

# Dokumentáció

---

## Tartalomjegyzék

- 1. Rotating Cube
  - 1. Bevezetés
  - 2. Kód Működése
  - 3. Felhasznált Könyvtárak
  - 4. Kód Magyarázat
    - 1. Inicializálás
    - 2. Kocka Adatok
    - 3. Forgás Változók
    - 4. Fő Ciklus
  - 5. Felhasználói Interakció
- 2. BM7SGF Projektkompgraf
  - 1. Bevezetés
  - 2. Kód Működése
  - 3. Felhasznált Könyvtárak
  - 4. Kód Magyarázat
    - 1. Inicializálás és Kép Betöltése
    - 2. Szív Alakú Maszk
    - 3. Kép Kirajzolása
  - 5. Felhasználói Interakció

## 1. Rotating Cube

### 1.1 Bevezetés

A **Rotating Cube** egy Pygame alkalmazás, amely egy forgó kockát jelenít meg. A felhasználó a nyílbillentyűkkel forgathatja a kockát a különböző tengelyek mentén.

### 1.2 Kód Működése

Ez az alkalmazás a Pygame könyvtár segítségével rajzolja ki a kockát, és a felhasználói bemenetek alapján forgatja azt.

### 1.3 Felhasznált Könyvtárak

- `pygame`
- `math`

### 1.4 Kód Magyarázat

#### 1.4.1 Inicializálás

A Pygame könyvtár inicializálása és a képernyő beállítása történik:

```
pygame.init()
WIDTH, HEIGHT = 800, 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Rotating Cube")
```

### 1.4.2 Kocka Adatok

A kocka csúcsainak és éleinek definiálása:

```
vertices = [(-100, -100, -100), (100, -100, -100),
            (100, 100, -100), (-100, 100, -100),
            (-100, -100, 100), (100, -100, 100),
            (100, 100, 100), (-100, 100, 100)]

edges = [(0, 1), (1, 2), (2, 3), (3, 0),
        (4, 5), (5, 6), (6, 7), (7, 4),
        (0, 4), (1, 5), (2, 6), (3, 7)]
```

### 1.4.3 Forgás Változók

A forgás szögei és az óra inicializálása:

```
angle_x = 0
angle_y = 0
angle_z = 0
clock = pygame.time.Clock()
```

### 1.4.4 Fő Ciklus

A fő ciklus, amely kezeli az eseményeket és kirajzolja a kockát:

```
running = True
while running:
    screen.fill((0, 0, 0))

    for event in pygame.event.get():
        if event.type == QUIT:
            running = False
        elif event.type == KEYDOWN:
            if event.key == K_ESCAPE:
                running = False
            elif event.key == K_LEFT:
                angle_y += 0.1
```

```
        elif event.key == K_RIGHT:
            angle_y -= 0.1
        elif event.key == K_UP:
            angle_x += 0.1
        elif event.key == K_DOWN:
            angle_x -= 0.1

    for edge in edges:
        points = []
        for vertex in edge:
            x, y, z = vertices[vertex]
            x = x * math.cos(angle_y) - z * math.sin(angle_y)
            z = x * math.sin(angle_y) + z * math.cos(angle_y)
            y = y * math.cos(angle_x) - z * math.sin(angle_x)
            z = y * math.sin(angle_x) + z * math.cos(angle_x)
            y = y * math.cos(angle_z) - x * math.sin(angle_z)
            x = y * math.sin(angle_z) + x * math.cos(angle_z)
            x += WIDTH / 2
            y += HEIGHT / 2
            points += [(x, y)]
        pygame.draw.line(screen, (255, 255, 255), points[0], points[1], 2)

    pygame.display.flip()
    clock.tick(30)

pygame.quit()
```

## 1.5 Felhasználói Interakció

A felhasználó a nyílbillentyűkkel forgathatja a kockát különböző tengelyek mentén.

# 2. BM7SGF Projektkompgraf

## 2.1 Bevezetés

A **BM7SGF\_projektkompgraf** egy Java alkalmazás, amely egy szív alakú maszkot rajzol egy képre, és kirajzolja a Tokaj-Hegyalja Egyetem nevét.

## 2.2 Kód Működése

Ez az alkalmazás a Java Swing és Graphics2D könyvtárakat használja a kép manipulálásához és megjelenítéséhez.

## 2.3 Felhasznált Könyvtárak

- `javax.swing`
- `java.awt`
- `java.awt.geom.Area`
- `java.awt.image.BufferedImage`
- `javax.imageio.ImageIO`
- `java.io.File`

- `java.io.IOException`

## 2.4 Kód Magyarázat

### 2.4.1 Inicializálás és Kép Betöltése

A kép betöltése és hiba esetén egy alapértelmezett kép létrehozása:

```
public BM7SGF_projektkompgraf() {
    try {
        image = ImageIO.read(new
File("C:\\Users\\kvnpa\\Documents\\NetBeansProjects\\BM7SGF_beadandokompgraf\\src\\
\\bm7sgf_beadandokompgraf\\the.jpg"));
    } catch (IOException e) {
        e.printStackTrace();
        image = new BufferedImage(600, 600, BufferedImage.TYPE_INT_RGB);
        Graphics g = image.getGraphics();
        g.setColor(Color.RED);
        g.fillRect(0, 0, 600, 600);
        g.setColor(Color.BLACK);
        g.drawString("Kép nem található", 250, 300);
        g.dispose();
    }
}
```

### 2.4.2 Szív Alakú Maszk

A szív alakú maszk létrehozása és alkalmazása a képre:

```
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    Graphics2D g2d = (Graphics2D) g;
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

    HeartShape heart = new HeartShape(100, 100, 600);
    g2d.setColor(Color.RED);
    g2d.fill(heart);

    Area heartArea = new Area(heart);
    heartArea.intersect(new Area(new Rectangle(100, 100, 600, 600)));
    g2d.setClip(heartArea);
    g2d.drawImage(image, 100, 100, 600, 600, this);
    g2d.setClip(null);

    g2d.setColor(Color.BLACK);
    g2d.setFont(new Font("Arial", Font.BOLD, 30));
    g2d.drawString("Tokaj-Hegyalja Egyetem", 220, 750);
}
```

### 2.4.3 Kép Kirajzolása

A JFrame létrehozása és megjelenítése:

```
public static void main(String[] args) {  
    JFrame frame = new JFrame("Heart Image Example");  
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    frame.add(new BM7SGF_projektkompgraf());  
  
    frame.setSize(800, 800);  
    frame.setLocationRelativeTo(null);  
    frame.setVisible(true);  
}
```

### 2.4.4 HeartShape Osztály

A szív alakú minta létrehozásához egy saját `HeartShape` osztályt definiálunk, amely a `java.awt.geom.Path2D.Double` osztályból származik. Ez az osztály határozza meg a szív alakú minta pontjait és íveit.

```
class HeartShape extends java.awt.geom.Path2D.Double {  
    public HeartShape(int x, int y, int width) {  
        super();  
        double w = width * 0.5;  
        double h = width * 0.8;  
        moveTo(x + w, y + h);  
        curveTo(x + w, y + h - 0.3 * h, x + w - 0.5 * w, y + h - 0.8 * h, x + w -  
0.5 * w, y + 0.2 * h);  
        curveTo(x + w - 0.5 * w, y - 0.6 * h, x + w, y - 0.9 * h, x + w, y + 0.2 *  
h);  
        curveTo(x + w, y - 0.9 * h, x + w + 0.5 * w, y - 0.6 * h, x + w + 0.5 * w,  
y + 0.2 * h);  
        curveTo(x + w + 0.5 * w, y + h - 0.8 * h, x + w, y + h - 0.3 * h, x + w, y  
+ h);  
        closePath();  
    }  
}
```

## 2.5 Felhasználói Interakció

Az alkalmazás elindításakor egy ablak jelenik meg, amely a szív alakú maszkot és a képet tartalmazza. A felhasználó nem tud közvetlenül interakcióba lépni az alkalmazással, a kép és a szív alakú maszk statikusak.

## Összefoglalás

Ez a dokumentáció bemutatta a két projekt kódját és működését:

**1. Rotating Cube:**

- Pygame segítségével egy forgó kockát jelenít meg.
- A felhasználó a nyílbillentyűkkel forgathatja a kockát.

**2. BM7SGF Projektkompgraf:**

- Java Swing segítségével egy szív alakú maszkot alkalmaz egy képre.
- A felhasználó elindíthatja az alkalmazást, de nem léphet vele közvetlen interakcióba.

Mindkét projekt egyszerű grafikus alkalmazás, amely különböző programozási nyelvek és könyvtárak használatát demonstrálja a grafikai elemek megjelenítéséhez és manipulálásához.