```java
public class Rectangle
{
     private int x;
     private int y;
     private int width;
     private int height;

     public Rectangle(int x, int y, int width, int height)
     {
          this.x = x;
          this.y = y;
          this.width = width;
          this.height = height;
     }

     public int getHeight()
     {
          return height;
     }

     public int getWidth()
     {
          return width;
     }

     public int getY()
     {
          return y;
     }

     public int getX()
     {
          return x;
     }

     public void setLocation(int x, int y)
     {
          this.x = x;
          this.y = y;
     }

     public boolean contains(int x, int y)
     {
          return x >= this.x && x <= this.x + width
              && y >= this.y && y <= this.y + height;
     }

     public boolean contains(Rectangle o)
     {
          return contains(o.x, o.y)
              && contains(o.x + o.width - 1, o.y + o.height - 1);
     }

     public boolean intersects(Rectangle o)
     {
          return !((x + width < o.x) || (x > o.x + o.width)
                || (y + height < o.y) || (y > o.y + o.height));
     }
}
```

```java
import java.awt.Color;
import java.awt.Graphics;
public class Crosshair extends Rectangle
{
     public Crosshair(int x, int y)
     {
          super(x, y, 1, 1);
     }

     public void draw(Graphics g)
     {
          int x = getX();
          int y = getY();

          g.setColor(Color.BLACK);
          g.drawLine(x, y - 40, x, y + 40);
          g.drawLine(x - 40, y, x + 40, y);
          g.drawOval(x - 10, y - 10, 21, 21);
          g.drawOval(x - 25, y - 25, 51, 51);
     }
}
```

```java
import java.awt.Color;
import java.awt.Graphics;
public class Turtle extends Rectangle
{
     private Color color;

     public Turtle(int x, int y, Color color)
     {
          super(x, y, 21, 25);
          this.color = color;
     }

     public void draw(Graphics g)
     {
          int x = getX();
          int y = getY();

          g.setColor(color);
          g.fillOval(x + 2, y + 6, 16, 16);
          g.fillOval(x + 7, y + 0, 6, 8);
          g.drawLine(x + 0, y + 7, x + 20, y + 24);
          g.drawLine(x + 20, y + 7, x + 0, y + 24);
     }

     public void goRight(int dist)
     {
          setLocation(getX() + dist, getY());
     }

     public void goLeft(int dist)
     {
          setLocation(getX() - dist, getY());
     }

     public void goUp(int dist)
     {
          setLocation(getX(), getY() - dist);
     }

     public void goDown(int dist)
     {
          setLocation(getX(), getY() + dist);
     }

     public void makeRandomMove()
     {
          int dx = (int) (Math.random() * 21) - 10;
          int dy = (int) (Math.random() * 21) - 10;

          setLocation(getX() + dx, getY() + dy);
     }

     public void moveBy(int dx, int dy)
     {
          setLocation(getX() + dx, getY() + dy);
     }
}
```

```java
import java.awt.Color;
import java.awt.Graphics;
public class TurtleHunting
{
     private Turtle turtle = null;
     private Crosshair crosshair = null;
     private boolean gameOver;
     private int bulletsLeft;
     private int turtlesLeft;
     private int width;
     private int height;

     public TurtleHunting(int width, int height)
     {
          this.width = width;
          this.height = height;
          crosshair = new Crosshair(width / 2, height / 2);
          turtle = new Turtle(0, 0, Color.BLACK);

          positionTurtleRandomly();
          bulletsLeft = 10;
          turtlesLeft = 5;
          gameOver = false;
     }

     public void fire()
     {
          if (gameOver)
               return;

          bulletsLeft--;
          if (turtle.intersects(crosshair))
          {
               turtlesLeft--;
               positionTurtleRandomly();
          }

          if (bulletsLeft <= 0 || turtlesLeft <= 0)
               gameOver = true;
     }

     private void positionTurtleRandomly()
     {
          int x = (int) (Math.random() * (width - turtle.getWidth()));
          int y = (int) (Math.random() * (height - turtle.getHeight()));
          turtle.setLocation(x, y);
     }

     public void makeTurtleMoveRandomly()
     {
          turtle.makeRandomMove();
     }

     public void moveCrosshairBy(int dx, int dy)
     {
          turtle.moveBy(dx, dy);
     }

     public void moveCrosshairUp()
     {
          turtle.goDown(10);
     }

     public void moveCrosshairDown()
     {
          turtle.goUp(10);
     }

     public void moveCrosshairRight()
     {
          turtle.goLeft(10);
     }

     public void moveCrosshairLeft()
     {
          turtle.goRight(10);
     }
```

// Suite page suivante .....

```
    public void draw(Graphics g)
    {
        g.setColor(Color.BLACK);
        g.fillRect(0, 0, width, height);
        g.setColor(Color.WHITE);
        int side = Math.min(width, height);
        g.fillOval((width - side) / 2, (height - side) / 2, side, side);

        g.setColor(Color.YELLOW);
        g.drawString("Bullets left: " + bulletsLeft, 5, 20);

        Turtle turtleIcon = new Turtle(0, 0, Color.YELLOW);
        turtleIcon.setLocation(width - 50, 2);
        turtleIcon.draw(g);

        g.drawString(String.valueOf(turtlesLeft), width - 20, 20);

        if (!gameOver)
            turtle.draw(g);

        crosshair.setLocation(width / 2, height / 2);
        crosshair.draw(g);

        if (gameOver)
        {
            g.setColor(Color.red);
            if (turtlesLeft > 0)
                g.drawString("You lose this game!", width / 2 - 55, height / 4);
            else
                g.drawString("You win this game!", width / 2 - 55, height / 4);
        }
    }
}
```

```java
import java.awt.Color;
import java.awt.Graphics;
public class DrawPanel extends javax.swing.JPanel
{
     private TurtleHunting turtleHunting = null;

     public DrawPanel()
     {
          initComponents();
     }

     public void setTurtleHunting(TurtleHunting turtleHunting)
     {
          this.turtleHunting = turtleHunting;
     }

     public void paintComponent(Graphics g)
     {
          g.setColor(Color.WHITE);
          g.fillRect(0, 0, getWidth(), getHeight());

          if (turtleHunting != null)
               turtleHunting.draw(g);
     }
// Skipped: ... initComponents { ... }
// Skipped: ... graphic attributes
}
```

```java
import java.awt.Point;
import java.awt.event.MouseEvent;
import javax.swing.Timer;
public class MainFrame extends javax.swing.JFrame
{
     private TurtleHunting turtleHunting;
     private Timer timer = null;
     private Point lastPosition;

     public MainFrame()
     {
          initComponents();
          turtleHunting = new TurtleHunting(drawPanel.getWidth(), drawPanel.getHeight());
          drawPanel.setTurtleHunting(turtleHunting);
          repaint();
          stepButton.setVisible(false);
          timer = new Timer(200, stepButton.getActionListeners()[0]);
          timer.start();
     }
// Skipped: ... initComponents { ... }
     private void upButtonActionPerformed(java.awt.event.ActionEvent evt)
     {//GEN-FIRST:event_upButtonActionPerformed
          turtleHunting.moveCrosshairUp();
          repaint();
     }//GEN-LAST:event_upButtonActionPerformed

     private void rightButtonActionPerformed(java.awt.event.ActionEvent evt)
     {//GEN-FIRST:event_rightButtonActionPerformed
          turtleHunting.moveCrosshairRight();
          repaint();
     }//GEN-LAST:event_rightButtonActionPerformed

     private void downButtonActionPerformed(java.awt.event.ActionEvent evt)
     {//GEN-FIRST:event_downButtonActionPerformed
          turtleHunting.moveCrosshairDown();
          repaint();
     }//GEN-LAST:event_downButtonActionPerformed

     private void leftButtonActionPerformed(java.awt.event.ActionEvent evt)
     {//GEN-FIRST:event_leftButtonActionPerformed
          turtleHunting.moveCrosshairLeft();
          repaint();
     }//GEN-LAST:event_leftButtonActionPerformed

     private void fireButtonActionPerformed(java.awt.event.ActionEvent evt)
     {//GEN-FIRST:event_fireButtonActionPerformed
          turtleHunting.fire();
          repaint();
     }//GEN-LAST:event_fireButtonActionPerformed

     private void stepButtonActionPerformed(java.awt.event.ActionEvent evt)
     {//GEN-FIRST:event_stepButtonActionPerformed
          turtleHunting.makeTurtleMoveRandomly();
          repaint();
     }//GEN-LAST:event_stepButtonActionPerformed

     private void drawPanelMouseDragged(java.awt.event.MouseEvent evt)
     {//GEN-FIRST:event_drawPanelMouseDragged
          if (lastPosition == null)
               lastPosition = evt.getPoint();

          int dx = lastPosition.x - evt.getX();
          int dy = lastPosition.y - evt.getY();
          turtleHunting.moveCrosshairBy(dx, dy);
          lastPosition = evt.getPoint();
          repaint();
     }//GEN-LAST:event_drawPanelMouseDragged

     private void drawPanelMousePressed(java.awt.event.MouseEvent evt)
     {//GEN-FIRST:event_drawPanelMousePressed
          if (evt.getButton() == MouseEvent.BUTTON3)
          {
               turtleHunting.fire();
               repaint();
          }
          else
          {
               lastPosition = evt.getPoint();
          }
     }//GEN-LAST:event_drawPanelMousePressed

     /**
      * @param args the command line arguments
      */
     public static void main(String args[])
     {
          java.awt.EventQueue.invokeLater(new Runnable()
          {
               public void run()
               {
                    new MainFrame().setVisible(true);
               }
          });
     }
// Skipped: ... graphic attributes
}
```