

Exercice 1: Bouger une tortue

Développez une application permettant de dessiner et de déplacer une tortue à l'écran.

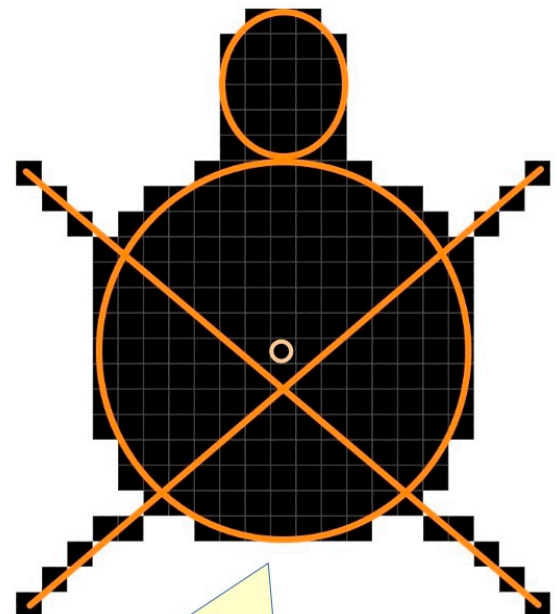
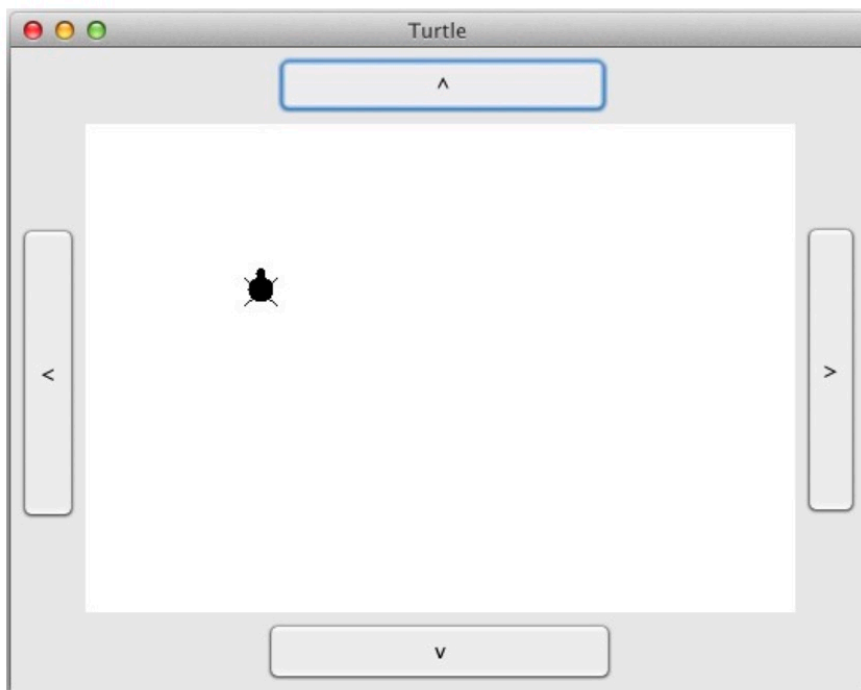
- Le déplacement se fait à l'aide de quatre boutons situés à l'extérieur de la surface de dessin.
- La vitesse de déplacement de la tortue est de 10 pixel par clic.
- La position initiale de la tortue est (100,100).
- Le dessin de la tortue se fait approximativement comme sur le dessin ci-dessous.
- La tortue 'regarde' toujours vers le haut, même si elle se déplace dans une autre direction. La tortue peut se balader aussi en-dehors de la surface visible.

Développer le programme en respectant le modèle MVC et en employant 3 classes.

Dans cette première version, les opérations de dessin sont effectuées dans **paintComponent** de **DrawPanel**.

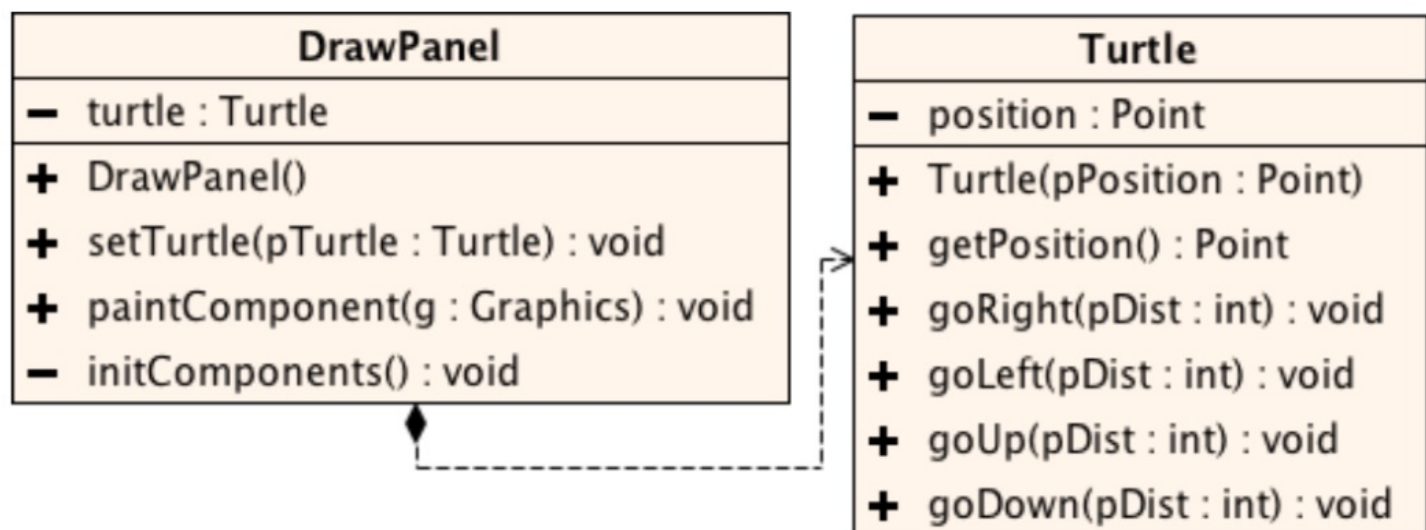
Principe de base :

Une nouvelle tortue est créée dans **MainFrame**. Elle est passée tout de suite à **DrawPanel** (à l'aide de **setTurtle**). Ceci est primordial, puisque **MainFrame** et **DrawPanel** doivent travailler avec LA MÊME tortue.



Largeur: 21 pixels
Hauteur totale : 24 pixels
Diamètre carapace: 15 pixels

Pour vous aider dans la conception du programme, voici le schéma UML de la classe **Turtle** représentant la tortue et de la classe **DrawPanel** pour représenter la tortue graphiquement sur un panneau. L'attribut **position** correspond au centre de la carapace de la tortue (représenté par le petit cercle dans le schéma).



```
1
2 import java.awt.Color;
3 import java.awt.Graphics;
4
5 /*
6  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.
7  * txt to change this license
8  * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JPanel.java to
9  * edit this template
10 */
11
12 /**
13  * @author luxformel
14  */
15 public class DrawPanel extends javax.swing.JPanel {
16     //Déclarer un attribut de type Turtle
17     //DrawPanel est responsable de dessiner t
18     private Turtle t;
19
20     //Setter pour l'attribut
21     public void setTurtle(Turtle t) {
22         this.t = t;
23     }
24
25     /**
26     * Creates new form DrawPanel
27     */
28     public DrawPanel() {
29         initComponents();
30     }
31
32     /**
33     * This method is called from within the constructor to initialize the
34     form.
35     * WARNING: Do NOT modify this code. The content of this method is
36     always
37     * regenerated by the Form Editor.
38     */
39     @SuppressWarnings("unchecked")
40
41
42
43
44
45
46
47
48
49
50
51
52
53     @Override
54     protected void paintComponent(Graphics g) {
55         super.paintComponent(g); // Generated from nbfs:
56         //nbhost/SystemFileSystem/Templates/Classes/Code/OverriddenMethodBody
57         //L'arrière plan du canvas est dessiné en blanc
58         g.setColor(Color.white);
59         g.fillRect(0, 0, getWidth(), getHeight());
60         //Si t a été instancié //Si t existe //Si l'objet t existe
61         if (t != null)
```

```
61          //Alors (et seulement dans ce cas il est dessiné)
62          t.draw(g);
63      }
64
65
66      // Variables declaration – do not modify//GEN-BEGIN:variables
67      // End of variables declaration//GEN-END:variables
68  }
69
```

```
1
2  import java.awt.Point;
3
4  /*
4   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.
5   txt to change this license
5   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to
6   edit this template
7   */
8
9  /**
10   *
11   * @author luxformel
12   */
13  public class MainFrame extends javax.swing.JFrame {
14      private Turtle t;
15
16      /**
17       * Creates new form MainFrame
18       */
19      public MainFrame() {
20          initComponents();
21          //La tortue est placée au milieu du drawPanel
22          int x = drawPanel.getWidth() / 2;
23          int y = drawPanel.getHeight() / 2;
24          Point position = new Point (x, y);
25
26          //Création de la tortue
27          t = new Turtle(position);
28
29          //Lier la tortue avec le drawPanel
30          drawPanel.setTurtle(t);
31          drawPanel.repaint();
32          positionLabel.setText("X:" + position.x + "; Y:" + position.y);
33      }
34
35      /**
35       * This method is called from within the constructor to initialize the
36       form.
36       * WARNING: Do NOT modify this code. The content of this method is
37       always
38       * regenerated by the Form Editor.
39       */
42      @SuppressWarnings("unchecked")
50
52
63
70
77
84
91
93
```

```
95
147
150
151
151 private void rightButtonActionPerformed(java.awt.event.ActionEvent evt)
152 {
153     t.goRight(10, drawPanel.getWidth());
154     drawPanel.repaint();
155     Point position = t.getPosition();
156     positionLabel.setText("X:" + position.x + "; Y:" + position.y);
157 }
158
158 private void leftButtonActionPerformed(java.awt.event.ActionEvent evt)
159 {
160     t.goLeft(10);
161     drawPanel.repaint();
162     Point position = t.getPosition();
163     positionLabel.setText("X:" + position.x + "; Y:" + position.y);
164 }
165
165 private void downButtonActionPerformed(java.awt.event.ActionEvent evt)
166 {
167     t.goDown(10, drawPanel.getHeight());
168     drawPanel.repaint();
169     Point position = t.getPosition();
170     positionLabel.setText("X:" + position.x + "; Y:" + position.y);
171 }
172
173 private void upButtonActionPerformed(java.awt.event.ActionEvent evt) {
174     t.goUp(10);
175     drawPanel.repaint();
176     Point position = t.getPosition();
177     positionLabel.setText("X:" + position.x + "; Y:" + position.y);
178 }
179
180 /**
181  * @param args the command line arguments
182  */
183 public static void main(String args[]) {
205     /* Set the Nimbus look and feel */
206
207     /* Create and display the form */
208     java.awt.EventQueue.invokeLater(new Runnable() {
209         public void run() {
210             new MainFrame().setVisible(true);
211         }
212     });
213 }
214
215 // Variables declaration - do not modify//GEN-BEGIN:variables
216 private javax.swing.JButton downButton;
217 private DrawPanel drawPanel;
```

```
218     private javax.swing.JLabel jLabel1;  
219     private javax.swing.JButton leftButton;  
220     private javax.swing.JLabel positionLabel;  
221     private javax.swing.JButton rightButton;  
222     private javax.swing.JButton upButton;  
223     // End of variables declaration//GEN-END:variables  
224 }  
225
```

```
1
2 import java.awt.Color;
3 import java.awt.Graphics;
4 import java.awt.Point;
5
6
7 /*
7 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.
8 txt to change this license
8 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
9 this template
10 */
11
12 /**
13 *
14 * @author luxformel
15 */
16 public class Turtle {
17     private Point position;
18
19     public Turtle(Point position) {
20         this.position = position;
21     }
22
23     public Point getPosition() {
24         return position;
25     }
26
27     public void goRight(int pDist, int width){
28         if (position.x + 10 + pDist <= width)
29             position.x = position.x + pDist;
30         else
31             position.x = width - 10;
32     }
33
34     public void goLeft(int pDist){
35         //Explication similiaires à goRight
36         //Ici la tortue se déplace vers la gauche
37         if (position.x - 10 - pDist >= 0)
38             position.x = position.x - pDist;
39         else
40             position.x = 10;
41     }
42
43     public void goUp(int pDist){
44         //Explication similiaires à goRight
45         //Ici la tortue se déplace vers la haut
46         if (position.y - 13 - pDist >= 0)
47             position.y = position.y - pDist;
48         else
49             position.y = 13;
50     }
51 }
```



```
51
52     public void goDown(int pDist, int height){
53         //Explication similiaires à goRight
54         //Ici la tortue se déplace vers le bas
55         if (position.y + 10 + pDist <= height)
56             position.y = position.y + pDist;
57         else
58             position.y = height - 10;
59     }
60
61     public void draw(Graphics g){
62         g.setColor(Color.black);
63         g.fillOval(position.x - 7, position.y - 7, 15, 15);
64         g.fillOval(position.x - 2, position.y - 13, 5, 6);
64         g.drawLine(position.x - 10, position.y - 7, position.x + 10,
65 position.y + 10);
65         g.drawLine(position.x - 10, position.y + 10, position.x + 10,
66 position.y - 7);
67     }
68
69 }
70
```