Grafični vmesniki v programskem jeziku Java

Splošnonamenski programski jezik **Java** je bil razvit kot **varen jezik za poljubno napravo**. Sintaksa zahteva daljše programe kot v programskem jeziku Python, ki pa so navadno veliko hitrejši. Jezik se prevaja, kar pomeni, da program `Demo.java` v ukazni vrstici najprej prevedemo z `javac Demo.java`, kar ustvari vmesno datoteko `Demo.class`, katero nato izvedemo z `java Demo`.

...

Animacija likov in grafičnih oblik

Geometrijske **like in** grafične **oblike lahko animiramo** tako, da pri izrisu upoštevamo stanje izbranih (zunanjih) spremenljivk, ki jih na primer posodabljamo v določenih časovnih intervalih.

Spodnji program doda izris slike nogometne žoge, katere središče je predstavljeno z javno statično spremenljivko `ball` tipa `java.awt.Point` definirano v razredu `Visuals2`.

Java

```java
public class Visuals2 {

    public static Point ball = new Point();

    public static void main(String[] args) {
        // ...
        ball = new Point(panel.getWidth() / 2, panel.getHeight() / 2);
        // ...
    }

}

class Panel2 extends JPanel {

    @Override
    public void paint(Graphics g) {
        // ...
        try { // shrani sliko v spremenljivko ali konstanto!
            graphics.drawImage(ImageIO.read(new File("images", "football.png")), (int)Visua
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}
```

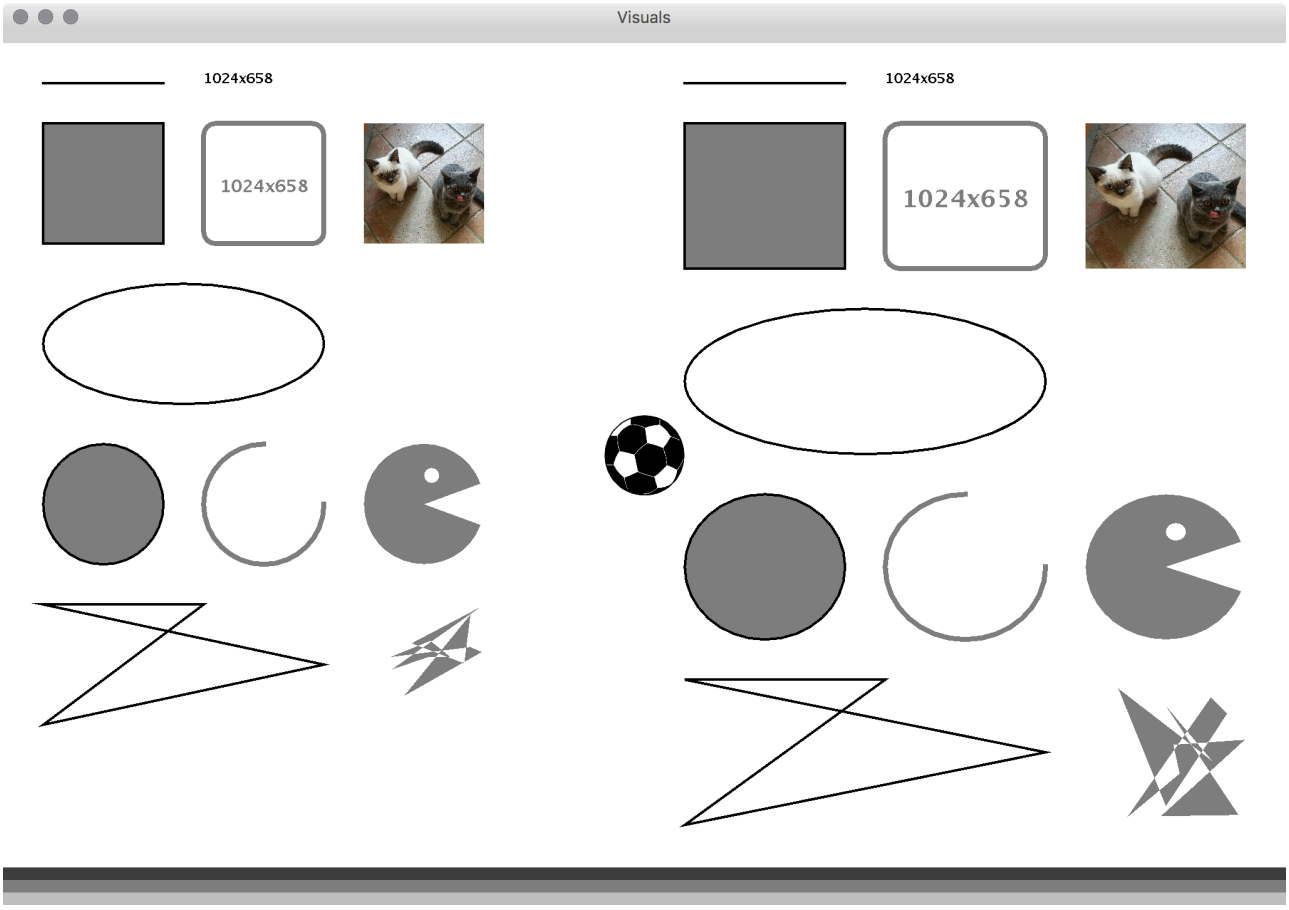
Spodnji program doda še animacijo nogometne žoge, ki se odbija od stranic panela po katerem rišemo.

```java
public class Visuals2 {

    public static void main(String[] args) {
        // ...
        Point direction = new Point(8, -8);
        while (true) {
            ball.setLocation(ball.getX() + direction.getX(), ball.getY() + direction.getY()
            if (ball.getX() <= 32) {
                ball.setLocation(32, ball.getY());
                direction.setLocation(-direction.getX(), direction.getY());
            }
            else if (ball.getX() >= panel.getWidth() - 32) {
                ball.setLocation(panel.getWidth() - 32, ball.getY());
                direction.setLocation(-direction.getX(), direction.getY());
            }
            if (ball.getY() <= 32) {
                ball.setLocation(ball.getX(), 32);
                direction.setLocation(direction.getX(), -direction.getY());
            }
            else if (ball.getY() >= panel.getHeight() - 32) {
                ball.setLocation(ball.getX(), panel.getHeight() - 32);
                direction.setLocation(direction.getX(), -direction.getY());
            }
            frame.repaint(); // ponoven izris okna
            try {
                Thread.sleep(50); // počakaj 50 ms
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }

}
```

Primer izgleda grafičnega vmesnika je prikazan spodaj.



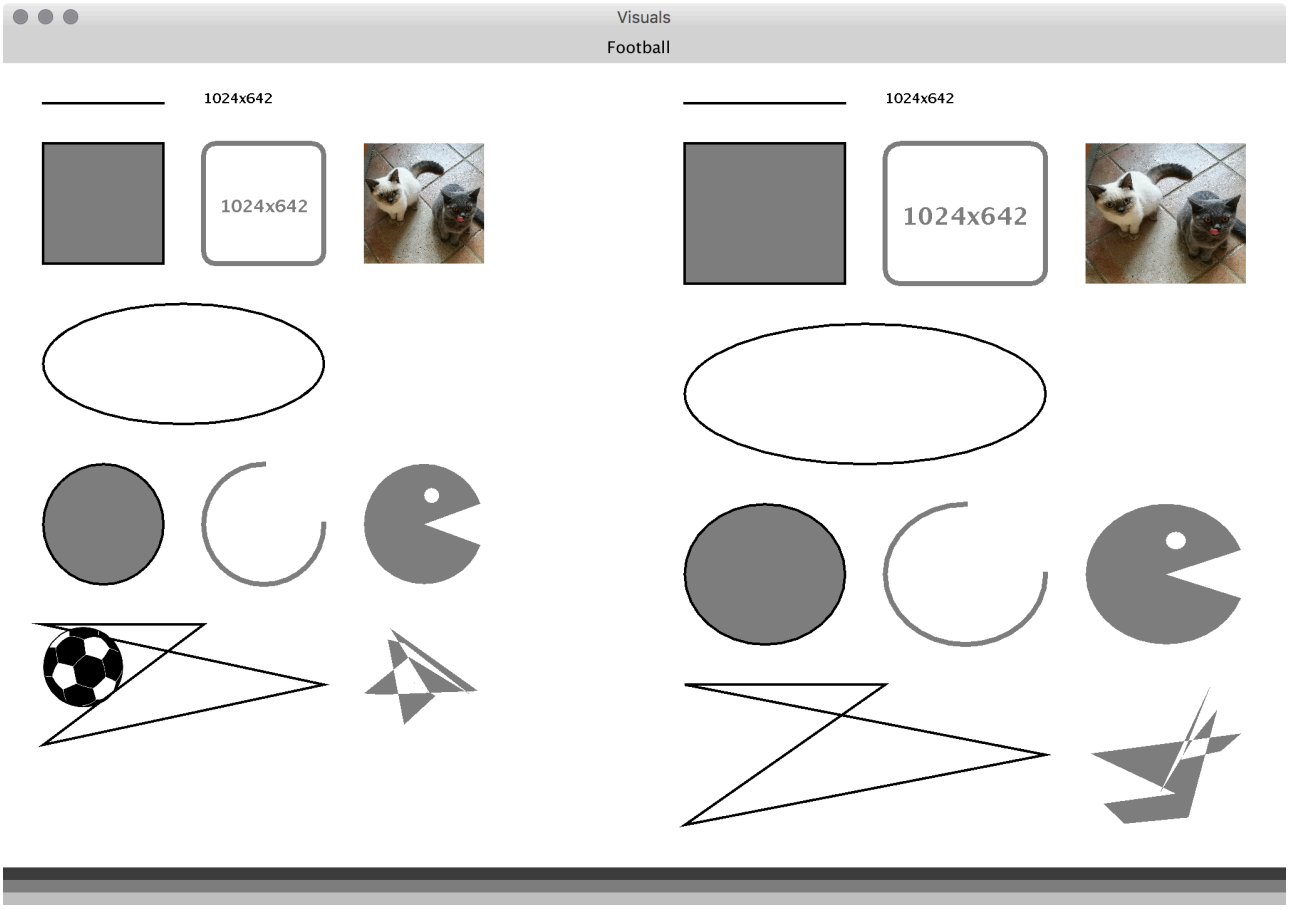
Grafični elementi in nadzor programa

Panelom razreda `JPanel` lahko **dodamo različne grafične elemente** z uporabo metode `add(Component component)`. Med **osnovne grafične elemente** štejemo labele razreda `JLabel`, gumbe razredov `JButton` in `JToggleButton`, drsnike razreda `JSlider`, vnosna polja razreda `JTextField`, izbirnike razredov `JCheckBox` in `JComboBox` itd. v paketu `javax.swing`.

Spodnji program **doda labelo** razreda `JLabel` severnemu panelu programskega okna.

```java
north.add(new JLabel("Football  "));
```

Primer izgleda grafičnega vmesnika je prikazan spodaj.

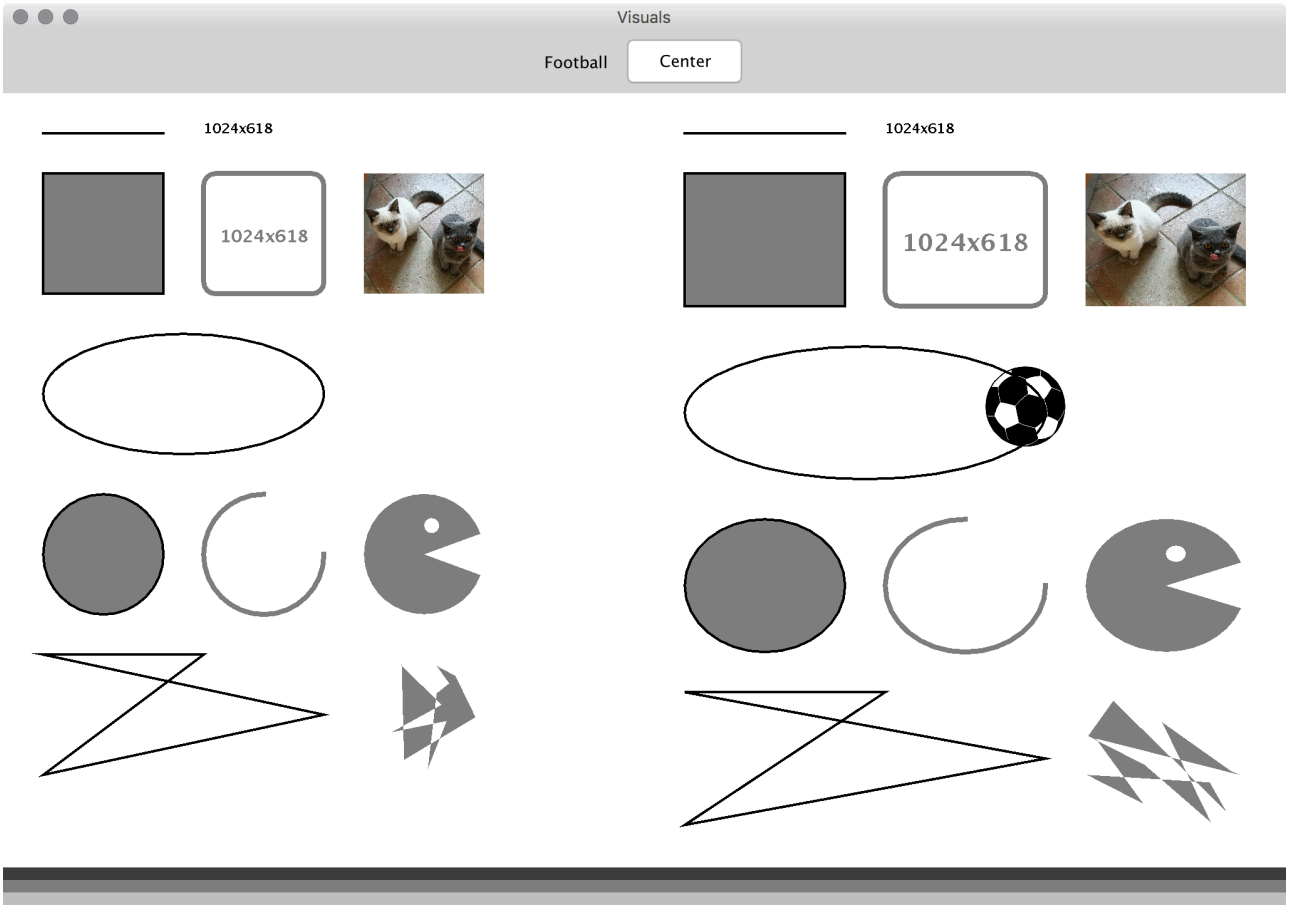


Grafični elementi **preko dogodkov** (npr. pritisk gumba, sprememba drsnika) **omogočajo nadzor** nad delovanjem programa oziroma grafičnega vmesnika. Na same dogodke se odzovemo tako, da grafičnemu elementu **dodamo poslušalca**, ki implementira vmesnik `java.util.EventListener` in s tem vse potrebne metode za odziv na dogodke grafičnega elementa.

Spodnji program **doda gumb** razreda `JButton` severnemu panelu programskega okna. Gumbu **dodamo poslušalca**, ki implementira vmesnik `java.awt.event.ActionListener` kot anonimni razred in ob kliku na gumb nogometno žogo prestavi v središče osrednjega panela.

```java
JButton button = new JButton("Center");
button.setPreferredSize(new Dimension(96, 40));
button.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        ball = new Point(panel.getWidth() / 2, panel.getHeight() / 2);
    }
});
north.add(button);
```

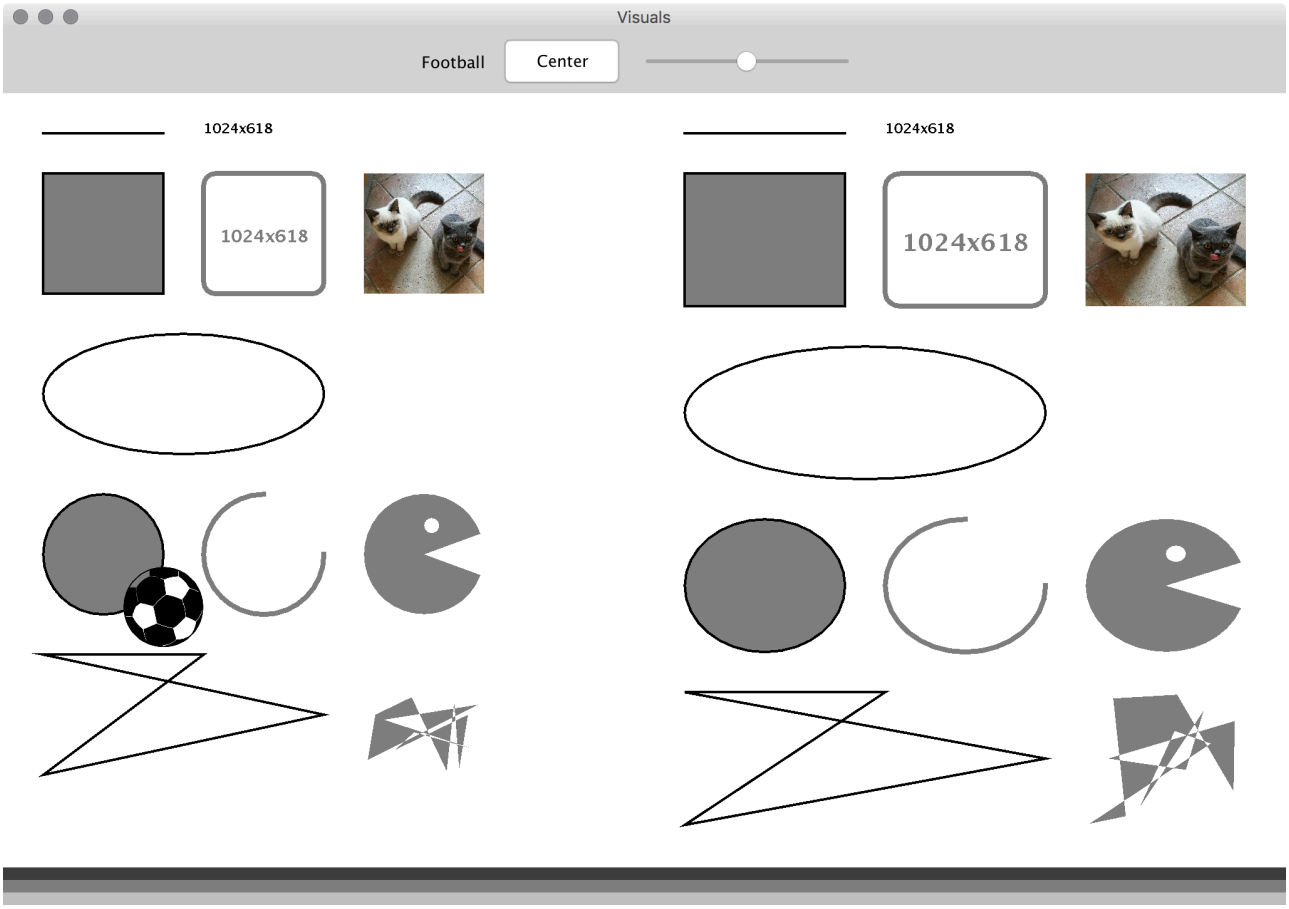
Primer izgleda grafičnega vmesnika je prikazan spodaj.



Spodnji program **doda drsnik** razreda `JSlider` severnemu panelu programskega okna, ki določa hitrost posodabljanja središča nogometne žoge. Drsniku **dodamo poslušalca**, ki implementira vmesnik `java.awt.event.ChangeListener` in ob spremembi zahteva ponoven izris osrednjega panela.

```java
JSlider slider = new JSlider(1, 100, 50);
slider.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {
        panel.repaint();
    }
});
north.add(slider);
// ...
while (true) {
    // ...
    try {
        Thread.sleep(slider.getValue());
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
```

Primer izgleda grafičnega vmesnika je prikazan spodaj.

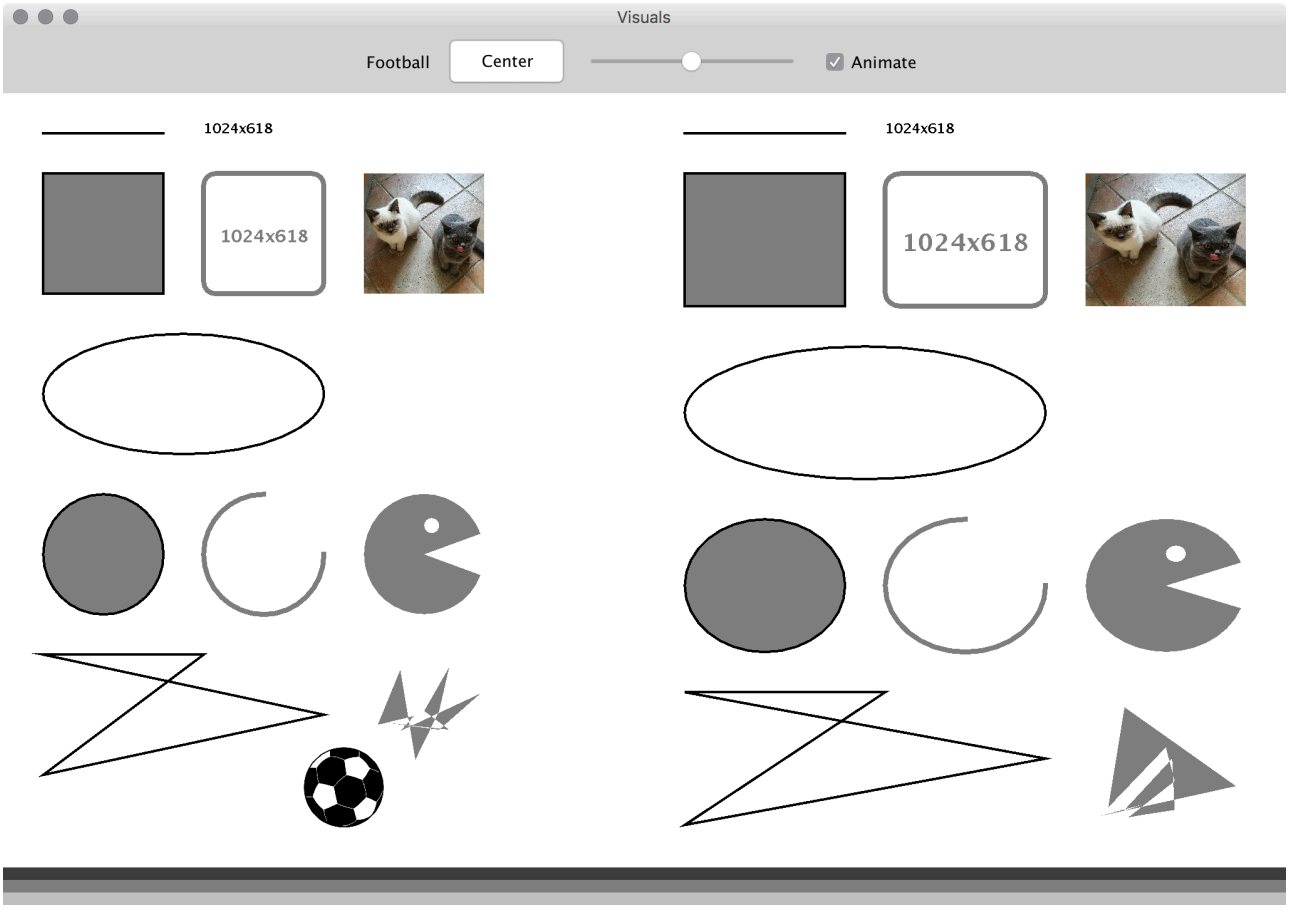


Spodnji program **doda izbirnik** razreda `JCheckBox` severnemu panelu programskega okna, ki nadzira posodabljanje središča nogometne žoge. Izbirniku **dodamo poslušalca**, ki implementira vmesnik `java.awt.event.ActionListener` in ob spremembi zahteva ponoven izris osrednjega panela.

Java

```
JCheckBox checkbox = new JCheckBox("Animate", true);
checkbox.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        panel.repaint();
    }
});
north.add(checkbox);
// ...
while (true) {
    if (checkbox.isSelected()) {
        // ...
    }
    // ...
}
```

Primer izgleda grafičnega vmesnika je prikazan spodaj.

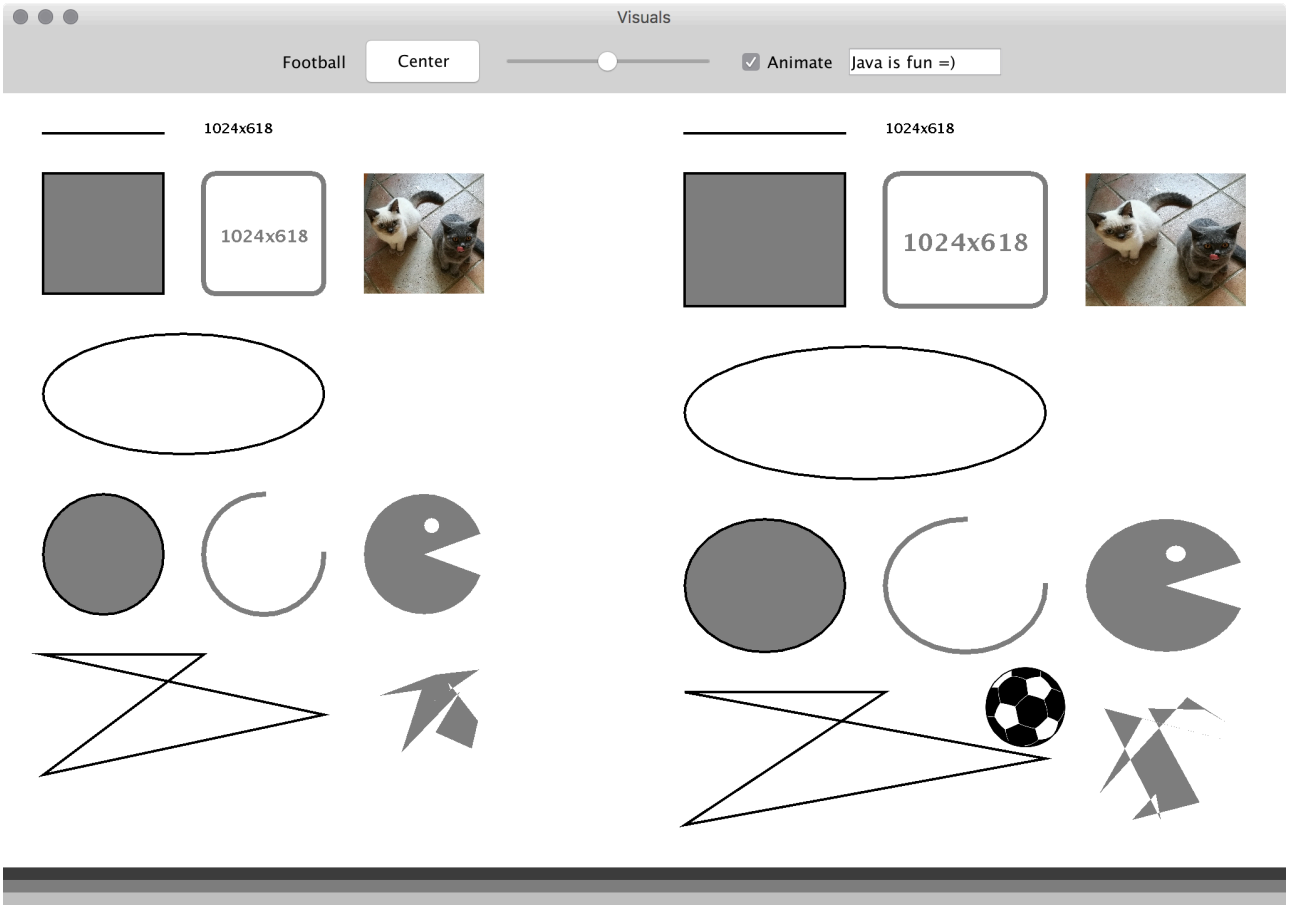


Spodnji program **doda vnosno polje** razreda `JTextField` severnemu panelu programskega okna, ki ob pritisku na vnašalko izbriše vsebino vnosnega polja. Vnosnemu polju **dodamo poslušalca**, ki implementira vmesnik `java.awt.event.KeyListener` in se izvede ob pritisku na poljubno tipko.

Java

```
JTextField field = new JTextField();
field.setPreferredSize(new Dimension(128, 24));
field.addKeyListener(new KeyListener() {
    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_ENTER)
            field.setText("");
    }
    @Override
    public void keyReleased(KeyEvent e) { }
    @Override
    public void keyTyped(KeyEvent e) { }
});
north.add(field);
```

Primer izgleda grafičnega vmesnika je prikazan spodaj.


Visuals
Football
Center
Animate
Java is fun =)
1024x618
1024x618
1024x618
1024x618