java programs through extensions to the Abstract Windowing Toolkit (AWT) »



LE POINT D'ENTRÉE

Quelques bouts de code avant le TP03...

```
private static class DrawPanel extends JPanel
{
    @Override
    protected void paintComponent(Graphics g)
    {
        super.paintComponent(g);

        Graphics2D g2d = (Graphics2D) g;

        g2d.drawRect(20, 20, 100, 100);
    }
}
```

UNE MACHINE À ÉTATS

```
private void stateSafeMethod(Graphics2D g2d)
{
    // Save current state for painting
    Paint oldPaint = g2d.getPaint();

    // Change color
    g2d.setPaint( Color.RED );

    g2d.fillRect(20, 20, 100, 100); // Will be drawn in RED

    // Restore paint
    g2d.setPaint( oldPaint );

    g2d.fillRect(120, 120, 100, 100); // Will be drawn in previous color
}
```

SHAPES : BEYOND POINTS AND LINES

```
GeneralPath path = new GeneralPath(); // Instance de Shape  
  
path.moveTo( x, y ); // Crayon levé, on part au début  
  
for (int i = 1; i < points.length; i++) // On dessine nos points  
{  
    path.lineTo( points[i].x, points[i].y );  
}  
  
path.closePath(); // On ferme le chemin  
  
g2d.fill(path);  
g2d.draw(path);
```

QUELQUES TIPS

Le point {0,0} est en haut à gauche, l'axe des Y orienté vers le bas

Mieux vaut changer le contexte que de faire des maths

La classe `AffineTransform` est votre amie

Attention quand même aux maths!

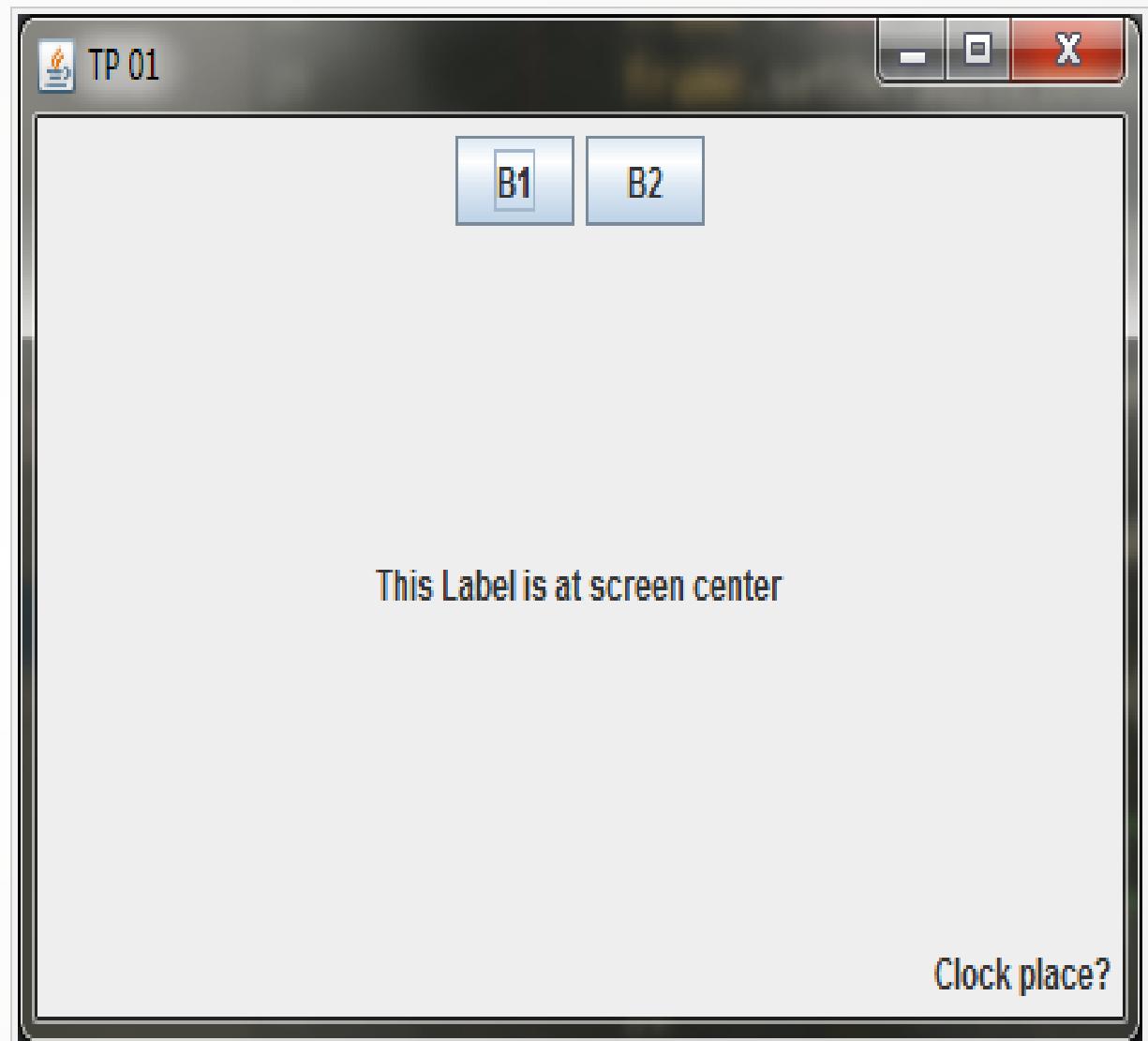
Pour faire joli:

```
g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
```

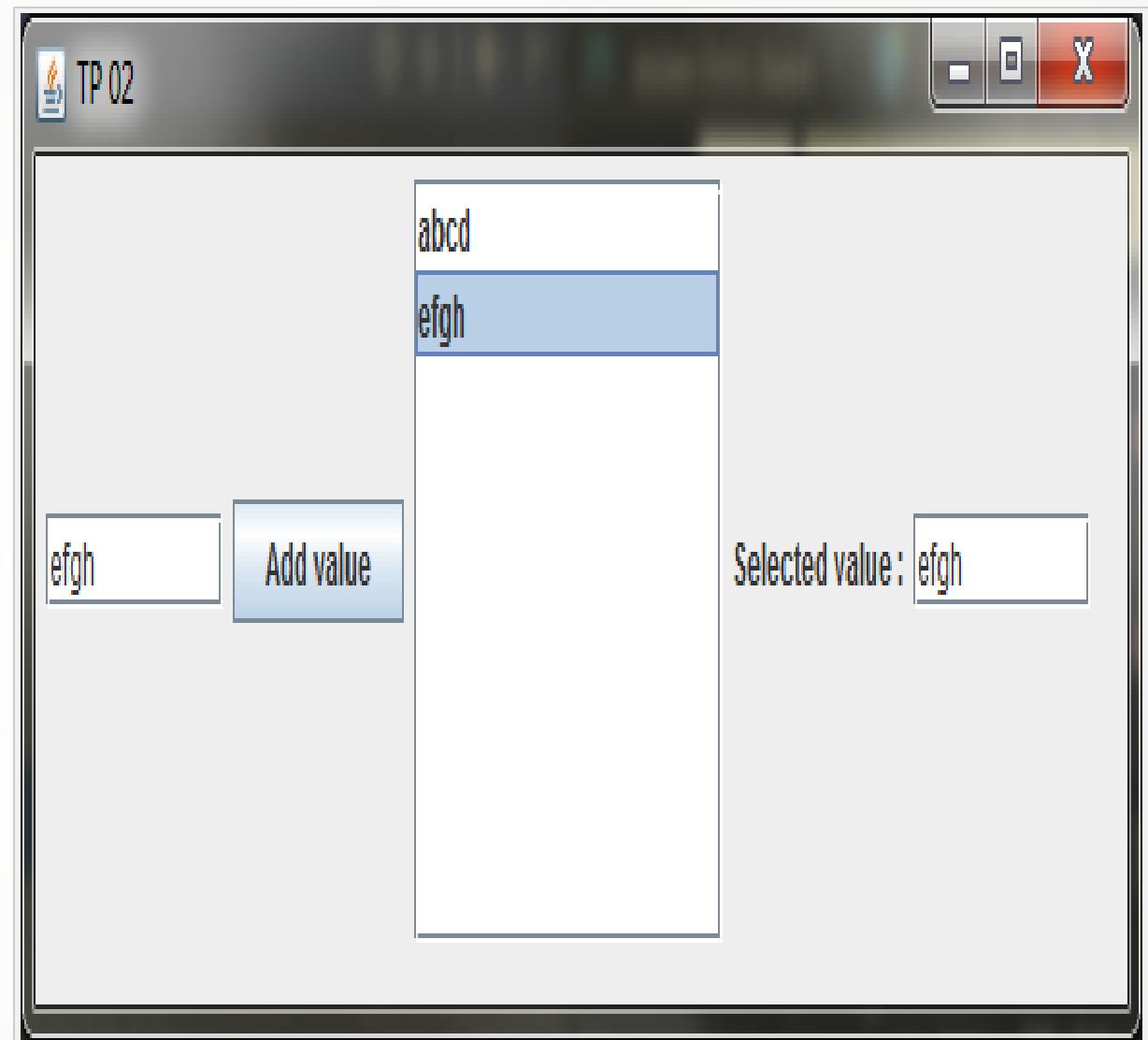
DES TPS

Dans le package com.esiee.mbdaihm.tps

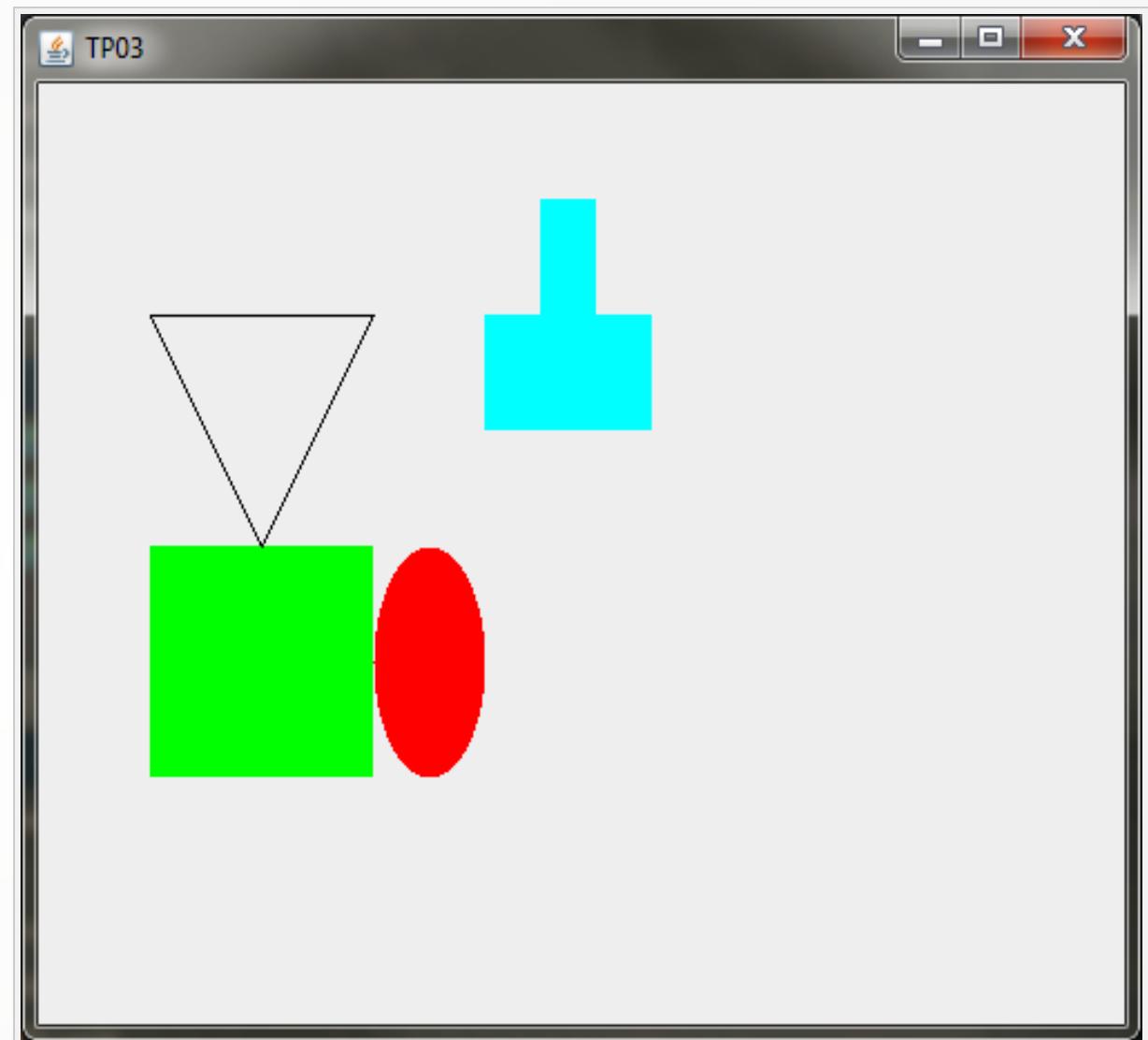
TP1



TP2



TP3



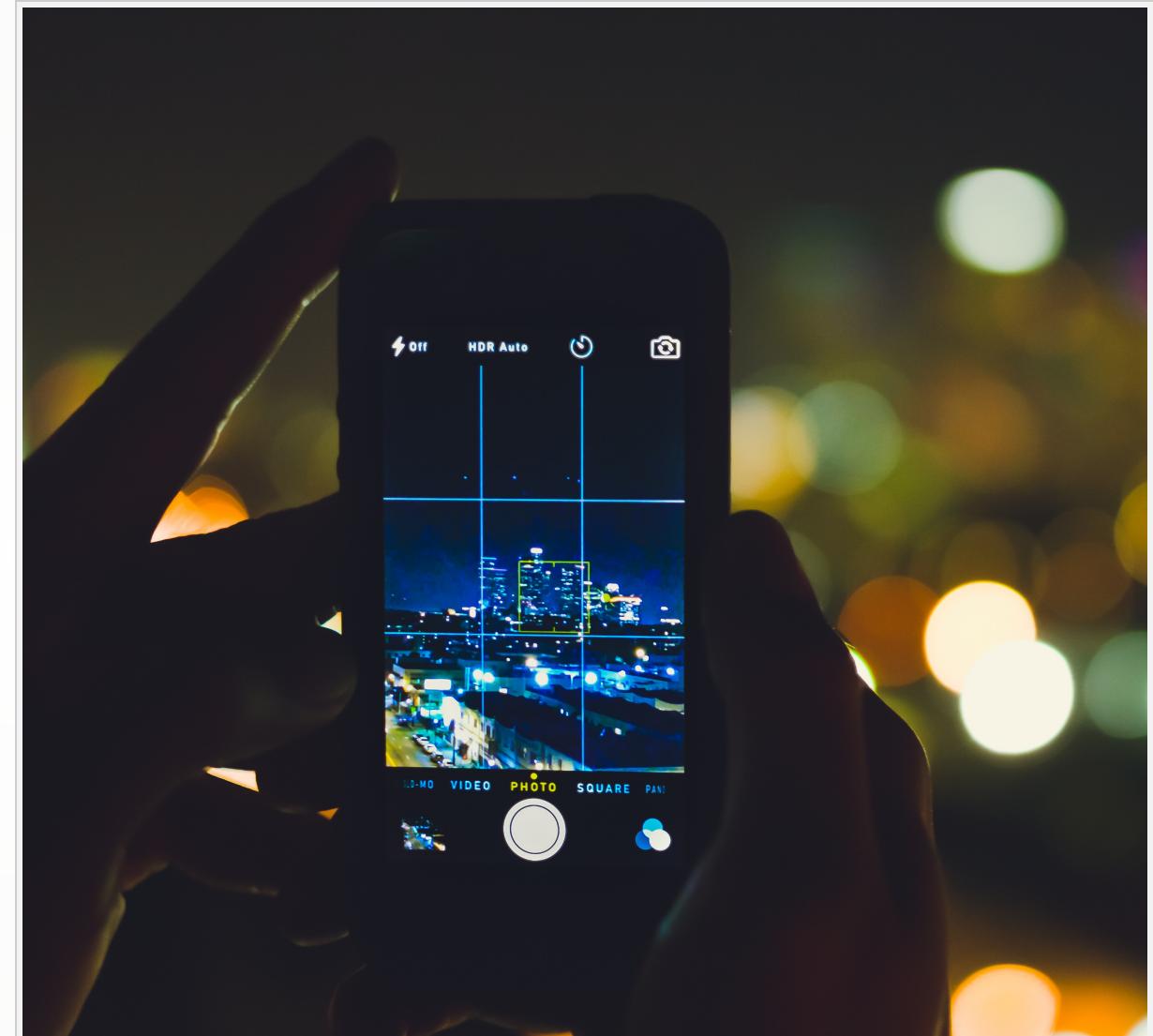
IMAGES

Licence Creative Commons 0

[unsplash.com](#)

Shlomit Wolf
Patrick Goethe
Aleks Dorohovich
Dominik Schröder
Luis Llerena
Glen Noble
Galymzhan Abdugalimov
Forrest Cavale

[pexels.com](#)



LIENS 1/2

Patterns d'IHM:

https://en.wikipedia.org/wiki/Architectural_pattern

Swing:

<http://docs.oracle.com/javase/tutorial/uiswing/components/toplevel.html>

<http://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

<http://docs.oracle.com/javase/tutorial/uiswing/components/componentlist.html>

<http://da2i.univ-lille1.fr/doc/tutorial-java/ui/features/components.html>

<http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>

<http://docs.oracle.com/javase/tutorial/uiswing/events/api.html>

LIENS 2/2

Java 2D:

<https://docs.oracle.com/javase/tutorial/2d/>