

Semestrální práce Šachy

Třída Chess_SP_2023

V této třídě je vytvořené dvojrozměrné pole, které zastupuje šachovnici. Dále je zde metoda makeMove, undoMove, isContainsWords.

Metoda makeMove(String move) , která pomocí parametru move“x1y1x2y2OldPiece“, význam vysvětlen níže, překreslí figurku na pozici kam ji hráč chce přesunout funguje následovně. charAt 2 a 3 zastupují x2y2, charAt 0 a 1 zastupují x1y1. Metoda vezme pozici kam se má figurka posunout (x2y2) a napíše do ní to co bylo v počáteční pozici (x1y1). Následně jen počáteční pozici (x1y1) přepíše na “ “. Pokud uvedeme příklad: Hráč klikne na pěšáka na pozici 06 a posune ho na pozici 05. metoda makeMove dostane parametr String move ve tvaru “0605 “. Následně se tedy cokoliv co bylo na pozici 06 přepíše na pozici 05, a pozice 06 se přepíše na “ “. Zároveň je zde kontrolováno zda se pohnul král, věže a pěšáci, kvůli rošádě a braní mimochodem.

Metoda undoMove(String move), která pomocí parametru move“x1y1x2y2OldPiece“, vrátí tah, který byl vykonán zpět. Metoda vezme cokoliv co je na pozici x2, y2 a vrátí to na pozici x1y1, s tím, že na pozici x2y2 vloží String OldPiece, který se tam předtím nacházel.

Metoda isContainsWords(String[] words, String Target, boolean player) prohlíží pole, ve kterém jsou případy možných písmen, a pokud se písmeno na pozici x1y1 rovná nějakému z možných čísel v poli, prohodí hodnotu na false. Tato metoda se stará o střídání se hráčů.

```
static String chessBoard[][] = {  
  
    {"r", "k", "b", "q", "a", "b", "k", "r"},  
  
    {"p", "p", "p", "p", "p", "p", "p", "p"},  
  
    {" ", " ", " ", " ", " ", " ", " ", " "},  
  
    {" ", " ", " ", " ", " ", " ", " ", " "},  
  
    {" ", " ", " ", " ", " ", " ", " ", " "},  
  
    {" ", " ", " ", " ", " ", " ", " ", " "},
```

`{"P", "P", "P", "P", "P", "P", "P", "P"},`

`{"R", "K", "B", "Q", "A", "B", "K", "R"};`

Malá písmena reprezentují figurky černého hráče, zatímco velká písmena reprezentují figurky bílého hráče.

Co které písmenko znamená:

- Pěšák = P/p
- Kůň = K/k
- Střelec = B/b
- Věž = R/r
- Královna = Q/q
- Král = A/a

Třída Sachovnice_a_sachy

Třída dědí od JPanel a implementuje MouseListener a MouseMotionListener.

V této třídě se vykresluje šachovnice a šachy a je zde vytvořen Drag and drop pohyb figurkami pomocí myši. Dále jsou v této třídě vypočítány všechny potřebné hodnoty pro scaling při zvětšování a zmenšování okna. (double scale, chessBoxX, chessBoxY, chessboxPositionX, chessboxPositinY.

Šachovnice se vykresluje pomocí dvou for cyklů, instance boolean white po každém projetí změní hodnotu, tak aby políčka měnila barvu. Jsou zde i vytvořena tlačítka pro funkce vrácení tahu, tahu dopředu, zapnutí UI jako černý hráč , přepnutí zpět na černého jako hráče, exportování do PNG, rozkliknutí grafu a resetování hry.

Jednotlivé figurky jsou pak vykreslovány pomocí Path2D a Graphics2D. Metoda switch projíždí pole chessBoard a pro určité případy vytváří instanci správné figurky. Pokud je figurka animována, je volán jiný, přetížený konstruktor, který vytvoří instanci figurky pouze pro animaci.

V této třídě se dále nachází metody pripravAnimaci, lastmove, saveImage, MATEvent, moveMaker, UIplay.

Metoda pripravAnimaci(int orgX, int orgY, int toX, int toY) se stara o spravne casovani animace, je zde pocitan počet opakovani kolikrat se bude volat okno, resp. Kolikrat se bude dit repaint.

Metoda lastmove(Graphics g2) vykresluje na sachovnici, který pohyb se vykonal jako poslední.

Metoda saveImage() uklada momentalni situaci na sachovnici do PNG formatu.

Metoda MATEvent(JFrame okno) rozhoduje o výsledku partie, pokud je jeden z králů v šachu a nebo hráč nemá žádný možný pohyb, na obrazovce se ukáže výsledek s možností buď hru restartovat nebo hru úplně vypnout. (bohužel v některých fázích začne blbnout a okno se tam ukáže vícekrát, neznám důvod).

Metoda moveMaker(Graphics2D g2, String moveP1, double scale, double chessBoxX, double chessBoxPositionX) vykresluje na šachovnici možný pohyb figurky, na kterou se zrovna kliknulo.

```
for (int i = 0; i < 64; i++) {  
    switch (Chess_SP_2023.chessBoard[i / 8][i % 8]) {  
  
        case "P" -> {  
            //Vykresli pesaka pro bileho hrace.  
  
            Pawn P = new Pawn(g2,  
                chessBoxX, chessboxPositionX,  
                chessboxPositionY, i, scale,  
                Color.BLACK, Color.WHITE);  
  
            .  
            .  
            .  
        }  
    }  
}
```

Pokud se jedná o animující se figurku, vykreslí se na orgX a orgY nová instance konkrétní figurky a vykoná animaci. Po skončení animace se opět pomocí switch vykreslí na správné pozici figurka.

O pohybování s figurkami je postaráno pomocí metod mousePressed a mouseReleased. Metoda mousePressed získává souřadnice myši na šachovnici ve chvíli, kdy uživatel stiskne tlačítko myši. Metoda mouseReleased získává souřadnice myši na šachovnici ve chvíli, kdy uživatel pustí tlačítko myši, pokud hráč kliknul na figurku a jedná se o možný pohyb, překreslí figurku na místo kde hráč pustil tlačítko myši.

```
if (ev.getButton()==MouseEvent.BUTTON1) {  
    String dragMove;  
  
    dragMove=""+(int)(mouseY/chessBoxY)+(int)(mouseX/chessBoxX)+(int)(newMouseY/ch  
    essBoxY)+(int)(newMouseX/chessBoxX)+Chess_SP_2023.chessBoard[(int)(newMouseY/c  
    hessBoxY)][(int)(newMouseX/chessBoxX)];  
  
    String userPosibilities=PiecesMovement.possibleMoves();  
  
    //Jestlize je to mozny pohyb, tak posune figurku, pokud neni, nic se nestane.  
    if(userPosibilities.replaceAll(dragMove, "").length() <  
        userPosibilities.length()) {  
        PiecesMovement.makeMove(dragMove);  
    }  
}
```

Třída PiecesMovement

Tato třída slouží pro vytvoření možných pohybů s figurkami.

Třída dědí od třídy Chess_SP_2023, aby mohla pracovat s polem chessBoard.

Metoda possibleMoves() projíždí pomocí for pole chessBoard a zároveň pomocí switch pro jednotlivé případy/figurky ukládá do Stringu list možné pohyby figurek. Vrací pak String všech možných pohybů ve formátu x1y1x2y2C... x1,y1 počáteční pozice – x2,y2 pozice kam bude figurka přesunuta – C jaká figurka bude sebrána.

```
public static String possibleMoves(){
    String list = " ";
    for(int i = 0; i < 64; i++){
        switch(chessBoard[i/8][i%8]){
            case "P":
                list += possibleP(i);
                break;
            case "R":
                list += possibleR(i);
                break;
            .
            .
            .
        }
    }
}
```

Ve třídě nachází metody pro vytvoření možných pohybů jednotlivých figurek, jak pro bílého, tak pro černého hráče. Každá metoda vrací String možných pohybů. Zjednodušeně každá metoda projíždí pole chessBoard, a pokud je splněna podmínka, že je index na této pozici roven " " nebo malému/velkému písmenku, tak vrátí String "x1y1x2y2C".

Dále se zde nachází metody kingSafeC a kingSafeL, které hlídají, jestli nějaká z figurek neútočí na krále, pokud ano, jsou povoleny jen pohyby, které zabrání útoku.

Třída Graf

V této třídě se vytváří graf pro to jak trval hráči jeden pohyb. Jsou zde dva ArrayListy, do kterých se ukládají časy jednotlivých kol. Tyto časy jsou pak předávány do metody createBarChart(), ve které se pak vykresluje celý graf.

Třídy Pawn, Rook, Bishop, Knight, King, Queen

V těchto třídách se nachází vždy přetížený konstruktor. Jeden, který vykresluje figurku, pokud není animovaná. A druhý, který vykresluje figurku v animaci.

Třída Main

Třída main je hlavní třídou celého projektu. Je zde vytvořené JFrame okno, na kterém se vše děje. Dále je tu kontrolováno, jestli je uložená pozice králů aktuální, pokud ne, aktualizuje ji.

Na čem je třeba pracovat

Určitě by to chtělo pořádně udělat šach mat, který občas nefunguje jak má(zejména pak u černého krále). Bohužel brání mimochodem nefunguje přesně jak má, nepodařilo se mi vytvořit takové podmínky, které by tuto situaci lépe ošetřily.

Mimo dokumentaci

Vytváření animace pro moje šachy bylo asi horší než zkouška z LAA.

KIV/UPG
2022/2023

Filip Chlad - A22B0060P
Celkový počet hodin strávený na řešení SP
dosud: **Neznámo**

28. May
2023

KIV/UPG
2022/2023

Filip Chlad - A22B0060P
Celkový počet hodin strávený na řešení SP
dosud: **Neznámo**

28. May
2023