

# Programmation Graphique

## PO2 (2014-2015)

### Chapitre 6 MVC

- MVC et Observable-Observer
- Exemples

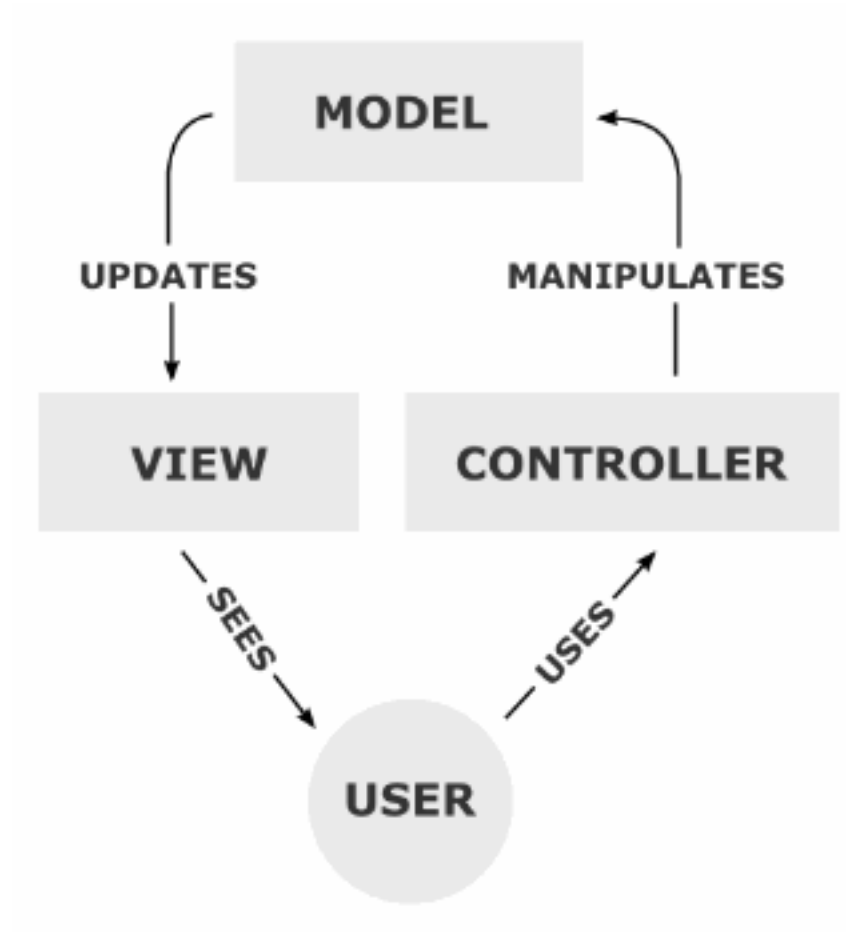
# MVC (Model/View/Controller)

- Afin de séparer les responsabilités des différents composants d'une GUI, il est nécessaire de disposer d'un moyen de synchroniser les données avec leur affichage, et les actions de l'utilisateur pour agir sur ces données.

# MVC

- Problème
  - On souhaite réaliser une GUI souple qui peut avoir plusieurs vues, ou plusieurs contrôleurs
- Solution
  - Le Modèle contient les données et les traitements des données
  - La Vue affiche les résultats des traitements effectués par le modèle
  - Le Contrôleur analyse la requête de l'utilisateur et demande au modèle d'effectuer les traitements

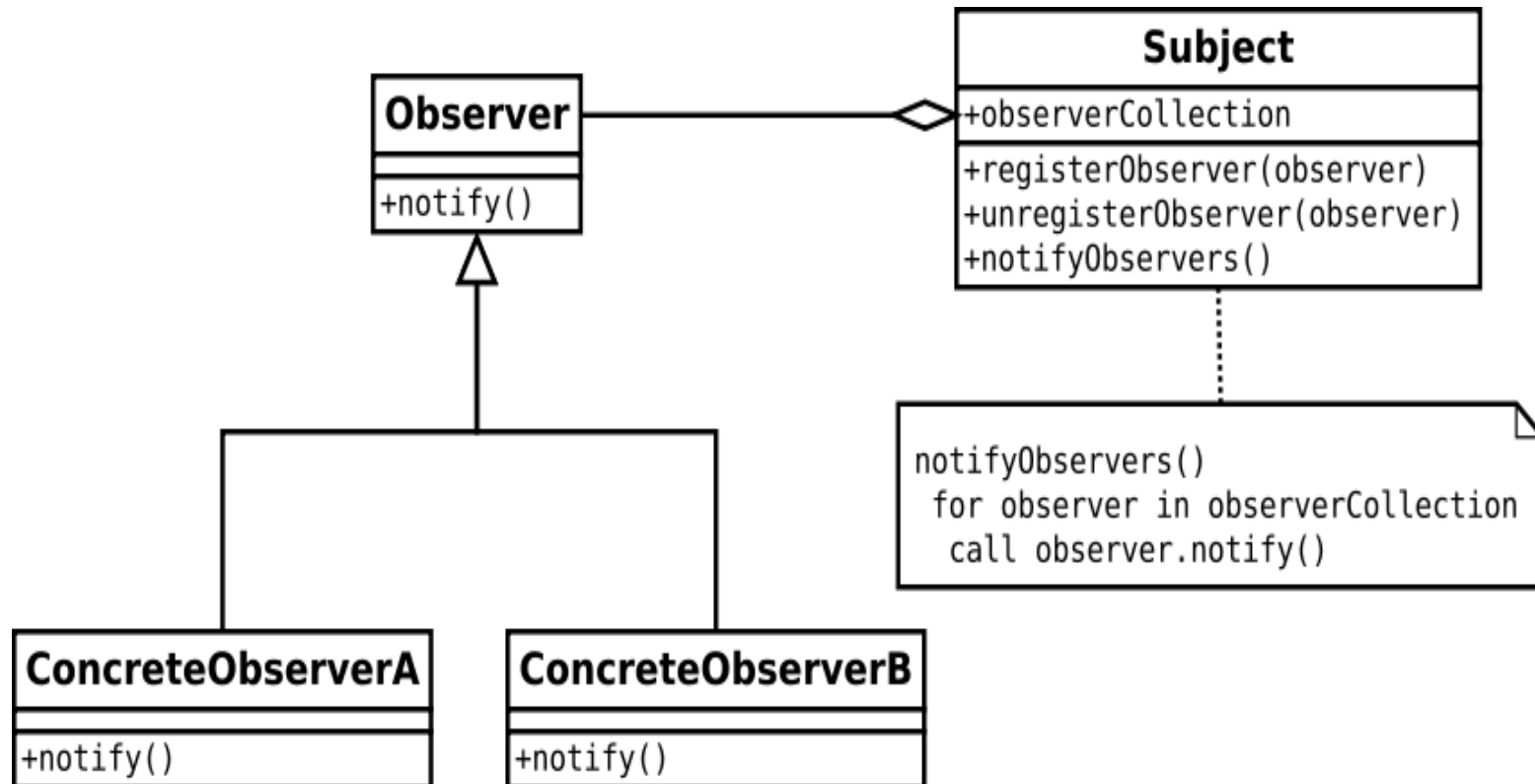
# MVC



# MVC et Observable-Observer

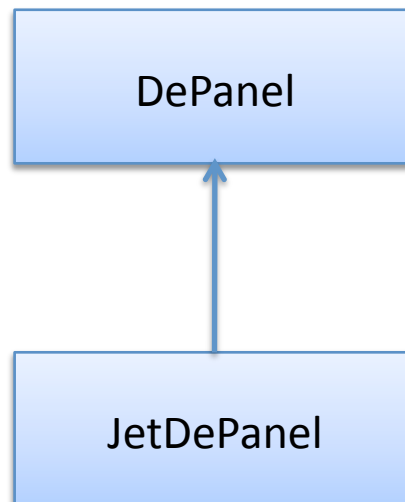
- Le MVC peut être réalisé par le pattern Observable-Observer
  - Modèle = Observable
  - Vue = Observer
  - Controleur = Observable-Observer

# Observateur/Observé



# Exemple : sans MVC

Le modèle et l'affichage ne sont pas séparés



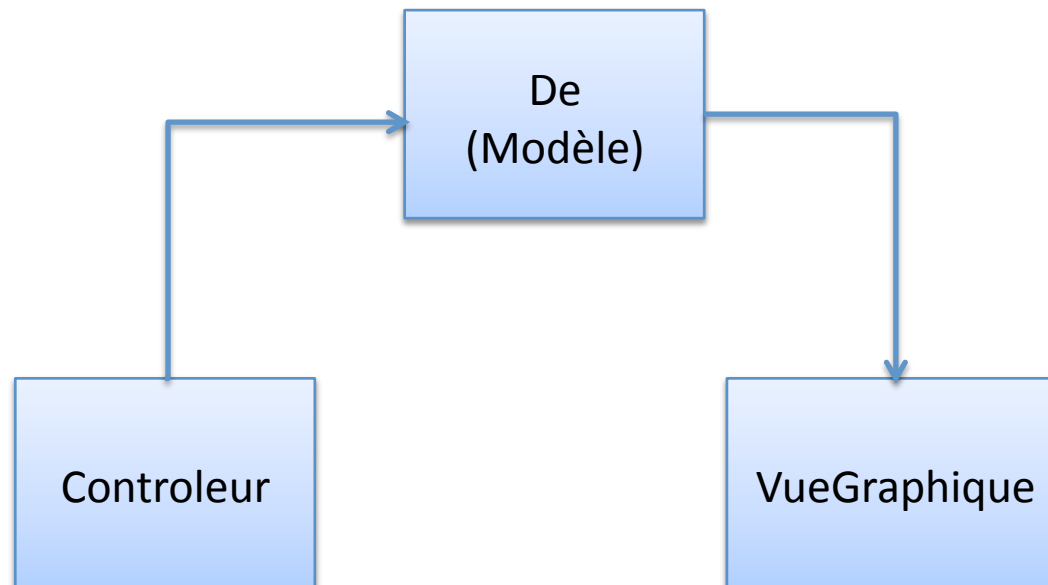


# Exemple : sans MVC

```
public class DePanel extends JPanel {
    private static final int SPOT_DIAMETER = 10;
    private int val;
    public DePanel() {
        setBackground(Color.white);
        setPreferredSize(new Dimension(80,80));
        val = (int)(Math.random()*6)+1;
    }
    public int getVal() {
        return val;
    }
    public void setVal(int val) {
        this.val = val;
    }
    public void jet() {
        val = (int)(6*Math.random() + 1);
        repaint();
    }
    public void paintComponent(Graphics g) {
        .....
    }
    private void drawSpot(Graphics g, int x, int y) {
        .....
    }
}
```

# Exemple : avec MVC

Le modèle et l'affichage sont séparés



# Exemple : avec MVC

```
public class De extends Observable {
    private VueGraphique v;
    private int val=0;
    public De() {
        val = (int)(Math.random()*6)+1;
    }
    public De(int val) {
        this.val = val;
    }
    public De(De d) {
        val = d.val;
    }
    public int getVal() {
        return val;
    }
    public void jet(){
        val = (int)(Math.random()*6)+1;
        setChanged();
        notifyObservers(new Integer(val));
    }
    public boolean equals(Object obj) {
        .....
    }
}
```

# Exemple : avec MVC

```
public class VueGraphique extends JPanel implements
Observer {
    private static final int SPOT_DIAMETER = 10;
    private int val;
    public VueGraphique() {
        setBackground(Color.white);
        setPreferredSize(new Dimension(100,100));
    }
    public void update(Observable o, Object arg) {
        Integer valObj = (Integer) arg;
        val = valObj.intValue();
        repaint();
    }

    public void paintComponent(Graphics g) {
        .....
    }
    private void drawSpot(Graphics g, int x, int y) {
        .....
    }
}
```

# Exemple : avec MVC

```
public class Controleur extends JPanel {
    private De de;
    private JButton jetButton;

    Controleur() {
        // creer un bouton pour jeter les deux De et ajouter
listener
        jetButton = new JButton("Jet");
        // classes interne anonyme
        jetButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                de.jet();
            }
        });

        //... placer components
        this.add(jetButton, BorderLayout.NORTH);
    }

    void setModel(De de) {
        this.de = de;
    }
}
```

# Exemple : avec MVC

```
public class JetDePanel extends JPanel {
    JetDePanel() {
        // creer modeles
        De deGauche = new De();
        De deDroit = new De();

        // creer vues
        VueGraphique vueGauche = new VueGraphique();
        deGauche.addObserver(vueGauche);
        VueGraphique vueDroit = new VueGraphique();
        deDroit.addObserver(vueDroit);
        VueTexte vueGaucheTexte = new VueTexte();
        deGauche.addObserver(vueGaucheTexte);
        VueTexte vueDroitTexte = new VueTexte();
        deDroit.addObserver(vueDroitTexte);

        // creer controleurs
        Controleur conGauche = new Controleur();
        conGauche.setModel(deGauche);
        Controleur conDroit = new Controleur();
        conDroit.setModel(deDroit);
    }
}
```

# Exemple : avec MVC

```
JLabel label1 = new JLabel("De1");
JLabel label2 = new JLabel("De2");
add(label1);
add(label2);
add(conGauche);
add(conDroit);
add(vueGauche);
add(vueDroit);
add(vueGaucheTexte);
add(vueDroitTexte);
    }
}
```