

MouseEvents gebruiken in NetBeans

Voorbeeld aan de hand van een eenvoudig tekenprogramma. We tekenen enkel rechte lijnen.

Op een hoofdPaneel plaatsen we een tekenPaneel, waarop de tekening komt.

Selecteer dit tekenPaneel in Design View, en via RK > Events > Mouse zie je 5 methoden, dit zijn normaal de 5 methoden die je verplicht moet maken als je de *interface* (zie H11)

MouseListener *implementeert*. In NetBeans moet je echter enkel maken wat je gebruikt.

[NetBeans gebruikt zelf een MouseAdapter, waardoor je niet altijd alle methoden uit een interface moet implementeren].

In ons voorbeeld moet je er twee van uitwerken : **MousePressed** en **MouseReleased**. Om te zien wat je gaat tekenen heb je ook nog **MouseDragged** nodig uit de MouseMotion groep. Als optie kan je ook nog in **MouseEntered** aangeven wat er moet veranderen als de muis het tekenPaneel binnenkomt (vb. vorm van de cursor aanpassen).

Bekijk zelf de code hieronder, hierbij zijn p,q en r van het type Point.

De variabele lijst is een ArrayList van Point objecten, ieder lijnstuk bestaat dus uit 2 elementen in die lijst !

Werking :

- indien de muisknop ingedrukt wordt, bepaalt dit het startpunt van een lijnstuk
- als je de muis versleept (met ingedrukte knop) zie je een zwevende lijn
- pas als je de muisknop loslaat wordt het lijnstuk definitief (eindpunt opnemen)

Hier wat voorbeeldcode, waarbij tekenPaneel het betreffende paneel is :

```
private void tekenPaneelMousePressed(java.awt.event.MouseEvent evt) {
    p = new Point();
    p.x = evt.getX();
    p.y = evt.getY();
    lijst.add(p); // startpunt toegevoegd
    System.out.println("startpunt geklikt" + p.toString());
    r = new Point(p); // deep copy van p , nodig in MouseDragged
}

private void tekenPaneelMouseDragged(java.awt.event.MouseEvent evt) {
    // zwevende lijn nodig, door de oude te wissen in XOR-mode
    // anders veeg je de rest van de tekening uit !
    Graphics g = tekenPaneel.getGraphics();
    g.setXORMode(Color.white);
    g.drawLine(p.x, p.y, r.x, r.y);
    // = de oude lijn wegdoen (tekenen in wit, behalve waar iets staat)
    r.x = evt.getX(); // actuele coördinaten van de muis opnemen
    r.y = evt.getY();
    g.drawLine(p.x, p.y, r.x, r.y); // lijn opnieuw tekenen
}

private void tekenPaneelMouseReleased(java.awt.event.MouseEvent evt) {
    q = new Point();
    q.x = evt.getX();
    q.y = evt.getY();
    lijst.add(q); // eindpunt toegevoegd
    System.out.println("eindpunt geklikt" + q.toString());
}

private void tekenPaneelMouseEntered(java.awt.event.MouseEvent evt) {
    tekenPaneel.setCursor(new Cursor(Cursor.CROSSHAIR_CURSOR));
}
```