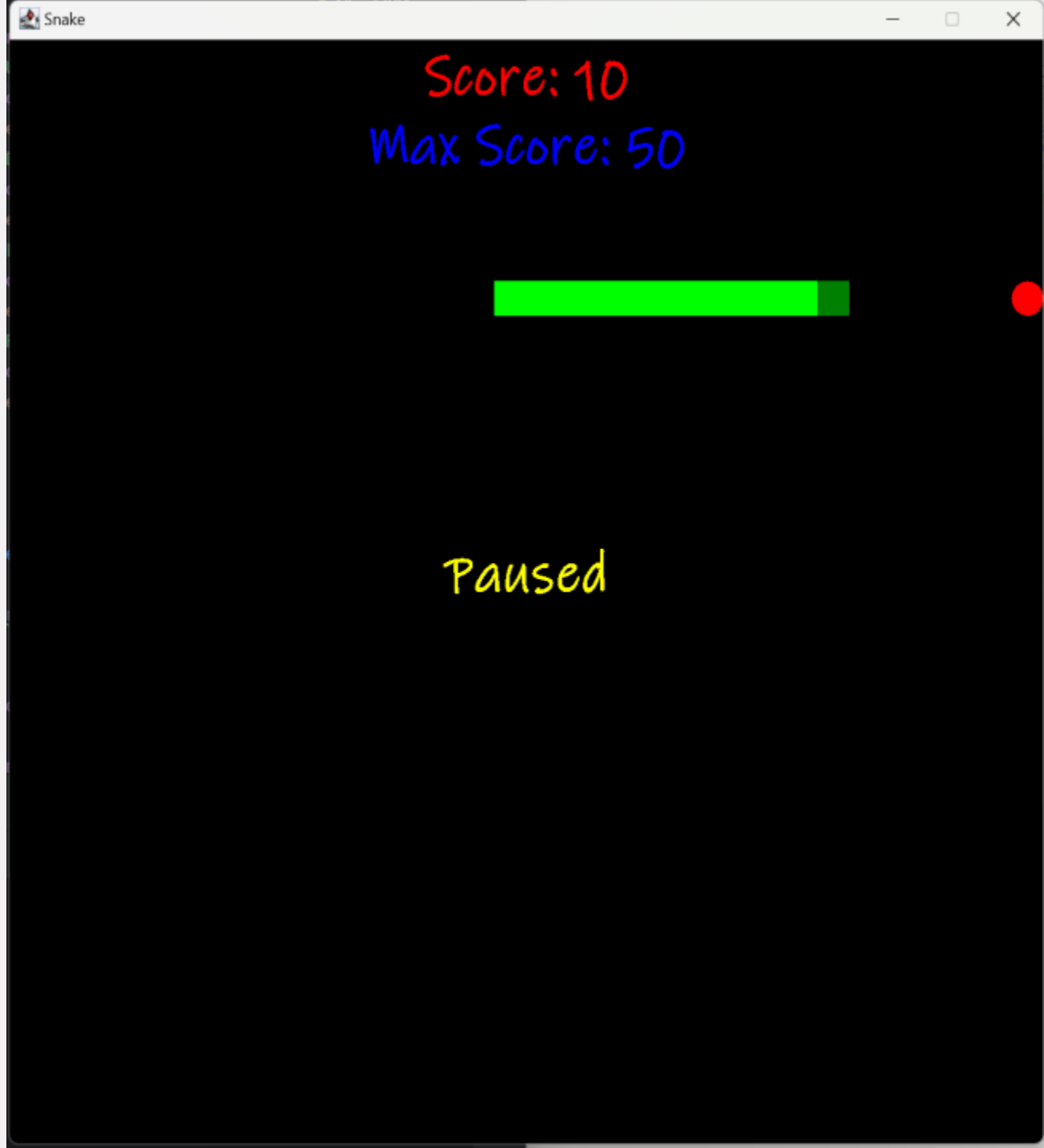


Ureche Simona Elena - An 2 -
Calculatoare Romana -
grupa 30224

SNAKE JAVA GAME

Proiect POO

Prof.: Hulea Andrei



CUPRINS:



- Introducere - scopul proiectului;
- Structura proiectului - pachete si clase;
- Functionalitati proiect - scopul claselor;
- Regulile jocului - exemple de utilizare;
- Posibile imbunatatiri viitoare;
- Diagrama UML;

INTRODUCERE:

Proiectul "Snake Game" este o implementare a clasicului joc Snake în limbajul de programare Java, folosind platforma grafică Swing. Acest proiect reprezintă o aplicație interactivă și distractivă, care permite utilizatorilor să controleze un șarpe pe un ecran și să interacționeze cu diverse obiecte, cum ar fi merele, pentru a crește scorul și totodată corpul sarpelui.

Scopul principal al acestui proiect este de a oferi o implementare funcțională și distractivă a jocului Snake. Acest proiect exemplifică utilizarea conceptelor de programare orientată pe obiect, interfețe grafice și gestionarea evenimentelor în Java.

STRUCTURA PROIECTULUI:

Proiectul este structurat in pachetul '**MySnakePack**', format la randul lui din 4 clase:

- **GUI** - O extensie a clasei **JPanel** care implementeaza metode ale clasei **ActionListener** (*actionPerformed* - acesta metoda este legata la un timer si controleaza daca jocul este in desfasurare). Această clasă reprezintă interfața grafică a jocului Snake. Se ocupă de desenarea elementelor, gestionarea evenimentelor de tastatură și logica jocului.
- **GameFrame** – O extensie a clasei **JFrame** care servește drept fereastra principală a aplicației și conține instanța **GUI**.
- **StartButtonPanel** - Clasă pentru panoul de butoane care conține butonul "Start". Se ocupă de gestionarea evenimentelor de start și ascunderea panoului la începerea jocului.
- **MyKeyAdapter** - O clasă care extinde **KeyAdapter** și gestionează evenimentele tastaturii pentru controlul direcției șarpelui în cadrul clasei **GUI**

FUNCTIONALITATI PROIECT - SCOPUL CLASELOR - CLASA GUI:

- **Constructorul GUI()** : Initializează variabilele și obiectele necesare pentru desenarea jocului si stabilește dimensiunile ferestrei și alte setări ale acesteia. De asemenea adaugă un `KeyListener` pentru a detecta evenimentele de tastatură (implementat în clasa `MyKeyAdapter`) si inițializează un buton de start și îl adaugă la panou.
- **Metoda initializeGame()**: Initializează variabilele pentru începerea unui nou joc, cum ar fi coordonatele șarpelui, viteza, numărul de mere mancate, dimensiunea șarpelui, etc . De asemenea generează direcția și poziția inițială a șarpelui random.
- **Metoda generateRandomDirection()**: Generează aleator o direcție pentru șarpe: stânga (`L`), dreapta (`R`), sus (`U`), jos (`D`).

- **Metoda startGame():** Inițializează un nou joc apelând metoda initializeGame(), generează un măr nou, porneste timer-ul pentru a gestiona logica jocului si ascunde butonul de start.
- **Metoda paintComponent(Graphics G):** Desenează elementele grafice pe panou.
- **Metoda draw(Graphics G):** Desenează elemente precum mărul și șarpele pe panou în funcție de starea jocului.
- **Metoda newApple():** Generează aleator poziția unui măr pe ecran.
- **Metoda increaseSnakeSpeed():** Crește viteza șarpelui după fiecare 5 mere consumate.
- **Metoda checkApple():** Verifică dacă șarpele a consumat un măr și actualizează numărul de mere mancate. De asemenea apeleaza condiția de câștig atunci când sunt consumate 50 de mere.
- **Metoda move():** Actualizează poziția șarpelui în funcție de direcția curentă.
- **Metoda checkCollisions():** Verifică coliziunile cu propriul corp sau cu marginile ecranului si actualizează scorul maxim și afișează butonul de start dacă jocul s-a încheiat.

- **Metoda actionPerformed(ActionEvent e):** - Legată la timer, controlează logica jocului la fiecare eveniment.
- **Metoda Pause():** Oprește sau reia jocul în funcție de starea actuală a tastei 'p'.
- **Metoda getDirection():** Returnează direcția curentă a șarpelui.
- **Metoda setDirection(char direction):** Setează direcția șarpelui în funcție de input.
- **Metoda main(String[] args):** Metodă principală pentru a crea și afișa fereastra jocului. (Folosita in toate cele 4 clase)

FUNCTIONALITATI PROIECT - SCOPUL CLASELOR - CLASA GAMEFRAME:

- GameFrame extinde JFrame, ceea ce înseamnă că reprezintă o fereastră a unei aplicații Java Swing. Constructorul public GameFrame() este apelat atunci când se creează o instanță a clasei, iar instanța clasei GUI numită `panel`, reprezintă panoul principal al jocului Snake.
- Constructorul GameFrame: Adaugă panoul de joc (`GUI`) la fereastra principală, setează titlul ferestrei la "Snake", afișează fereastra în centrul ecranului și o face vizibilă. De asemenea, metoda pack() ajustează dimensiunile ferestrei astfel încât să încadreze conținutul și metoda setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE) stabilește acțiunea de închidere a ferestrei când este apăsat butonul "X" pentru închidere.

FUNCTIONALITATI PROIECT - SCOPUL CLASELOR - STARTBUTTONPANEL SI MYKEYADAPTER:

- **1.1.Constructorul public StartButtonPanel** primește o referință la obiectul GUI si crează butonul de "Start" de o anumita dimensiune.
- **1.2.Metoda startButton.addActionListener(...):** La apăsarea butonului, se apelează metoda **startGame()** pentru a începe jocul, iar panoul curent este făcut invizibil, astfel încât butonul să dispară din ecran în timpul jocului.
- **2.1.Constructorul public MyKeyAdapter** primește o referință la obiectul GUI.
- **2.2. Metoda keyPressed(KeyEvent e):** Detectează tasta apăsată utilizând obiectul **e**, iar pentru tastele săgeți stânga, dreapta, sus și jos, schimbă direcția șarpelui si pentru tasta 'P', activează sau dezactivează pauza în joc, apelând metoda **Pause()**.

REGULILE JOCULUI

Jocul Snake este un joc clasic în care jucătorul controlează un șarpe care se mișcă pe o tablă de joc. Scopul este să controlezi șarpele pentru a mânca merele care apar aleator pe ecran și să eviți să te ciocnești de pereții tablei de joc sau de propriul tău corp.

Pe măsură ce mănânci mere, șarpele tău crește în lungime, iar obiectivul este să obții un scor cât mai mare.

- **Reguli de bază ale jocului:**

1. **Mâncarea Merele:** Merele apar pe tablă în locații aleatorii. Atunci când șarpele mănâncă un măr, acesta crește în lungime.
2. **Mișcarea Șarpelui:** Jucătorul controlează direcția șarpelui, care poate fi sus, jos, stânga sau dreapta. Șarpele se mișcă în direcția specificată de jucător și continuă să se deplaseze în acea direcție până când jucătorul îi schimbă direcția.
3. **Evitarea Coliziunilor:** Scopul este să eviți coliziunile cu pereții tablei de joc și cu propriul corp. Dacă șarpele se ciocnește cu pereții sau cu el însuși, jocul se încheie.
4. **Scorul:** Scorul este determinat de numărul de mere mâncate. Scopul este să obții un scor cât mai mare.
5. **Victorie:** În implementarea specifică a codului furnizat, există și un element de victorie. Jucătorul câștigă dacă mănâncă un număr specific de mere (în acest caz, 50).

POSSIBLE IMPROVEMENTS:

- **Design Grafic îmbunătățit:** Îmbunătățirea aspectului vizual al jocului prin adăugarea unor imagini sau animații pentru șarpe, măr sau labirint.
- **Nivele de Dificultate:** Implementarea nivelelor de dificultate crescătoare pe măsură ce jocul avansează (mai multe labirinturi).
- **Mod Multiplayer:** Adăugarea unui mod multiplayer pentru a permite mai multor jucători să joace simultan și să concureze între ei.

