

Colegiul Național “Emil Racoviță”
Cluj-Napoca
2020

SNAKE

Îndrumător,
Prof. Lăpuștea Anca

Autor,
Crișan Filip-Daniel

Cuprins

1. Cuvânt înainte.....	3
2. Descriere.....	4
2.1. Instalare.....	4
2.2. Funcționare.....	4
2.3. Cod sursă.....	8
2.4. Extinderi.....	12
3. Bibliografie.....	13

1. Cuvânt înainte

Probabil singurul joc pe multe telefoane apărute înainte de 2005, Snake este un clasic de care mulți oameni se atașează, iar unul dintre aceștia sunt și eu. Astfel, a fi capabil de a crea o astfel de aplicație de la 0 este pentru mine un vis din copilărie devenit realitate.

Este bine cunoscut și evident faptul că implementarea jocului nu este deloc revoluționară, iar ideea în sine nu este originală. Totuși, consider că am adus un plus de creativitate în ceea ce privește utilizarea structurilor de date și interconectarea lor pentru a ajunge la produsul final.

Mai mult, limbajul de programare ales (Java) este unul foarte ofertant, cu dezvoltări posibile atât în sfera Android, cât și în cea Windows.

Realizarea proiectului a avut un rol semnificativ în reactualizarea principiilor programării orientate-obiect și în dezvoltarea abilității de construire modulară a unui program.

2. Descriere

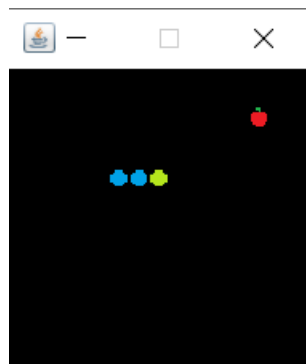
2.1. Instalare

Aplicația vine sub forma unei arhive .zip, care, odată dezarhivată, va oferi accesul la executabilul reprezentat de aplicația în sine. Rularea aplicației necesită software-ul Java (Java Runtime Environment), care conține toate elementele platformei. Descărcarea acestuia este gratuită, accesând unul dintre link-urile de mai jos:

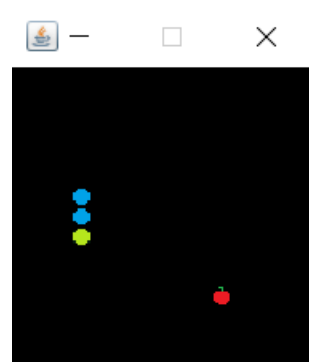
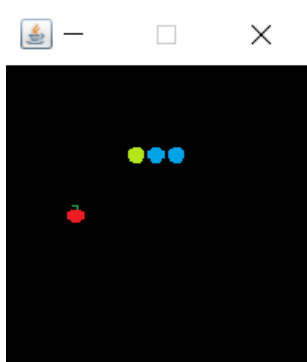
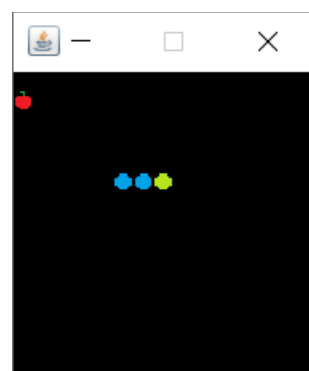
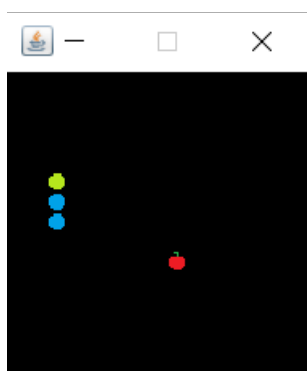
- a. <https://www.oracle.com/java/technologies/javase-jre8-downloads.html>
- b. <https://www.java.com/en/download/windows-64bit.jsp>

2.2. Funcționare

Odată cu deschiderea fișierului Snake.exe, pe ecran va apărea următoarea fereastră, iar utilizatorul poate începe jocul.



Șarpele se poate deplasa în toate direcțiile:

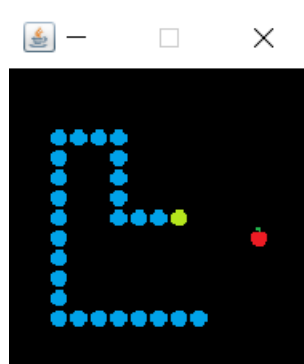


De asemenea, după cum utilizatorul este obișnuit, lungimea șarpelui va crește odată cu înghițirea mărului.

Mărul va apărea, evident, pe o nouă poziție aleatorie.



Șarpele se poate extinde până când corpul său ocupă întregul panou.

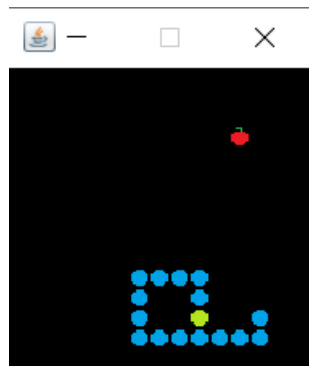


Regulile se aplică, adică de fiecare dată când:

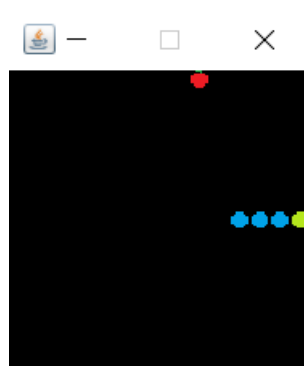
- (1) capul șarpelui va înghiți o parte din propriul corp sau
- (2) șarpele se lovește de limitele panoului,

jocul se încheie.

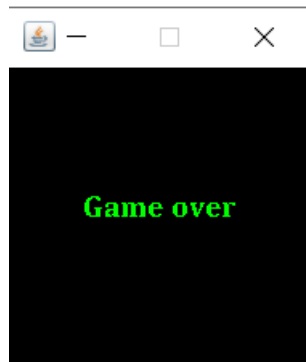
(1)



(2)



În toate aceste cazuri, pe ecran se va afișa:

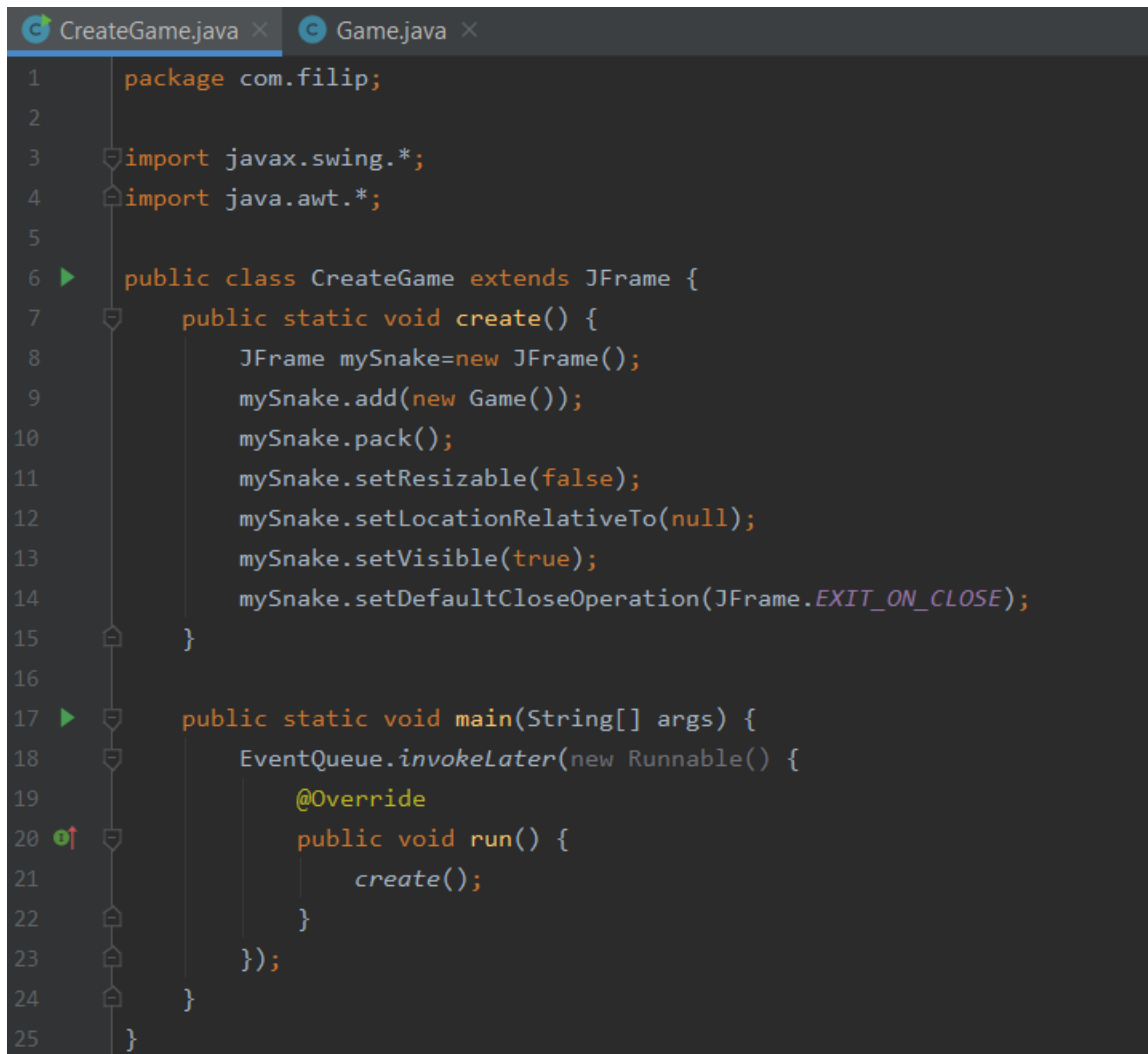


2.3. Cod sursă

Structura programului constă în două clase:

- a. *CreateGame* = creează fereastra cu care utilizatorul interacționează
- b. *Game* = se ocupă de panoul afișat de fereastră și de toate detaliile funcționării jocului

a. *CreateGame*



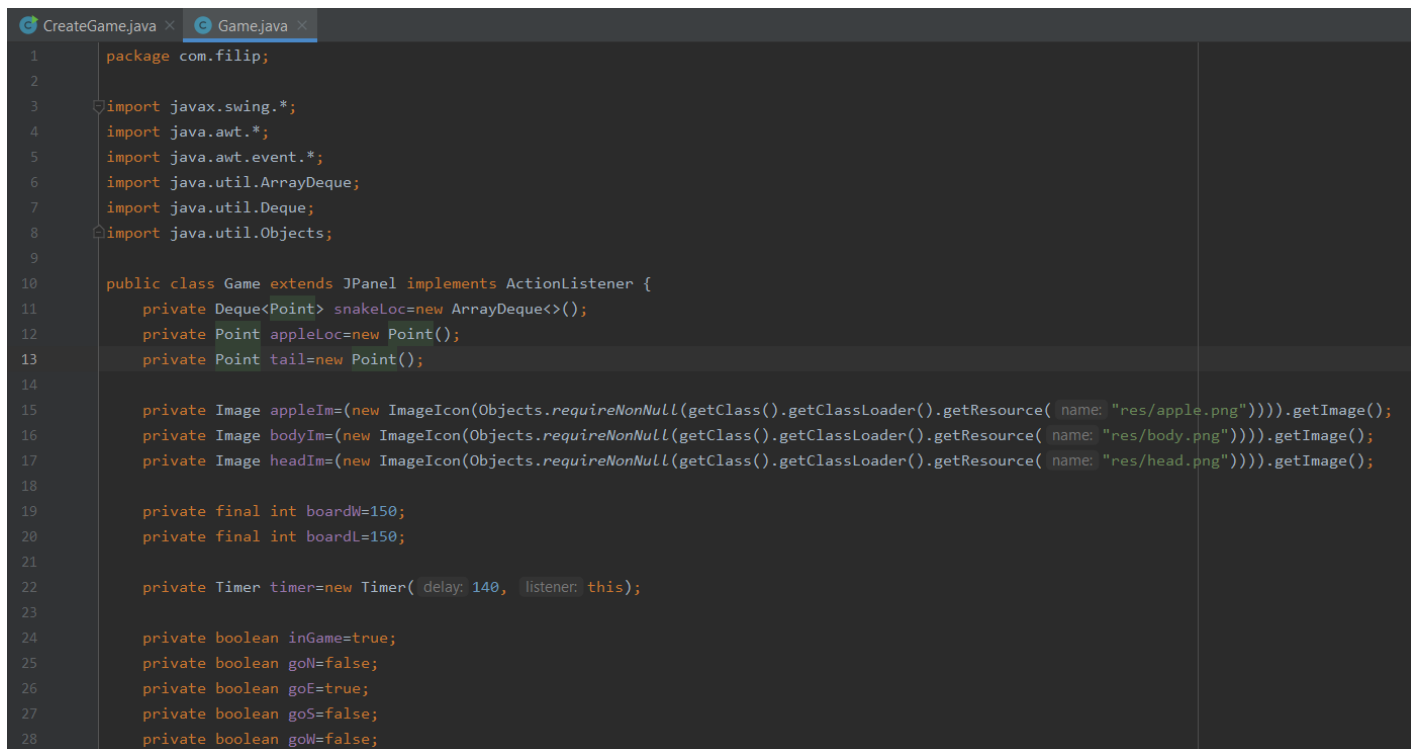
```
1 package com.filip;
2
3 import javax.swing.*;
4 import java.awt.*;
5
6 public class CreateGame extends JFrame {
7     public static void create() {
8         JFrame mySnake=new JFrame();
9         mySnake.add(new Game());
10        mySnake.pack();
11        mySnake.setResizable(false);
12        mySnake.setLocationRelativeTo(null);
13        mySnake.setVisible(true);
14        mySnake.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15    }
16
17    public static void main(String[] args) {
18        EventQueue.invokeLater(new Runnable() {
19            @Override
20            public void run() {
21                create();
22            }
23        });
24    }
25 }
```


Clasa creează un obiect de tip JFrame, peste care adaugă panelul pe care îl voi prezenta în cele ce urmează. De asemenea, se setează diferite detalii, cum ar fi dimensiunile ferestrei, vizibilitatea etc.

b. Game

Din cauza faptului că reprezintă o clasă mai complexă, care inclusiv conține o altă clasă, îmi voi permite să o prezint fragmentată.

b.1. Inițializarea câmpurilor clasei



```
1 package com.filip;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.*;
6 import java.util.ArrayDeque;
7 import java.util.Deque;
8 import java.util.Objects;
9
10 public class Game extends JPanel implements ActionListener {
11     private Deque<Point> snakeLoc=new ArrayDeque<>();
12     private Point appleLoc=new Point();
13     private Point tail=new Point();
14
15     private Image appleIm=(new ImageIcon(Objects.requireNonNull(getClass().getClassLoader().getResource( name: "res/apple.png")))).getImage();
16     private Image bodyIm=(new ImageIcon(Objects.requireNonNull(getClass().getClassLoader().getResource( name: "res/body.png")))).getImage();
17     private Image headIm=(new ImageIcon(Objects.requireNonNull(getClass().getClassLoader().getResource( name: "res/head.png")))).getImage();
18
19     private final int boardW=150;
20     private final int boardL=150;
21
22     private Timer timer=new Timer( delay: 140, listener: this);
23
24     private boolean inGame=true;
25     private boolean goN=false;
26     private boolean goE=true;
27     private boolean goS=false;
28     private boolean goW=false;
```

b.2. Constructor-ul și funcția de pornire a jocului

```
30     public Game() {
31         setBackground(Color.black);
32         setPreferredSize(new Dimension(boardL, boardW));
33
34         setFocusable(true); // to fire keyboard events
35         addKeyListener(new changeDir());
36
37         startGame();
38     }
39
40     private void startGame() {
41         snakeLoc.clear();
42
43         for(int i=1;i<=3;i++)
44             snakeLoc.addLast(new Point( x: 50-i*10, y: 50));
45
46         tail=new Point(snakeLoc.getLast());
47
48         createAL();
49
50         timer.start();
51     }
```

b.3. Funcția principală de mutare a șarpelui

```
73     private void moveSnake() {
74         Point nextHP=new Point(snakeLoc.getFirst());
75         if(goN)
76             nextHP.y-=10;
77         else if(goE)
78             nextHP.x+=10;
79         else if(goS)
80             nextHP.y+=10;
81         else if(goW)
82             nextHP.x-=10;
83
84         tail=new Point(snakeLoc.getLast());
85
86         snakeLoc.removeLast();
87         snakeLoc.addLast(nextHP);
88         int length=snakeLoc.size()-1;
89         while(length>=1) {
90             snakeLoc.addLast(snakeLoc.getFirst());
91             snakeLoc.removeFirst();
92             length--;
93         }
94     }
```

b.4. Funcția care este executată de fiecare dată la un interval setat de timer și pune totul în mișcare

```
124     @Override
125     public void actionPerformed(ActionEvent e) {
126         if(inGame) {
127             grow();
128             moveSnake();
129             checkInGame();
130         }
131
132         repaint();
133     }
```

b.5. Funcția care afișează GUI-ul

```
135     @Override
136     public void paintComponent(Graphics g) {
137         super.paintComponent(g);
138
139         if(inGame) {
140             g.drawImage(appleIm, appleLoc.x, appleLoc.y, observer: this);
141
142             int length=snakeLoc.size();
143             Point headLoc=new Point(snakeLoc.getFirst());
144             while(length>=1) {
145                 if(snakeLoc.getFirst().equals(headLoc))
146                     g.drawImage(headIm, snakeLoc.getFirst().x, snakeLoc.getFirst().y, observer: this);
147                 else
148                     g.drawImage(bodyIm, snakeLoc.getFirst().x, snakeLoc.getFirst().y, observer: this);
149                 snakeLoc.addLast(snakeLoc.getFirst());
150                 snakeLoc.removeFirst();
151                 length--;
152             }
153             Toolkit.getDefaultToolkit().sync();
154         }
155         else
156             gameOver(g);
157     }
```

b.6. Alte funcții, precum cea de mutare a mărului, de verificare a continuării jocului sau de schimbare a direcției șarpelui.

2.4. Extinderi

Identific dezvoltări ale aplicației pe următoarele niveluri:

- Un buton de start și de reîncepere a jocului
- O interfață mai prietenoasă cu utilizatorul
- O bază de date care să rețină highscore-ul jucătorului.

3. Bibliografie

- a. <https://docs.oracle.com/javase/tutorial/>
- b. <https://stackoverflow.com/questions/tagged/java>
- c. <https://javatutorial.net/jframe-buttons-listeners-text-fields>