

## MyPhotoshop

Adrien Toubiana   Nils Layet   Tania Situm



April 18, 2016



# Copyright

This document is for internal use only at EPITA <<http://www.epita.fr/>>.

Copyright © 2015/2016 Assistants <[yaka@yaka.epita.fr](mailto:yaka@yaka.epita.fr)>.

Copying is allowed only under these conditions:

- You must have downloaded your copy from the Assistants' Intranet <<https://www.acu.epita.fr/>>.
- You must make sure you have the latest version of this document.
- It is your responsibility to make sure that this document stays out of reach of students or individuals outside your year (your "promotion").

- ① Introduction
- ② Mandatory part
- ③ MVC Design pattern
- ④ Free part
- ⑤ Conclusion



## ① Introduction

### ■ Goals

## ② Mandatory part

## ③ MVC Design pattern

## ④ Free part

## ⑤ Conclusion



## Goals

# Overview

- A fun and sexy project
- Two parts: Mandatory and free bonuses
- Mandatory is easy
- A lot of possible features
- Evaluated in defense



# What is it?

- A graphical image processing application using **Swing**
- Support filter plugins
- Use design patterns



## ① Introduction

## ② Mandatory part

- What you have to do
- Provided tools

## ③ MVC Design pattern

## ④ Free part

## ⑤ Conclusion





What you have to do

# Image handling

- You have to handle (at least): **bmp, jpg, gif, png**
- Open and save images
- Apply filters



# Project handling

- Create, open, modify, save and close a project
- Image handling
- Serialization
- History
- Custom format **\*.myPSD**
- Architectural design patterns



# Filter

- You have to code filters
- Respect the *Filter* interface



# Filters handling

- Loading JAR files
- *Filter* interface
- Cancellable actions
- Asynchronous execution



# History

- Each action is logged
- History saved with a project
- Undo and redo what you have done



# Project tabs

- Open each image or project in a different tab
- You have to be able to switch and close tabs independently
- Tabs display the project name



# Mandatory features

- Implement some mandatory features from the list given in the subject
- The choice of the features is *free*
- Even if the choice is free, this part is mandatory





## Provided tools

## Some sexy tools for you

- *ImagePanel* to serialize *BufferedImage*
- *Filter* interface
- Two beautiful filters: *AllBlue* and *VeryLong*
- Base of the projects:
  - `build.xml`
  - `build.properties`
  - All needed JARs for JUnit and Hamcrest



① Introduction

② Mandatory part

③ MVC Design pattern

■ The architectural MVC design pattern

④ Free part

⑤ Conclusion

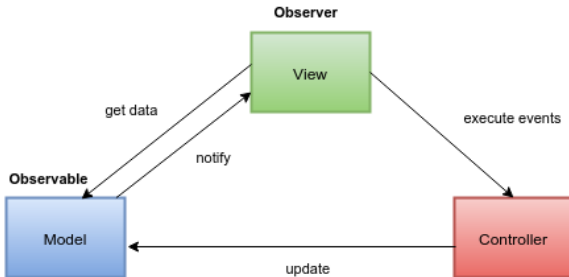


## The architectural MVC design pattern

# What is it for ?

- Organize the components of a graphical user interface
- Decouple code to simplify dependencies
- Reduce code complexity
- Factorize and reuse code
- Improve code quality

# MVC architecture



- **Model:** Represents the state and the behavior of the component and notifies the view of any change.
- **View:** Manages the visual display of the state represented by the model (user interface).
- **Controller:** Interprets the user interaction between the view and the model. This is where all the logic is implemented.

# Event handling

- The Model needs to notify the View of any change so that it can be updated
- The Observer design pattern provides a mechanism to alert other objects of any state change
- Other event handling patterns satisfy the requirements pretty well too



# ButtonModel Interface

- Get inspired by this piece of code of the ButtonModel interface used for regular buttons.

```
public interface ButtonModel extends ItemSelectable {  
    /**/  
  
    boolean isSelected();  
  
    void setSelected(boolean var1);  
  
    void addActionListener(ActionListener var1);  
  
    /**/  
}
```



## ① Introduction

## ② Mandatory part

## ③ MVC Design pattern

## ④ Free part

### ■ Freedom!

## ⑤ Conclusion



Freedom!

# Mandatory features

- We provide you with a list of features
- Each feature is set to a level
- You have to choose which features you implement
- The total level of your features must be **at least 6**



# List of features

Here are the main themes of the list:

- Image handling (layers, gif, histogram...)
- Tools (selection, pencil, zoom, eraser...)
- Interface (status bar, look and feel...)
- Misc features



## Free bonus

Even if the whole mandatory part is perfectly done, it's still not enough!

Here is some example of bonus you may want to implement:

- Filter preview
- Cloud storage integration
- Icons and buttons
- All Adobe Photoshop tools...



① Introduction

② Mandatory part

③ MVC Design pattern

④ Free part

⑤ Conclusion

■ Cheating



## Cheating

# Cheating is really bad, right?

## Don't try to cheat!

- Code is easily found on the internet
- YAKAs can find code too
- Crawling the Web since September
- Do not even try





# What is *cheating* in myPhotoshop?

- Filter share
- JAR loading
- Swing interface
- Every code found on the internet, even incomplete
- Using a friend's file
- Using an external source manager



# What cheating is not

- Filter algorithms
- Asking an assistant for help
- Being at school
- If you want to use other JARs, post a news with the tag [MPH]



# Evaluation

- Only evaluated in defense with an assistant!



Any questions?

