

**MonObjet**

Ahcène Bounceur

**VERSION 1**

# Classe MonObjet

```
Class MonObjet {  
  
}
```

# Classe MonObjet

```
// Attributs  
private int x;  
private int y;  
private int longueur;  
private int largeur;  
private Color couleur;  
private String nom;
```

# Classe MonObjet

```
// Constructeurs
public MonObjet(int x, int y, int longueur, int
    largeur, Color couleur, String nom) {
    setX(x);
    setY(y);
    setLongueur(longueur);
    setLargeur(largeur);
    setCouleur(couleur);
    setNom(nom);
}

public MonObjet()
    {this(0,0,100,80,Color.black,"");}
```

# Classe MonObjet

```
// Accesseurs
```

```
    public void setX(int x) {this.x = x;}
    public void setY(int y) {this.y = y;}
    public void setLongueur(int longueur) {
this.longueur = longueur;}
    public void setLargeur(int largeur) {
this.largeur = largeur;}
    public void setCouleur(Color couleur) {
this.couleur = couleur;}
    public void setNom(String nom) {
this.nom = nom;}
```

```
.../...
```

# Classe MonObjet

.../...

```
public int getX() {return x;}  
public int getY() {return y;}  
public int getLongueur() {return longueur;}  
public int getLargeur() {return largeur;}  
public Color getCouleur() {return couleur;}  
public String getNom() {return nom;}
```

# Classe MonObjet

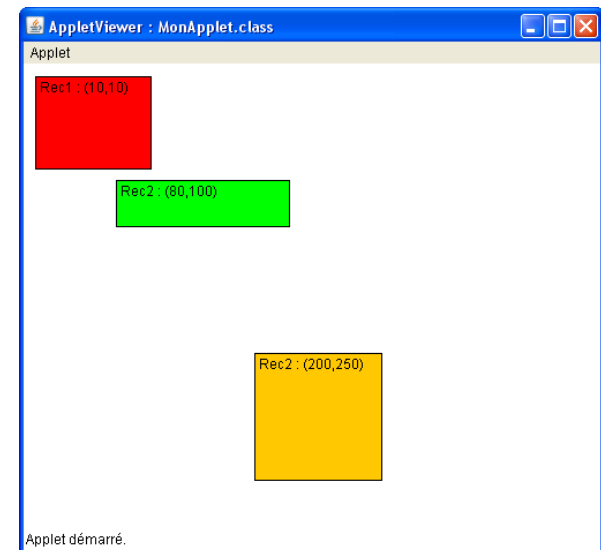
```
// Les Méthodes
```

```
public void draw(Graphics g) {  
    g.setColor(couleur);  
    g.fillRect(x, y, longueur, largeur);  
    g.setColor(Color.black);  
    g.drawRect(x, y, longueur, largeur);  
    g.drawString(nom+" : ("+x+", "+y+")",  
                 x+4, y+14);  
}
```



# Classe MonApplet

```
public class MonApplet extends JApplet {  
    private MonObjet objet1;  
    private MonObjet objet2;  
    private MonObjet objet3;  
  
    public void init() {  
        this.resize(500, 400);  
        objet1 = new MonObjet(10,10,100,80,Color.red,"Rec1");  
        objet2 = new MonObjet(80,100,150,40,Color.green,"Rec2");  
        objet3 = new MonObjet(200,250,110,110,Color.orange,"Rec2");  
    }  
  
    public void paint(Graphics g) {  
        objet1.draw(g);  
        objet2.draw(g);  
        objet3.draw(g);  
    }  
}
```



# Classe Panneau

```
public class Panneau extends JPanel {  
  
    private MonObjet objet1;  
    private MonObjet objet2;  
    private MonObjet objet3;  
  
    public Panneau() {  
        super() ;  
        objet1 = new MonObjet(10,10,100,80,Color.red,"Rec1");  
        objet2 = new MonObjet(80,100,150,40,Color.green,"Rec2");  
        objet3 = new MonObjet(200,250,110,110,Color.orange,"Rec2");  
    }  
  
    .../...
```

# Classe Panneau

.../...

```
@Override  
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    objet1.draw(g);  
    objet2.draw(g);  
    objet3.draw(g);  
}
```

# Classe Main

```
public static void main(String[] args) {  
  
    JFrame jf = new JFrame() ;  
  
    Panneau panneau = new Panneau() ;  
    jf.add(panneau);  
  
    jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    jf.setBounds(100, 100, 700, 500);  
    jf.setVisible(true);  
  
}
```

Objet vivant !

# **EVÈNEMENTS DE LA SOURIS**

# Classe MonObjet

Class MonObjet implements

**MouseListener**, **MouseMotionListener**

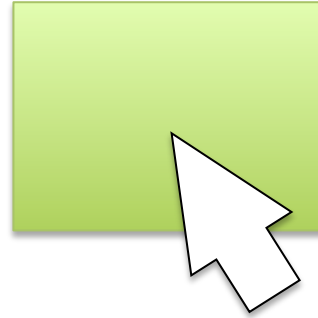
{

}

# Classe Panneau

```
public class Panneau extends JPanel {  
    public Panneau() {  
        super() ;  
        objet1 = new MonObjet(10,10,100,80,Color.red,"Rec1");  
        objet2 = new MonObjet(80,100,150,40,Color.green,"Rec2");  
        objet3 = new MonObjet(200,250,110,110,Color.orange,"Rec2");  
        addMouseMotionListener(objet1);  
        addMouseMotionListener(objet2);  
        addMouseMotionListener(objet3);  
        addMouseListener(objet1);  
        addMouseListener(objet2);  
        addMouseListener(objet3);  
    }  
}
```

.../...



Réagir au passage de la souris

**SOURIS : SUIS-JE À L'INTÉRIEUR ?**



# Classe MonObjet

```
public boolean dansObjet(int x, int y) {  
    if (  
        x > this.x &&  
        x < (this.x + longueur) &&  
        y > this.y &&  
        y < (this.y + largeur)  
    )  
        return true;  
    return false;  
}
```

Réagir au passage de la souris

**JAUNE = SOURIS AU DESSUS !**

# Classe MonObjet

```
@Override
public void mouseMoved(MouseEvent e) {
    surObjet = false;
    if (dansObjet(e.getX(), e.getY()))
        surObjet = true;
}
```

# Classe MonObjet

```
// Les Méthodes
```

```
public void draw(Graphics g) {  
    g.setColor(couleur);  
    if(surObjet)  
        g.setColor(Color.yellow);  
    g.fillRect(x, y, longueur, largeur);  
    g.setColor(Color.black);  
    g.drawRect(x, y, longueur, largeur);  
    g.drawString(nom+" : ("+x+", "+y+")",  
                x+4, y+14);  
}
```

Réagir au clic de la souris

**CADRÉ = OBJET SÉLECTIONNÉ**

# Classe MonObjet

```
// Les Méthodes
public void draw(Graphics g) {
    g.setColor(couleur);
    if(surObjet)
        g.setColor(Color.yellow);
    g.fillRect(x, y, longueur, largeur);
    g.setColor(Color.black);
    g.drawRect(x, y, longueur, largeur);
    g.drawString(nom+" : ("+x+", "+y+")",
                 x+4, y+14);
    if (selectionne) {
        g.setColor(Color.black);
        g.drawRect(x - 2, y - 2, longueur + 4, largeur + 4);
    }
}
```

# Classe MonObjet

```
@Override
```

```
    public void mouseClicked(MouseEvent e) {  
        if (dansObjet(e.getX(), e.getY())) {  
            selectionne = !selectionne;  
        }  
    }  
}
```

Ça bouge

**DÉPLACER UN OBJET**



# Déplacer un objet

- La souris est sur l'objet  
**surObjet = true**
- Mouvement de la souris avec le bouton appuyé (drag)  
**presse = true**

# Classe MonObjet

**@Override**

```
public void mousePressed(MouseEvent e) {  
    if (dansObjet(e.getX(), e.getY())) {  
        presse = true;  
        dx = e.getX() - x;  
        dy = e.getY() - y;  
    }  
}
```

# Classe MonObjet

```
@Override
```

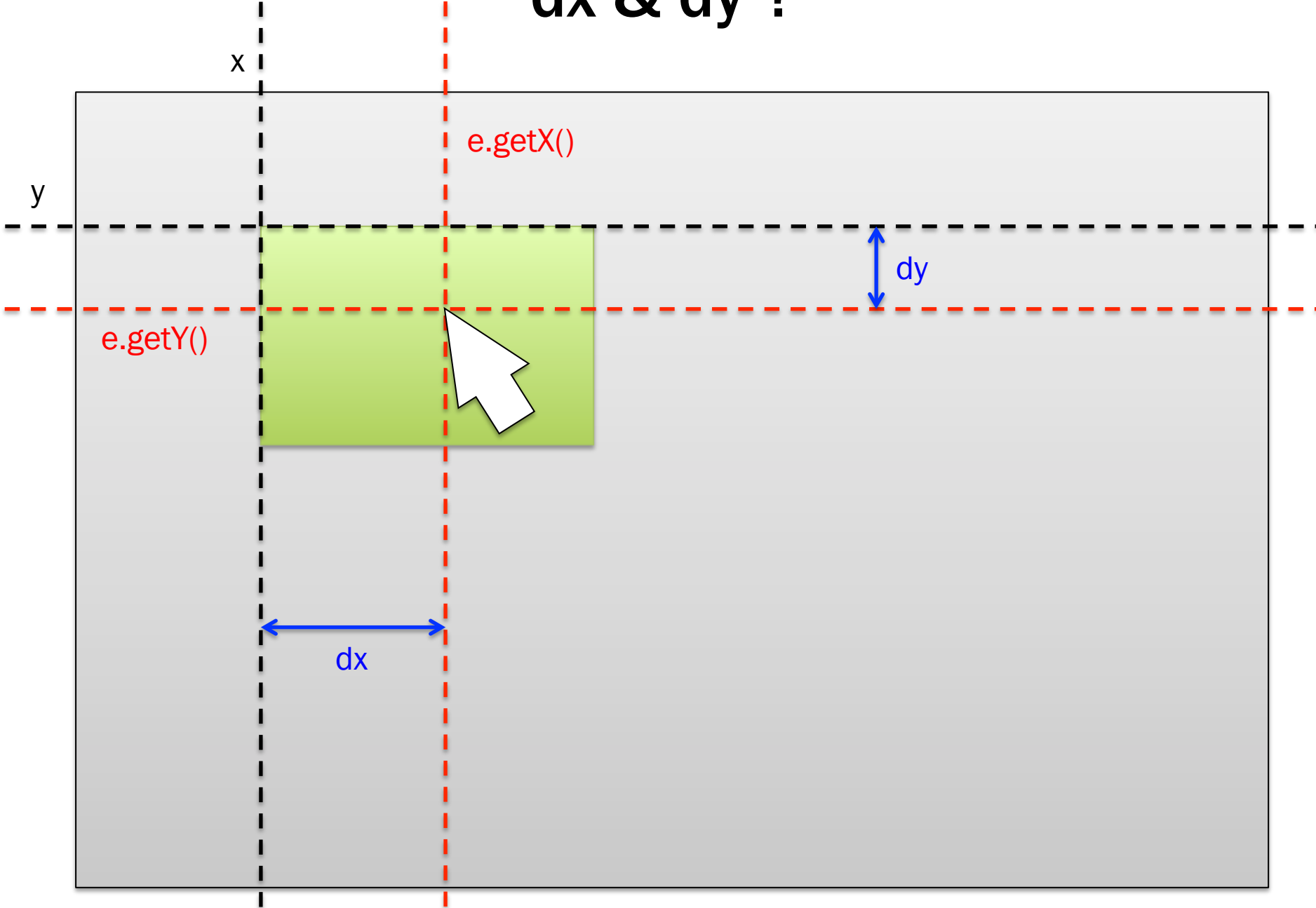
```
    public void mouseReleased(MouseEvent e) {  
        if (dansObjet(e.getX(), e.getY()))  
            presse = false;  
    }
```

# Classe MonObjet

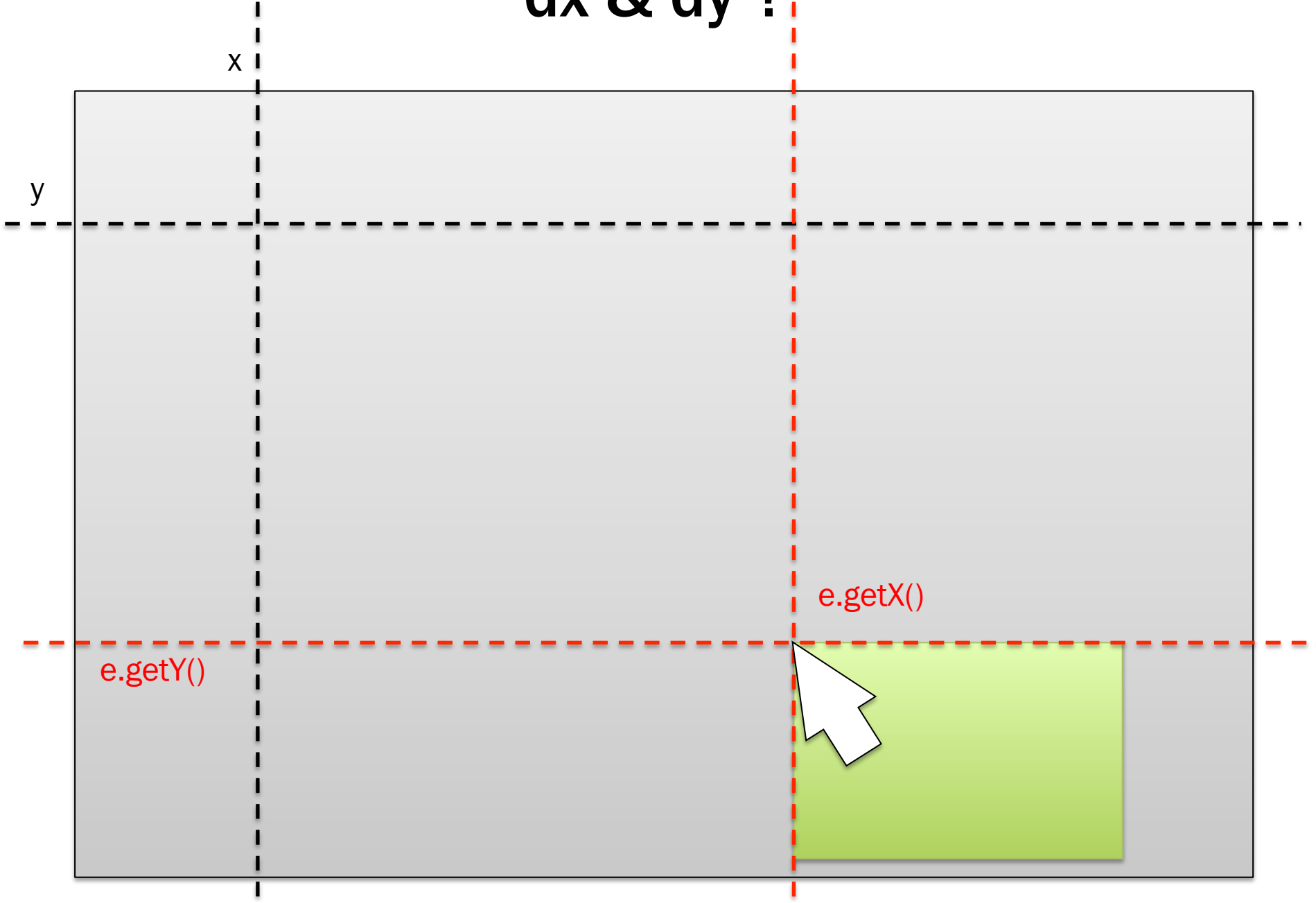
```
@Override
```

```
public void mouseDragged(MouseEvent e) {  
    if (surObjet && presse) {  
        x = e.getX() - dx;  
        y = e.getY() - dy;  
    }  
}
```

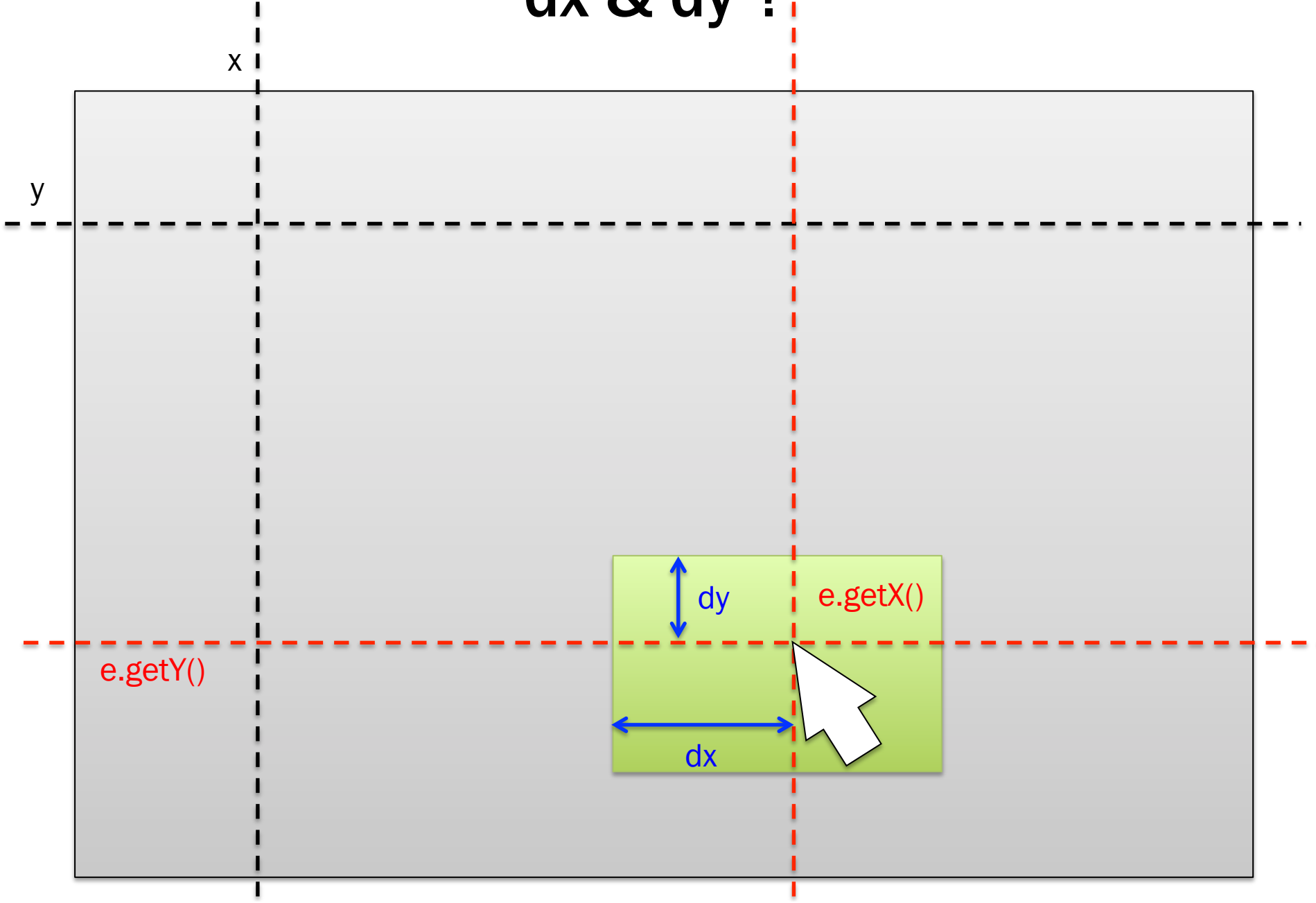
# dx & dy ?



# dx & dy ?



# dx & dy ?



Je ne vois rien moi !

**RAFRAICHIR LE GRAPHISME !!**



# repaint()

- A chaque action sur MonObjet il faut rafraichir le graphisme de la classe Panneau : appeler la méthode **paintComponent()**.
- Pour appeler la méthode **paintComponent** on peut utiliser la méthode **repaint()**
- Il faut donc un lien entre MonObjet et Panneau

# Classe MonObjet

```
private Panneau panneau ;
```

```
public MonObjet(int x, int y, int longueur, int  
    largeur, Color couleur, String nom,
```

```
    Panneau panneau) {  
    setX(x); setY(y);  
    setLongueur(longueur);  
    setLargeur(largeur);  
    setCouleur(couleur);  
    setNom(nom);  
    this.panneau = panneau ;  
}
```

```
public MonObjet(Panneau panneau) {  
    this(0,0,100,80,Color.black,"");  
}
```

# Classe MonObjet

`@Override`

```
public void mouseClicked(MouseEvent e) {  
    if (dansObjet(e.getX(), e.getY())) {  
        selectionne = !selectionne;  
        panneau.repaint();  
    }  
}
```

# Classe MonObjet

```
@Override  
public void mouseMoved(MouseEvent e) {  
    surObjet = false;  
    if (dansObjet(e.getX(), e.getY()))  
        surObjet = true;  
    panneau.repaint();  
}
```

# Classe MonObjet

@Override

```
public void mouseDragged(MouseEvent e) {  
    if (surObjet && presse) {  
        x = e.getX() - dx;  
        y = e.getY() - dy;  
        panneau.repaint();  
    }  
}
```

Et si on profitait (héritait) de SWING directement

## **VERSION 2**

# Une fenêtre

1. Je crée une **Fenêtre** simple (héritant de **JFrame**) :

```
public class Fenetre extends JFrame {  
  
    public Fenetre() {  
        super();  
        setTitle("Mon Application");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setBounds(100, 100, 800, 600);  
        setVisible(true);  
    }  
  
}
```

# Un Plateau

2. Je crée un objet **Plateau** (héritant de **JPanel**) :

```
public class Plateau extends JPanel {  
  
    public Plateau() {  
        super();  
        setBackground(Color.WHITE);  
    }  
  
}
```



# Un Sommet Carré

3. Je crée un objet **Sommet** carré de couleur rouge (héritant de **JPanel**) :

```
public class Sommet extends JPanel {  
  
    public Sommet() {  
        super();  
        setBackground(Color.RED);  
    }  
  
}
```

# Ajouter le plateau à la fenêtre

## 4. J'ajoute le plateau à la fenêtre :

```
public class Fenetre extends JFrame {  
  
    public Fenetre() {  
        Plateau plateau = new Plateau();  
        add(plateau);  
        setTitle("Mon Application");  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setBounds(100, 100, 800, 600);  
        setVisible(true);  
    }  
  
}
```

# Ajouter un écouteur de clic au plateau

5. J'ajoute un écouteur de clic au plateau :

```
public class Plateau extends JPanel implements MouseListener {  
  
    public Plateau() {  
        super();  
        setBackground(Color.WHITE);  
        addMouseListener(this);  
    }  
  
}
```

# Ajouter un sommet à l'endroit cliqué

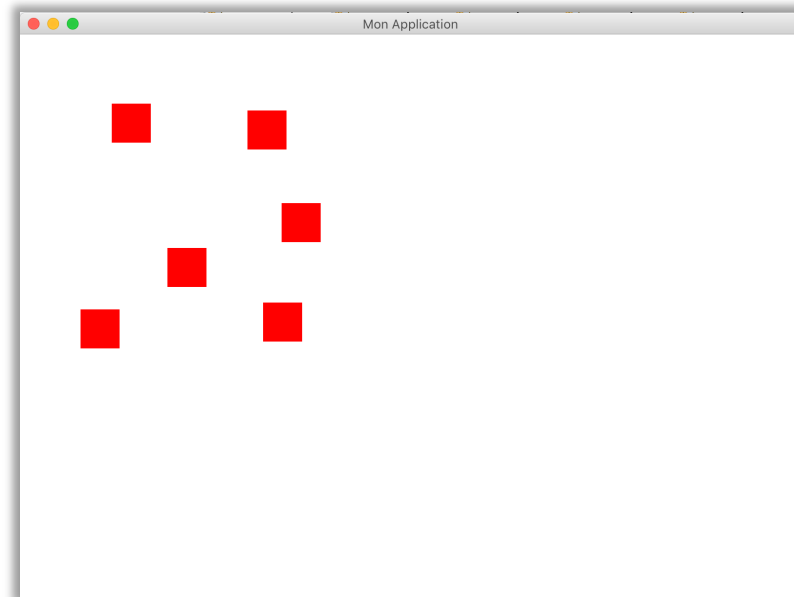
6. A chaque clic sur le plateau on ajoute un Sommet à l'endroit cliqué :

```
public class Plateau extends JPanel implements MouseListener {  
  
    public Plateau() {  
        setBackground(Color.WHITE);  
        addMouseListener(this);  
        setLayout(null);  
    }  
  
    @Override  
    public void mouseClicked(MouseEvent e) {  
        Sommet sommet = new Sommet();  
        sommet.setBounds(e.getX()-20, e.getY()-20, 40, 40);  
        add(sommet);  
        repaint();  
    }  
}
```

# Test

## 7. Testez :

```
public class ClassePrincipale {  
    public static void main(String[] args) {  
        Fenetre fenetre = new Fenetre();  
    }  
}
```




# Ajouter un sommet sous forme de disque

## 8. Transformer la forme du sommet en un disque :

```
public class Sommet extends JComponent {  
  
    public Sommet() {  
        super();  
    }  
  
    @Override  
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        g.setColor(new Color(230, 0, 0, 100));  
        g.fillOval(2, 2, getWidth()-4, getHeight()-4);  
    }  
}
```

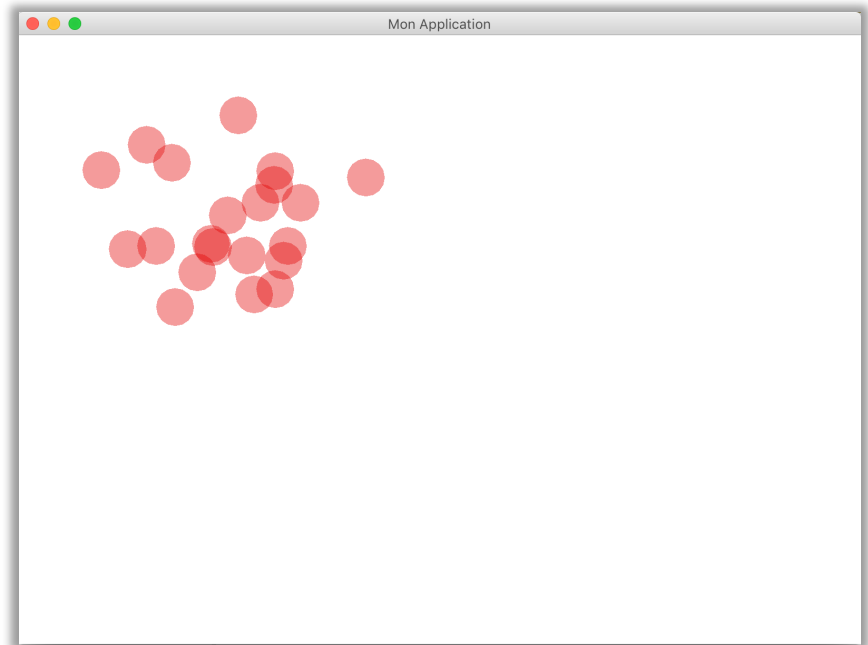
*Dessine un disque rouge transparent*



# Une fenêtre

## 9. Testez :

```
public class ClassePrincipale {  
    public static void main(String[] args) {  
        Fenetre fenetre = new Fenetre();  
    }  
}
```



# Une fenêtre

## 10. Déplacez les sommets :

```
public class Sommet extends JComponent implements MouseMotionListener {

    public Sommet() {
        super();
        addMouseMotionListener(this);
    }

    @Override
    public void paintComponent(Graphics g) { ... }

    @Override
    public void mouseDragged(MouseEvent e) {
        setBounds(e.getX()+getX()-20, e.getY()+getY()-20,
            getWidth(), getHeight());
    }
}
```



# Une fenêtre

## 11. Testez :

```
public class ClassePrincipale {  
    public static void main(String[] args) {  
        Fenetre fenetre = new Fenetre();  
    }  
}
```

# Une fenêtre

## 12. Plus pro !

```
public class Sommet extends JComponent implements MouseMotionListener {
    private int dx, dy;
    ...
    @Override
    public void mousePressed(MouseEvent e) {
        dx = getWidth()/2-e.getX();
        dy = getHeight()/2-e.getY();
    }

    @Override
    public void mouseDragged(MouseEvent e) {
        setBounds(
            e.getX()+getX()-getWidth() /2 + dx,
            e.getY()+getY()-getHeight()/2 + dy,
            getWidth(), getHeight());
    }
}
```

# Questions ?

- Source :

`http://pagesperso.univ-brest.fr/~bounceur/cda`