

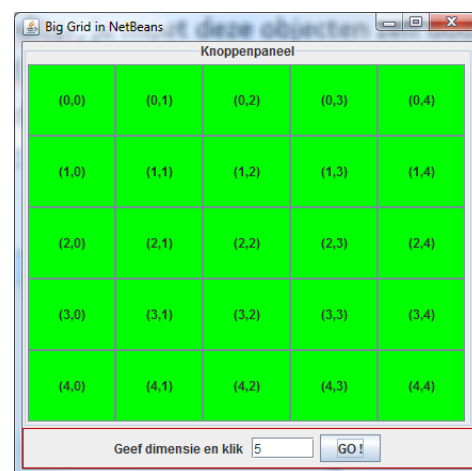
Maak : File > New Project > Java Application > enz ... zoals in de beknopte handleiding (zie Toledo). Een rooster van knoppen of tekstvakken (of nog andere JComponents) kan je NIET via de GUI editor maken, gezien je de pointers naar die objecten in een array of arraylist wil bijhouden (en niet elk object een eigen identifier heeft).

Je moet deze objecten dus via eigen code genereren. In het voorbeeld van *Vb1005_BigGrid.java* (zie de versie gemaakt in JCreator), mogen de knoppen bvb. ook pas gemaakt worden NADAT de gebruiker een dimensie ingetikt heeft, deze componenten kan je dus onmogelijk vooraf plaatsen :

Voor ...



en na klikken :



Om deze layout te bekomen moet je onderstaande stappen volgen, en gebruik exact dezelfde identifiers, zodat de copy/paste code ook in jouw programma werkt :

1. Maak in je JFrame een JPanel, noem dit **mainPanel** + RK> Set Layout : BorderLayout
2. Zet op dit mainPanel nog twee JPanels en geef ze volgende namen :
RK> Change Variable Name : **zuidPaneel** + RK> Set Layout : FlowLayout
centerPaneel + RK> Set Layout : GridLayout
3. Verander bij zuidPaneel > Direction : South
4. Geef beide panelen kleurtjes, border, enz ... wat je maar wil (= opsmuk)
5. zuidPaneel : label, vak, knop ... toevoegen via de gewone weg (GUI editor van NetBeans)
6. op centerPaneel is nu array van knoppen nodig ! → wordt een 2D-array van JButtons ...

2D-array van JButtons (dit moet je dus zelf "programmeren") :

1. helemaal onderaan in de klasse je eigen globale variabelen en de globale array referentie declareren (je moet er in alle methoden aan kunnen):

```
int dim;
private JButton[][] knoppenGrid; // 2D-array van knoppen !
private KnopHandler kh;
```

Instantieer deze knophandler in de Constructor (onder initComponents();)

```
kh = new KnopHandler();
```

2. maak een **factory method** *maakKnop()* (plaats deze juist na de Constructor)

```
// factory method : argumenten : op welk paneel + rij + kolom
public JButton maakKnop ( JPanel cp, int rij, int kolom ) {
    JButton knop = new JButton ( "(" + rij + "," + kolom + " " );
    knop.setBackground( Color.green );
    knop.addActionListener( kh );
    cp.add( knop );
    return knop; // stuur ref terug om in array op te bergen
}
```

3. in de *actionPerformed()* (die bij de GO knop hoort) moet code komen in deze zin :

```
dim = Integer.parseInt( jTextField1.getText());
knoppenGrid = new JButton[dim][dim];
centerPaneel.removeAll(); //voor je een nieuwe grid vult, de oude wissen

// hier bepaal je dus pas hoeveel knoppen er zullen zijn
centerPaneel.setLayout (new GridLayout (dim, dim));

for (int rij=0; rij<dim; rij++){
    for (int kolom=0; kolom<dim; kolom++){
        knoppenGrid[rij][kolom] = maakKnop (centerPaneel, rij, kolom);
    }
}

centerPaneel.revalidate();
```

Dit doet het volgende :

- initialiseer de array van JButtons
- print op iedere JButton de corresponderende rij en kolom
(dit mag je vervangen door iets anders natuurlijk)
- voeg iedere knop toe aan de GridLayout van het centerPaneel + herteken het paneel

Klaar !

Natuurlijk moet je dan zelf nog de code voor je KnopHandler schrijven ;-)

Veel succes !