Ngarkimi i imazheve, efektet vizuele dhe animimi

Faton Berisha

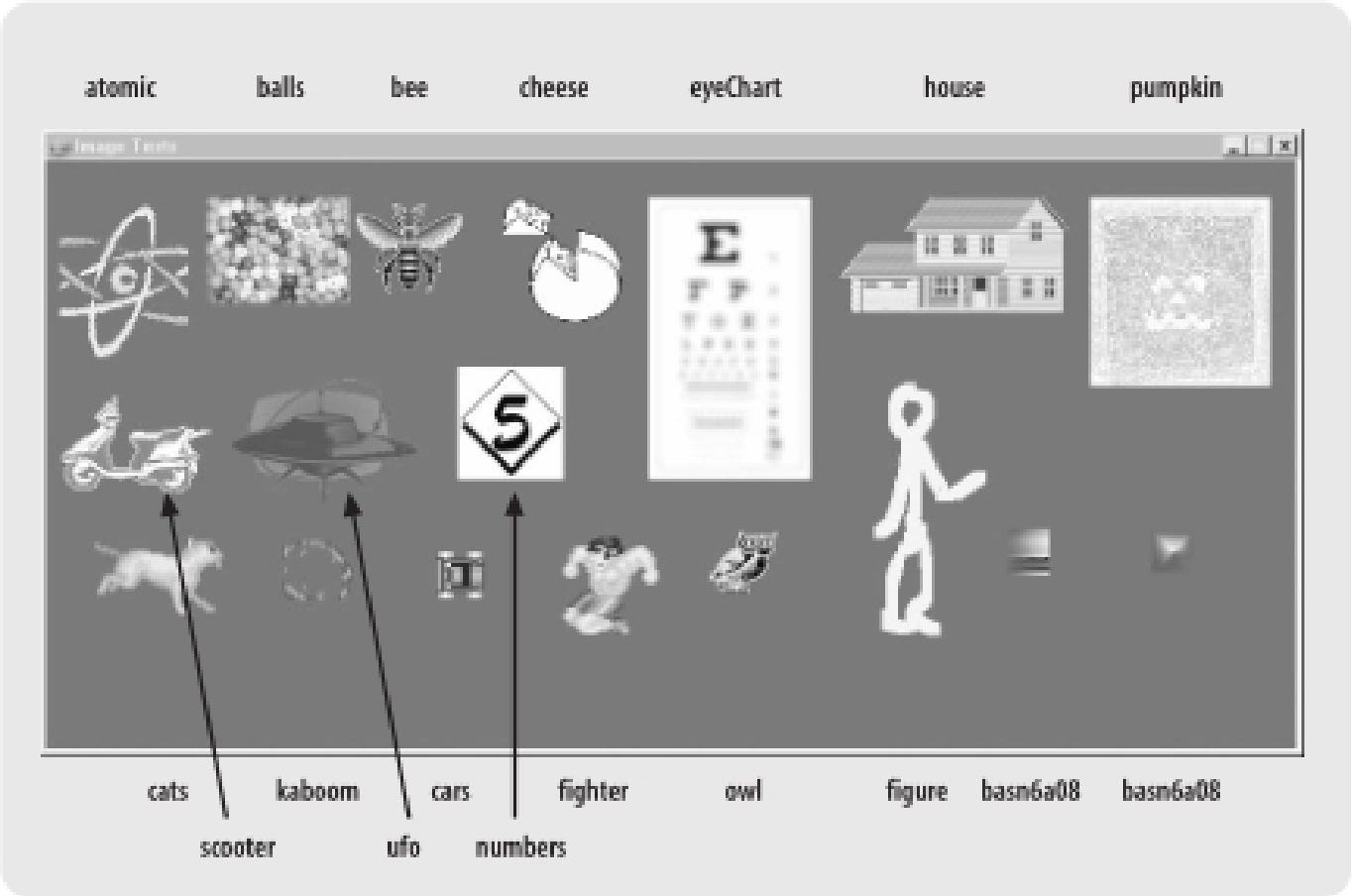
Fakulteti i Shkencave Kompjuterike
Universiteti i Prizrenit

Qëllimet dhe objektivat

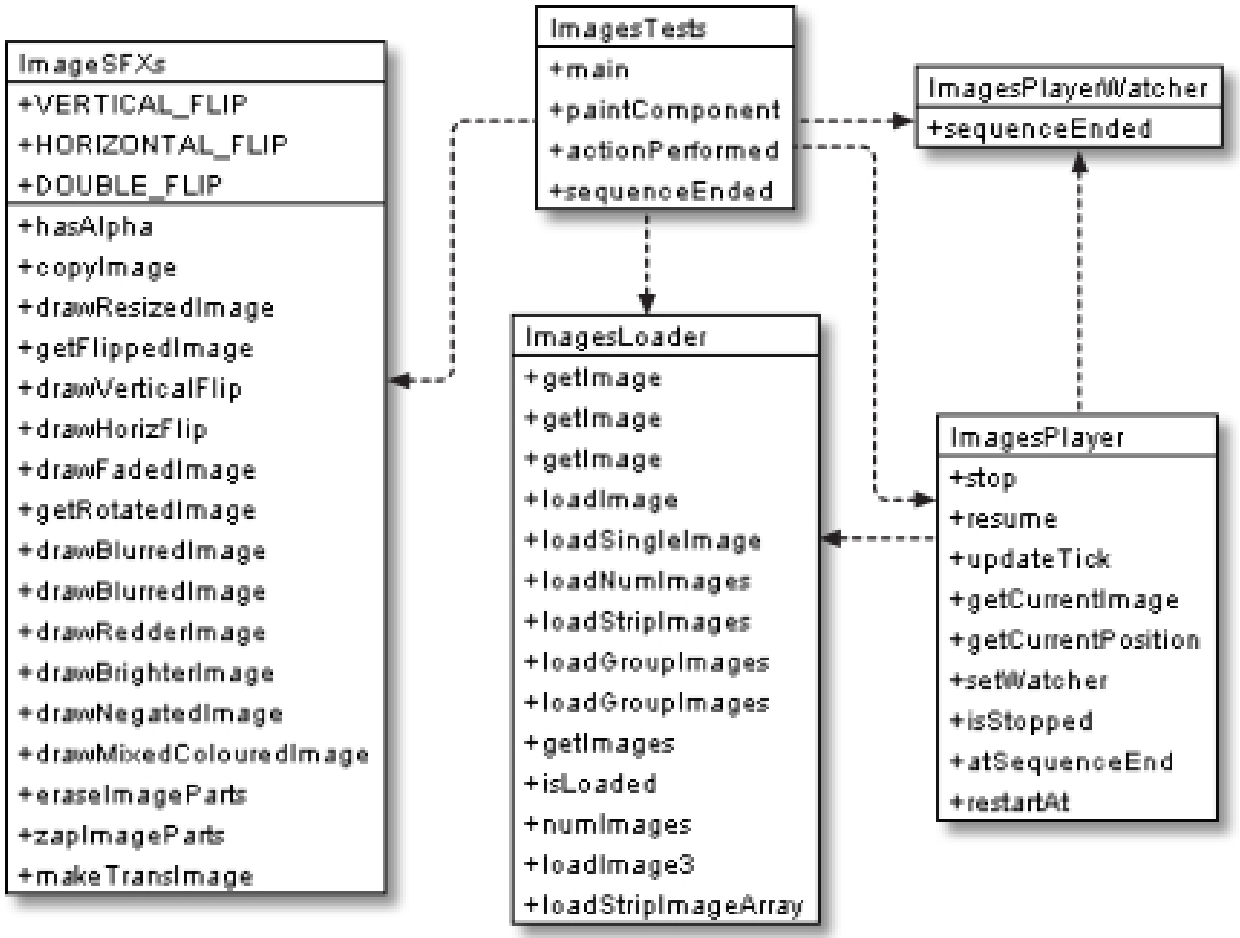
- Ngarkimi dhe afishimi i imazheve, aplikimi i efekteve vizuele, si kthjelltësia (bluring), zbehja (fading) e rotacioni, dhe animimi i tyre
- Ndërtimi i një klase `ImagesLoader` e cila ngarkon imazhe individuale, shirite imazhesh dhe imazhe të shumëfishta
- Implementimi i efekteve të dy kategorive: duke aplikuar në mënyrë të vazhdueshme një efekt vizuel në imazhin, dhe animimi i paraqitur me një seri imazhesh të ndryshme

Përmbajtja

Aplikacioni ImagesTests



Diagramet e klasave



Ngarkimi i imazheve

- Klasa `ImagesLoader` mund të ngarkojë katër formate të ndryshme imazhesh: o, n, s, g.
- Përdorimi me anë të një fajli konfigurues ose load metodave
  - Në aplikacionin është përdorur `imsInfo.txt`
- Fajlat e imazheve duhet të jenë në nënfolderin `Images/`

Ngarkimi i imazheve (Vazhdim)

```
// imsInfo.txt imazhet

o atomic.gif
o balls.jpg
o bee.gif
o cheese.gif
o eyeChart.gif
o house.gif
o pumpkin.png
o scooter.gif
o ufo.gif
o owl.png

n numbers*.gif  6
n figure*.gif 9

g fighter  left.gif right.gif still.gif up.gif

s cars.gif 8
s cats.gif 6
s kaboom.gif 6

o basn6a08.png
o basn6a16.png
```

Sintaksa e fajlit konfigurues

```
o <fnm>
n <fnm*.ext> <number>
s <fnm> <number>
g <name> <fnm> [ <fnm> ]*
```

Shfrytëzimi i ImagesLoader

- Qasja o-imazheve

```
BufferedImage atomic = imsLoader.getImage("atomic");
```

- Qasja n, s, g-imazheve

```
BufferedImage cats1 = imsLoader.getImage("cats", 1);
int numCats = imsLoader.numImage("cats");
```

- Qasja g-imazheve

```
BufferedImage leftFighter = imsLoader.getImage("fighter", "left");
```

Detaje implementimi

```
private void loadImagesFile(String fnm) {
  String imsFNm = IMAGE_DIR + fnm;
  System.out.println("Reading file: " + imsFNm);
  try {
    InputStream in = this.getClass().getResourceAsStream(imsFNm);
    BufferedReader br = new BufferedReader( new InputStreamReader(in));
    // BufferedReader br = new BufferedReader( new FileReader(imsFNm));
    String line;
    char ch;
    while((line = br.readLine()) != null) {
      if (line.length() == 0)  // rresht bosht
        continue;
      if (line.startsWith("//"))   // koment
        continue;
      ch = Character.toLowerCase( line.charAt(0) );
      if (ch == 'o')  // një imazh
        getFileNameImage(line);
      else if (ch == 'n')  // varg i numëruar imazhesh
        getNumberedImages(line);
      else if (ch == 's')  // shirit imazhesh
        getStripImages(line);
      else if (ch == 'g')  // grup imazhesh
        getGroupImages(line);
      else
        System.out.println("Do not recognize line: " + line);
    }
    br.close();
  }
```

Detaje implementimi (Vazhdim)

```
  catch (IOException e) {
    System.out.println("Error reading file: " + imsFNm);
    System.exit(1);
  }
}
```

Detaje implementimi (Vazhdim)

```
private void getFileNameImage(String line) {
  StringTokenizer tokens = new StringTokenizer(line);

  if (tokens.countTokens() != 2)
    System.out.println("Wrong no. of arguments for " + line);
  else {
    tokens.nextToken();    // kapërce komandën
    System.out.print("o Line: ");
    loadSingleImage(tokens.nextToken());
  }
}
```

Detaje implementimi (Vazhdim)

```
// Mund të invokohet drejtpërdrejt
public boolean loadSingleImage(String fnm) {
  String name = getPrefix(fnm);

  if (imagesMap.containsKey(name)) {
    System.out.println( "Error: " + name + "already used");
    return false;
  }

  BufferedImage bi = loadImage(fnm);
  if (bi != null) {
    ArrayList imsList = new ArrayList();
    imsList.add(bi);
    imagesMap.put(name, imsList);
    System.out.println("  Stored " + name + "/" + fnm);
    return true;
  }
  else
    return false;
}
```

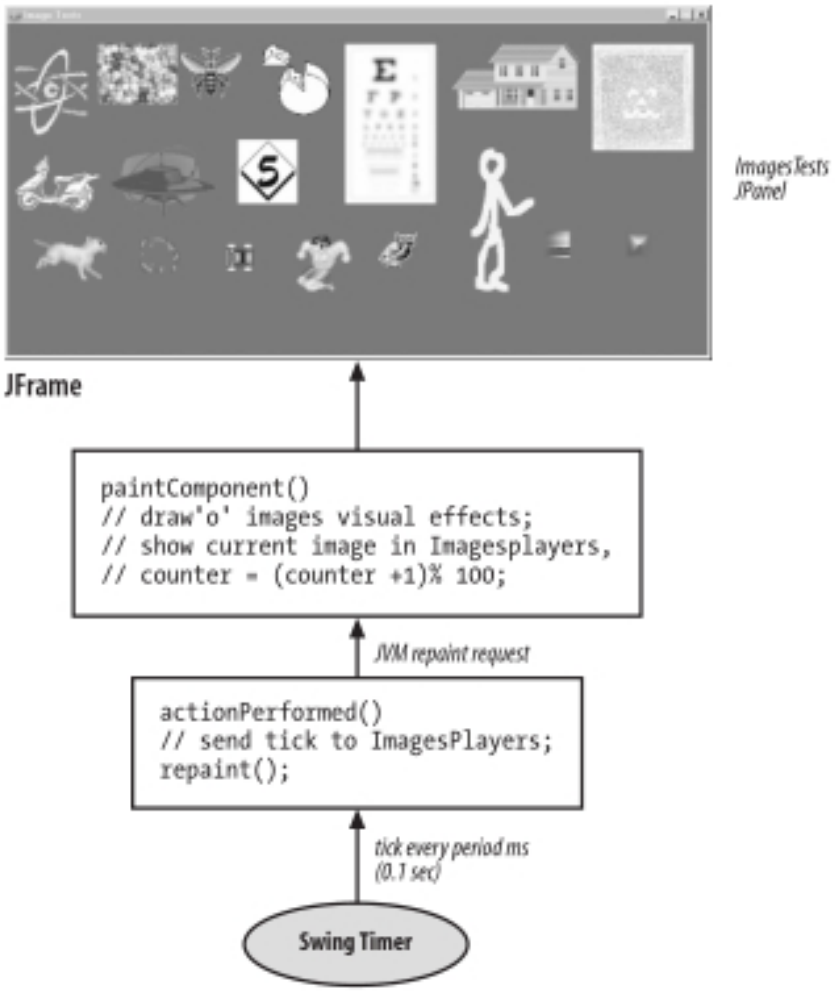
Ngarkimi i një imazhi

```
public BufferedImage loadImage(String fnm) {
  try {
    BufferedImage im =  ImageIO.read(
        getClass().getResource(IMAGE_DIR + fnm) );

    // kopjo për të krijuar një imazh të menagjueshëm
    int transparency = im.getColorModel().getTransparency();
    BufferedImage copy =  gc.createCompatibleImage(
        im.getWidth(), im.getHeight(),
        transparency );
    // krijo një kontekst grafik
    Graphics2D g2d = copy.createGraphics();

    // kopjo imazhin
    g2d.drawImage(im,0,0,null);
    g2d.dispose();
    return copy;
  }
  catch(IOException e) {
    System.out.println("Load Image error for " +
        IMAGE_DIR + "/" + fnm + ":\n" + e);
    return null;
  }
}
```

Aplikimi i efekteve vizuele



Aplikimi i efekteve vizuele (Vazhdim)

```
public class ImagesTests extends JPanel
    implements ActionListener, ImagesPlayerWatcher {

  // ... anëtarët tjerë

  public static void main(String args[]) {
    ImagesTests ttPanel = new ImagesTests();

    JFrame app = new JFrame("Image Tests");
    app.getContentPane().add(ttPanel);
    app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    app.pack();
    app.setResizable(false);
    app.setVisible(true);
}
```

Aplikimi i efekteve vizuele (Vazhdim)

```
// ... fushat tjera
private ImagesLoader imsLoader;   // ngarkuesi i imazheve
private int counter;
private boolean justStarted;
private ImageSFXs imageSfx;    // klasa e efekteve vizuele
private GraphicsDevice gd;     // për raportim
private int accelMemory;
private DecimalFormat df;
public ImagesTests() {
  df = new DecimalFormat("0.0");  // 1 dp
  GraphicsEnvironment ge =
      GraphicsEnvironment.getLocalGraphicsEnvironment();
  gd = ge.getDefaultScreenDevice();
  accelMemory = gd.getAvailableAcceleratedMemory();  // në byte
  System.out.println("Initial Acc. Mem.: " +
      df.format( ((double)accelMemory)/(1024*1024) ) + " MB" );
  setBackground(Color.white);
  setPreferredSize( new Dimension(PWIDTH, PHEIGHT) );
  // ngarko dhe inicializo imazhet
  imsLoader = new ImagesLoader(IMS_FILE);   // "imsInfo.txt"
  imageSfx = new ImageSFXs();
  initImages();
  counter = 0;
  justStarted = true;
  new Timer(PERIOD, this).start();    // PERIOD = 0.1 s
}
```

Aplikimi i efekteve vizuele (Vazhdim)

```
// o-imazhet
private BufferedImage atomic, balls, bee, cheese, eyeChart,
    house, pumpkin, scooter, fighter, ufo, owl, basn8, basn16;
// n- dhe s-imazhet
private ImagesPlayer numbersPlayer, figurePlayer, carsPlayer,
    catsPlayer, kaboomPlayer;

private void initImages() {
  // inicializo fushat e o-imazheve
  atomic = imsLoader.getImage("atomic");
  balls = imsLoader.getImage("balls");
  bee = imsLoader.getImage("bee");
  cheese = imsLoader.getImage("cheese");
  eyeChart = imsLoader.getImage("eyeChart");
  house = imsLoader.getImage("house");
  pumpkin = imsLoader.getImage("pumpkin");
  scooter = imsLoader.getImage("scooter");
  ufo = imsLoader.getImage("ufo");
  owl = imsLoader.getImage("owl");
  basn8 = imsLoader.getImage("basn6a08");
  basn16 = imsLoader.getImage("basn6a16");
```

Aplikimi i efekteve vizuele (Vazhdim)

```
  // inicializo fushat ImagesPlayer pët n- dhe s-imazhet
  numbersPlayer =
      new ImagesPlayer("numbers", PERIOD, 1, false, imsLoader);
  numbersPlayer.setWatcher(this);  // raporto prapa fundin e vargut

  figurePlayer =
      new ImagesPlayer("figure", PERIOD, 2, true, imsLoader);
  carsPlayer =
      new ImagesPlayer("cars", PERIOD, 1, true, imsLoader);
  catsPlayer =
      new ImagesPlayer("cats", PERIOD, 0.5, true, imsLoader);
  kaboomPlayer =
      new ImagesPlayer("kaboom", PERIOD, 1.5, true, imsLoader);

  // inicializo fushën e g-imazhit
  fighter = imsLoader.getImage("fighter", "left");
}
```

Udhëzime për lexim të mëtejmë

- `http://www.fberisha.org`
- A. Davison, *Killer Game Programming in Java,* kapitujt 5, 6
  `http://fivedots.coe.psu.ac.th/ ad/jg/`
- A. Feldman, *SpriteLib,*
  `http://www.arifeldman.com/games/spritelib.html`

Përfundim

- Zhvillimi i një kornize për ngarkim e afishim imazhesh.
- Ndërtimi i `class ImagesLoader` për ngarkim imazhesh
- Paisja e kornizës me efekte vizuele të imazheve