

# **Miniproiect Java**

## **Connect4**

**Zoican Denis, 30223**

**An academic: 2019-2020**

- **Introducere**

Pentru proiect am avut de implementat jocul Connect4 folosind Swing si MySQL. Jocul este multiplayer, acesta putand fi jucat de 2 persoane in acelasi timp, precum si singleplayer, jucand contra unui bot.

Regulile jocului sunt simple: primul jucator ce reuseste sa plaseze 4 piese de ale sale in orizontala, verticala sau diagonala va castiga. Pentru a spori perspicacitatea jucatorilor, am implementat un timer in partea stanga care va arunca piesa automat daca jucatorul nu face o alegere la timp. Cu scopul de atrena atentia jucatorilor, am implementat si regula de a nu pozitiona o piesa in afara tablei (pe pozitia -1 a matricei).

Meciurile anterioare sunt salvate intr-o baza de date locala unde sunt retinute numele jucatorului care a castigat, precum si numele celui care a pierdut. In timpul sesiunii, retinem meciurile care au avut loc.

Puteti personaliza jetonul cu care veti juca in momentul alegerii numelui.

- **MVC**

Proiectul este structurat in 3 pachete:

- i. controllers
- ii. views
- iii. frames

Clasa MainApp este aceea pe care o folosim sa instatiem tot jocul prin intermediul metodei main.

- **Controllers**

- a. MainController**

In aceasta clasa setam Listener-urile pentru butonul Start prin care jocul va incepe. Acest listener va analiza numele oferit jucatorilor si va alege jetonul potrivit. Dupa setarea jetoanelor, jocul va incepe prin pornirea timer-ului din MainFrame.

In constructor realizam interogari initiale din MySQL pentru a afisa jocurile recente. Aceste informatii vor fi procesate si adaugate in panel-ul RightSqlHistory .

Metoda change player turn are rolul de reseta animatia si de a schimba numele jucatorului.

Metoda remove panels matrix este folosita pentru a elimina toate panel-urile din PanelMatrix.

Metoda end game opreste jocul si reseteaza matricea si restul panel-urilor.

Metoda add panels matrix adauga panel-urile in MatrixPanel.

Metoda init panels initializeaza panel-urile.

Metoda drop token este apelata cand dorim sa “aruncam” jetonul in tebla, modificand matricea de valori si schimband jucatorul. Dupa ce piesa a fost plasata, verificam daca jucatorul respectiv a castigat.

Metoda change token modifica pozitia jetonului inainte de a fi aruncat.

OverAnimation este un timer care este pornit dupa oprirea jocului pentru a astepta 5 secunda cu scopul de a vizualiza miscarea castigatoare.

KeyAdapter este folosit pentru a raspunde la apasatul tastelor. Acestea vor modifica pozitia jetonului si vor declansa plasarea acestuia.

## **b. BotController**

Aceasta clasa este folosita pentru a ajuta bot-ul sa aleaga mutarea cea mai buna. In procesul de selectare am folosit algoritmul "MinMax". Metoda principala este chooseBestMove care ne va da un obiect Point cu pozitia cea mai buna.

## **c. LimitText**

Este o clasa folosita pentru JTextField pentru a limita numarul de caractere.

## **d. TimerDrop**

Este clasa ce ne ofera timerul pentru animatie. Aceasta va cobora numele jucatorului care trebuie sa mute la un interval de 5 secunde.

## **e. ConnectionSQL**

Folosita pentru a ne conecta la baza de date locala.

- **Models**

### **a. MainModel**

Clasa in care initializam matricea de valori matVal. O matrice folosita pentru a memora pozitiile pieselor pozitionate.

### **b. NodeMatrix**

Reprezinta fiecare nod al matricei. Instantele acestei clase vor retine valoarea , player-ul, imaginea jetonului, numarul liniei si numarul coloanei.

### **c. Point**

Clasa folosita in BotController pentru a retine pozitia favorabila.

- **Views**

- a. MainFrame**

- Frame-ul in care se afla toate instantele ce sunt folosite pentru interfata grafica. Frame-ul este impartit in 3 parti principale:

- LeftPanel
    - RightPanel
    - MiddlePanel

- LeftPanel contine animatia si cele 2 JTextField-uri pentru nume.

- MiddlePanel prezinta matricea de valori.

- RightPanel este alcatuita din panel-urile cu meciurile salvate in baza de date.

- b. Button**

- Este clasa pentru butonul de START prin intermediul caruia vom incepe jocul.

- c. HistoryDate**

- Este un panel folosit pentru RightPanel prin care afisam data cand au avut loc jocurile.

- d. HistoryLabel**

- Este un panel folosit pentru RightPanel prin care afisam informatiile despre meci.

- e. PlayerLabel**

- Este un panel pe care il folosim ca sa inseram numele jucatorilor.

- f. TokenImage**

- Este un panel ce contine imaginea jetonului pe care il salvam in amtricea de valori. Fiecare token are ca parametru un String pentru a stabili ce jeton folosim.

