CSCI-UA 9102 DATA STRUCTURES
Assignment 1

Augustin Cosse

May 2025

Given date: May 22
Due date: June 1
Total: 10pts

In this first assignment, we will design two simple graphical user interfaces to test your understanding of control flow statements in java.

Question 1 (5pts)

Consider the class Target given in the folder 'Assignment1'.

In this first question, you will be asked to draw the white and black rings around a target as shown in Fig. 1 below. To do this, you can find the files `'Target.java'` and `'TargetPanel.java'` in the Assignment 1 folder. The second file contains the definition of the class `'TargetPanel'` which is used in the main method of the first file. You should complete the file `'TargetPanel.java'` so that it draws alternating white and black rings around the center of the target. To do this:

- Use a for loop whose body will be executed `'NUM_RINGS'` times.
- Each time the body of the for loop gets executed, you should check and set the color using the method `'page.getColor()'` as well as the method `'page.setColor(Color.xxx)'` where `'xxx'` represents a color. The color should change each time (hint: use an if-else statement)
- The rings will in fact be represented using superimposed disks of decreasing diameters. Start with a diameter = `2*MAX_WIDTH` and then decrease this diameter by `(2*RING_WIDTH)` each time you draw a new ring.

- To draw each disk we will use the function `'page.fillOval(x, y, diameter, diameter)'` whose specification is given in Fig. 2 below. The function will draw an oval bounded by the specified rectangle with the current color (as set using `'page.setColor()'`).

Question 2 (5pts)

In this second question, you will be asked to use a `'switch case'`statement to handle a GUI with a series of buttons. You should use the two files `'ButtonDigicode.java'` and `'PanelwithButtons.java'` that you can find in the assignment folder. The first file defines the class `'ButtonDigicode'` which builds a panel with two buttons, `button0` and `button1`. Those buttons are associated to the numbers 0 and 1 respectively. When pressed, they should display these numbers.

- Start by increasing the number of buttons to include the 10 digits $0, \ldots, 9$.
- Replace the `'if else'` statement by a `switch case` in order to display each of the $0, \ldots, 9$ digits of your buttons when those are pressed.

Bonus (3pts)

To optimize the display of your buttons, Java `Swing` and `awt` packages provide several layout managers including `BoxLayout`. Using `BoxLayout` as well as `setAlignmentX` on your buttons, try to turn your GUI panel into a proper digicode.

Figure 1: What your target should look like

fillOval

```
public abstract void fillOval(int x,
            int y,
            int width,
            int height)
```

Fills an oval bounded by the specified rectangle with the current color.

Parameters:

`x` - the *x* coordinate of the upper left corner of the oval to be filled.

`y` - the *y* coordinate of the upper left corner of the oval to be filled.

`width` - the width of the oval to be filled.

`height` - the height of the oval to be filled.

See Also:

`drawOval(int, int, int, int)`

Figure 2: Specification of the `'fillOval'` method

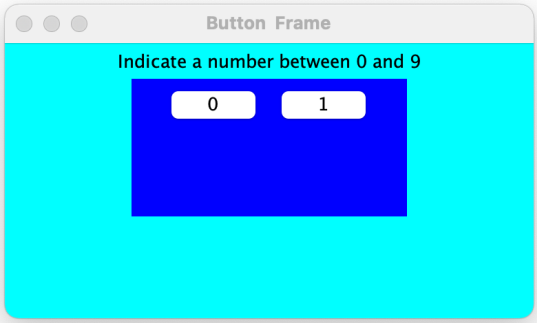

Figure 3: The button Panel