



CFGS - Desenvolupament d'aplicacions multiplataforma

Mòdul 9 – Programació de serveis i processos

UF2 – Processos i fils

Per l'activitat heu de lliurar un PDF amb el següent contingut:

- Nom o noms dels membres que fan l'activitat (màxim 2 alumnes).
- URL del repositori on hi ha els fonts (github, bitbucket,...).
- Noms d'usuari utilitzats per cada membre del grup del repositori del núvol.
- Explicacions, línies de codi creades/modificades i bolcats de pantalla de l'execució dels diferents algorismes que es demanen en la tasca.
- La claredat de la documentació lliurada (a banda del codi) serà directament proporcional a la nota obtinguda.

El pdf s'ha d'anomenar seguint el format NomCognom1Cognom2.pdf del membre que fa el lliurament. Si un grup està format per dos membres NOMÉS UN l'ha de lliurar.

Tasques entregades fora de termini i/o on els fonts del repositori del núvol hagin estat creats/modificats fora de termini tindran, com a màxim un 5 sempre que el lliurament i/o la modificació del repositori sigui de, com a molt, 3 dies després del venciment. Passats 3 dies no s'avaluarà la tasca i es qualificarà com a 0.

No s'acceptarà tampoc cap tasca que no es lliuri en el format especificat i/o que no contingui de forma clara els punts que es demanen.

Enllaços d'interès:

<https://docs.oracle.com/javase/7/docs/api/index.html?java/lang/Thread.html>

Objectius:

Identificar i utilitzar les paraules reservades del llenguatge Java per gestió de processos/fils

Activitat 7

A partir del codi que es subministra:

- 1) Si es pitja la tecla d'espai la nostra nau ha de disparar
- 2) Per a cada dispar s'ha de crear un fil que gestionarà un raig làser.
- 3) El raig làser ha de partir de la nostra nau i anar cap amunt amb certa velocitat
- 4) Si el raig làser és prop d'alguna nau que no sigui la nostra, l'ha de destruir
- 5) El joc ha d'acabar si destruïm totes les naus (inicialment, enlloc de 3 que n'hi hagi 10)

AMPLIACIÓ/NO_DUAL

- 6) Si les naus enemigues xoquen amb la nostra perdem la partida i s'acaba el joc
- 7) les naus enemigues també poden disparar-nos (en aquest cas, de forma aleatòria i els seus dispars van cap avall). Si els dispars en toquen, perdem partida i s'acaba el joc.

ATENCIÓ: Penseu que l'important, a banda de crear els fils, es detectar aquells mètodes o parts de codi que han d'estar sincronitzades. La màxima nota s'obté si, a banda de sincronitzar mètodes/codi, agrupem Threads per gestionar la sincronització i usem les prioritats adients per tenir un funcionament fluid.



Arxiu	Document extern
Elaborat	Cap d'estudis

Codi	MO-CAP012			1 de 1
Versió	6	Data	19/12/2018	



NOMS

Raúl Santos Espino

Tiffany Fernández Anguera

URL

<https://github.com/TiffanyFA/m9uf2>

NOMS USUARIS

RaulSE98

TiffanyFA

EVIDÈNCIES

- 1) Si es pitja la tecla d'espai la nostra nau ha de disparar

```
182     public void keyPressed(KeyEvent e) {  
183         // System.out.println("Key pressed code=" + e.getKeyCode() + ", char=" +  
184         // e.getKeyChar());  
185         if (e.getKeyCode() == 37) {  
186             nauPropia.esquerra();  
187         }  
188         if (e.getKeyCode() == 39) {  
189             nauPropia.dreta();  
190         }  
191         if (e.getKeyCode() == 32) {  
192             colorDispar = "disparBlau";  
193             dispar = new Dispar(colorDispar, nauPropia.getPosicioX()+35, nauPropia.getPosicioY(), -3, 100);  
194             dispars.add(dispar);  
195         }  
196     }  
197 }
```

- 2) Per a cada dispar s'ha de crear un fil que gestionarà un raig làser.

```
298     class Dispar extends Thread {  
299         private int x, y;  
300         private int dsy, v;  
301         private int ty = 10;  
302         private boolean seguir;  
303         private String img = "disparVermell.jpg";  
304         private Image image;  
305         private String colorDispar = "disparVermell";  
306         ThreadGroup grupDispars = new ThreadGroup ("dispars");  
307  
308         public Dispar (String colorDispar, int x, int y, int dsy, int v) {  
309             this.x = x;  
310             this.y = y;  
311             this.dsy = dsy;  
312             this.v = v;  
313             seguir = true;  
314             image = new ImageIcon(Nau.class.getResource(colorDispar + ".png")).getImage();  
315             Thread t = new Thread(grupDispars, this, "blaus");  
316             t.start();  
317         }  
318     }
```



Arxiu	Document extern
Elaborat	Cap d'estudis

Codi	MO-CAP012			2 de 1
Versió	6	Data	19/12/2018	



- 3) El raig làser ha de partir de la nostra nau i anar cap amunt amb certa velocitat

```

182 public void keyPressed(KeyEvent e) {
183     // System.out.println("Key pressed code=" + e.getKeyCode() + ", char=" +
184     // e.getKeyChar());
185     if (e.getKeyCode() == 37) {
186         nauPropia.esquerra();
187     }
188     if (e.getKeyCode() == 39) {
189         nauPropia.dreta();
190     }
191     if (e.getKeyCode() == 32) {
192         colorDispar = "disparBlau";
193         dispar = new Dispar(colorDispar, nauPropia.getPosicioX()+35, nauPropia.getPosicioY(), -3, 100);
194         dispars.add(dispar);
195     }
196 }

```

- 4) Si el raig làser és prop d'alguna nau que no sigui la nostra, l'ha de destruir

```

202 public void comprovaDispars(){
203     //En cas que el dispar impacti amb una nau
204     for (int i = 0; i < dispars.size(); i++) {
205         Dispar index = dispars.get(i);
206         for (int j = 0; j < nau.length; j++) {
207             if (nau[j]!=null) {
208                 double calcul;
209                 calcul = Math.sqrt(Math.pow(nau[j].getPosicioX() - index.getPosicioX(), 2) +
210                 Math.pow(nau[j].getPosicioY() - index.getPosicioY(), 2));
211                 if (calcul < 10){
212                     System.out.println("XOCA!!");
213                     nau[j].noSeguir();
214                     nau[j] = null;
215                     dispars.remove(index);
216                     i--;
217                 }
218             }
219         }
220     }
221 }

```

- 5) El joc ha d'acabar si destruïm totes les naus (inicialment, enlloc de 3 que n'hi hagi 10)

```

141 public void run() {
142     while (continuar) {
143         try {
144             Thread.sleep(100);
145         } catch (Exception e) {
146         } // espera 0,1 segons
147         //fa explotar el dispar si arriba al final de la pantalla
148         explota();
149         //destrueix les naus que són disparades
150         comprovaDispars();
151         //comprova si queden naus enemigues
152         comprovarEnemics();
153         //comprovar si alguna nau xoca amb mi
154         comprovarXoc();
155         // System.out.println("Repintant");
156         repaint();
157     }
158     //tanca el joc
159     System.exit(0);
160 }

```



Arxiu	Document extern
Elaborat	Cap d'estudis

Codi	MO-CAP012			3 de 1
Versió	6	Data	19/12/2018	



- 6) Si les naus enemigues xoquen amb la nostra perdem la partida i s'acaba el joc

```
141 public void run() {  
142     while (continuar) {  
143         try {  
144             Thread.sleep(100);  
145         } catch (Exception e) {  
146             // espero 0,1 segons  
147             //fa explotar el dispar si arriba al final de la pantalla  
148             explota();  
149             //destrueix les naus que són disparades  
150             comprovaDispars();  
151             //comprova si queden naus enemigues  
152             comprovarEnemics();  
153             //comprovar si alguna nau xoca amb mi  
154             comprovarXoc();  
155             // System.out.println("Repintant");  
156             repaint();  
157         }  
158         //tanca el joc  
159         System.exit(0);  
160     }  
}
```

- 7) les naus enemigues també poden disparar-nos (en aquest cas, de forma aleatòria i els seus dispars van cap avall). Si els dispars en toquen, perdem partida i s'acaba el joc.



Arxiu	Document extern
Elaborat	Cap d'estudis

Codi	MO-CAP012			4 de 1
Versió	6	Data	19/12/2018	



```
package uf2.practica;

import java.awt.*;
import java.util.*;
import javax.swing.*;
import java.awt.event.*;

public class NauEspaial extends javax.swing.JFrame {

    public NauEspaial() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setBackground(new java.awt.Color(255, 255, 255));
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);

        layout.setHorizontalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(40, 40, Short.MAX_VALUE)
                .addGap(0, 400, Short.MAX_VALUE));

        layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(40, 40, Short.MAX_VALUE)
                .addGap(0, 300, Short.MAX_VALUE));

        pack();
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (Exception ex) {
            java.util.logging.Logger.getLogger(NauEspaial.class.getName()).log(
                java.util.logging.Level.SEVERE, null, ex);
        }
        NauEspaial f = new NauEspaial();
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setTitle("Naus Espais");
        f.setContentPane(new PanelNau());
        f.setSize(480, 560);
        f.setVisible(true);
    }
}
```



Arxiu	Document extern
Elaborat	Cap d'estudis

Codi	MO-CAP012			5 de 1
Versió	6	Data	19/12/2018	



```
class PanelNau extends JPanel implements Runnable, KeyListener{
    private int numNaus=3;
    Nau[] nau;
    Nau nauPropia;

    public PanelNau(){
        nau = new Nau[numNaus];
        for (int i=0;i<nau.length;i++) {
            Random rand = new Random();
            int velocitat=(rand.nextInt(3)+5)*10;
            int posX=rand.nextInt(100)+30;
            int posY=rand.nextInt(100)+30;
            int dX=rand.nextInt(3)+1;
            int dY=rand.nextInt(3)+1;
            String nomNau = Integer.toString(i);
            nau[i]= new Nau(nomNau,posX,posY,dX,dY,velocitat);
        }

        // Creo la nau propia
        nauPropia = new Nau("NauNostra",200,400,10,0,100);

        // Creo fil per anar pintant cada 0,1 segons el joc per pantalla
        Thread n = new Thread(this);
        n.start();

        // Creo listeners per a que el fil principal del programa gestioni
        // esdeveniments del teclat
        addKeyListener(this);
        setFocusable(true);

    }

    public void run() {
        System.out.println("Inici fil repintar");
        while(true) {
            try { Thread.sleep(100);} catch(Exception e) {} // espero 0,1 segons
            //System.out.println("Repintant");
            repaint();
        }
    }

    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        for(int i=0; i<nau.length;++i) nau[i].pinta(g);
        nauPropia.pinta(g);
    }

    // Metodes necessaris per gestionar esdeveniments del teclat
    @Override
```



Arxiu	Document extern
Elaborat	Cap d'estudis

Codi	MO-CAP012			6 de 1
Versió	6	Data	19/12/2018	



```
public void keyTyped(KeyEvent e) {  
}  
  
@Override  
public void keyPressed(KeyEvent e) {  
    //System.out.println("Key pressed code=" + e.getKeyCode() + ", char=" +  
e.getKeyChar());  
    if (e.getKeyCode()==37) { nauPropia.esquerra(); } //System.out.println("a l'esquerra"); }  
    if (e.getKeyCode()==39) { nauPropia.dreta(); } //System.out.println("a la dreta"); }  
}  
  
@Override  
public void keyReleased(KeyEvent e) {  
}  
}
```

```
class Nau extends Thread {  
    private String nomNau;  
    private int x,y;  
    private int dsx,dsy,v;  
    private int tx = 10;  
    private int ty = 10;  
  
    private String img = "/images/nau.jpg";  
    private Image image;  
  
    public Nau(String nomNau, int x, int y, int dsx, int dsy, int v ) {  
        this.nomNau = nomNau;  
        this.x=x;  
        this.y=y;  
        this.dsx=dsx;  
        this.dsy=dsy;  
        this.v=v;  
        image = new ImageIcon(Nau.class.getResource("nau.png")).getImage();  
        Thread t = new Thread(this);  
        t.start();  
    }  
  
    public int velocitat (){  
        return v;  
    }  
  
    public void moure (){  
        x=x + dsx;  
        y=y + dsy;  
        // si arriba als marges ...  
        if ( x>= 450 - tx || x<= tx) dsx = - dsx;  
        if ( y >= 500 - ty || y<=ty ) dsy = - dsy;  
    }  
}
```



Arxiu	Document extern
Elaborat	Cap d'estudis

Codi	MO-CAP012			7 de 1
Versió	6	Data	19/12/2018	



```
public void pinta (Graphics g) {  
    Graphics2D g2d = (Graphics2D)g;  
    g2d.drawImage(this.image, x, y, null);  
}  
  
public void run() {  
    while (true) {  
        //System.out.println("Movent nau numero " + this.nomNau);  
        try { Thread.sleep(this.v); } catch (Exception e) {}  
        moure();  
    }  
}  
  
public void esquerra() {  
    this.dsx = -10;  
}  
  
public void dreta() {  
    this.dsx = 10;  
}  
}
```



Arxiu	Document extern
Elaborat	Cap d'estudis

Codi	MO-CAP012			8 de 1
Versió	6	Data	19/12/2018	