Gymnázium, Praha 6, Arabská 14
Arabská, Praha 6, 160 00

ROČNÍKOVÝ PROJEKT

Předmět: Programování
Téma: Společenská hra Solitaire

Autor: Nikola Jankovič
Třída: 1.E
Školní rok: 2021/2022
Vyučující: Mgr. Jan Lána
Třídní učitel: Mgr. Blanka Hniličková

Čestné prohlášení:

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne: .....

Nikola Jankovič

Anotace:

Pro tuto ročníkovou práci jsem si připravil program v programovacím jazyce Java. Program je desková hra s názvem Solitaire (nezaměnit se stejnojmennou karetní hrou). Hra funguje na podobném principu jako populární desková hra dáma. Solitaire se na rozdíl od dámy hraje pouze v jednom hráči. Hráč má za úkol postupně přeskakovat žetony dokud nezbyde na desce pouze jeden žeton.

Annotation (English):

For this coursework I have prepared a programme in the Java programming language. The programme is a board game named Solitaire (different from the card game Solitaire). The game follows a similar principle to checkers. The main difference between the two games is that Solitaire is only played by one player. The player's goal is to remove pegs one by one until there is only one peg left on the board.

Zadání ročníkového projektu:

Vytvořte hru Solitaire, kde se při zapnutí objeví žetony v předurčeném rozhraní. Hráč pak bude moci přetahovat kurzorem jednotlivé žetony a postupným přeskakováním žetony vyhazovat dokud na desce nezbude jenom jeden žeton.

Možné vylepšení do budoucna:

- Přidat novou scénu pokud hráč vyhraje nebo prohraje
- Změnit vzhled rohů hracího pole, protože se v nich nemohou nacházet žetony
- Zlepšit vzhled žetonů

Platforma:

- Java
- JavaFX

Obsah

Úvod

Tato dokumentace pojednává o hře Solitaire napsané v programovacím jazyce Java. Cílem hry je přeskakovat postupně přeskakovat žetony, dokud nezbyde jenom jeden.

Toto téma mi zvolil pan profesor Mgr. Jan Lána, jelikož jsem v době vybírání témat nebyl ve škole, z důvodu vážného úrazu pravé spodní končetiny. Jsem názoru, že téma mi vybral dobré a přiměřeně obtížné.

Program jsem napsal v programovacím jazyce Java a vývojovém prostředí IntelliJ IDEA, který používáme i na hodinách Programování.

1. Programovací jazyk a vývojové prostředí

Pro napsání předkládaného programu jsem použil Javu 11 a vývojové prostředí IntelliJ IDEA.

1.1. Java

Java je programovací jazyk vyvinutý firmou Sun Microsystems v roce 1995. Do roku 2020 byla Java nejpoužívanější programovací jazyk v roce 2020 jí předběhl Python a C. Velkou výhodou Javy je dostupnost. Aplikace vytvořené v programovacím jazyce Java se dají bez problému spustit na jakémkoliv zařízení bez ohledu na operační systém zařízení.

Obrázek 1 - Duke maskot Javy

1.2. JavaFX

JavaFX je rozšíření Javy které umožňuje tvoření grafických aplikací. FX je zkratka značící slovo effects.

1.3. IntelliJ IDEA

IntelliJ IDEA je komerční vývojové prostředí vynalezeno firmou JetBrains. Zároveň existuje bezplatná Community verze. Tato verze je open-source a využil jsem jí k napsání mého kódu.

Obrázek 2 - Logo IntelliJ IDEA

2. Solitér

2.1. Solitér a Pasiáns

Pod hrou Solitér (v překladu samotář) si většina lidí představí karetní hru pro jednoho hráče. Tato hra se oficiálně nazývá Pasians (v překladu trpělivost). Název solitér se v USA a používá pro spoustu karetních her pro jednoho hráče.

2.2. Solitér

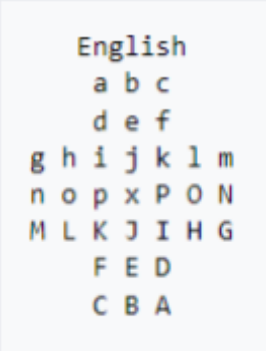
Cílem hry solitér je postupně legálními tahy vyřazovat jednotlivé žetony, dokud na desce nezbyde pouze jeden žeton. Tradičně existují dvě počáteční desky, které se liší v rozmístění žetonů. Anglická a evropská deska (viz. obrázek 3).

Obrázek 3 - Anglická deska (nalevo) Evropská (napravo)

Hra se dá hrát jak s diagonálními tahy tak bez. V mé verzi hry používám anglickou desku a diagonální tahy jsou legální. V případě evropské desky s počátečním volným polem uprostřed se hra nedá bez diagonálních tahů vyhrát.

2.3. Strategie

Existuje mnoho různých řešení standardního problému a jeden zápis používaný k jejich popisu přiřazuje dírám písmena(viz. obrázek 4):



Obrázek 4 - popsání jednotlivých polí

*Na následujících obrázcích · označuje žeton v díře, zvýrazněná * označuje žeton, který má být přemístěn, a o označuje prázdné pole. Modrá ¤ je prázdné pole, ze kterého se aktuální žeton přesunul; červená * je konečná pozice tohoto žetonu, červené o je volné pole po žetonu, který byl přeskočen a odstraněn.*[1]

Obrázek 5 - Nejkratší možné řešení anglické desky

Toto řešení bylo nalezeno v roce 1912 Ernestem Bergholtem a v roce 1964 John Beasley dokázal, že je to nejkratší možné řešení.[1]

3. Program

3.1. Konstanty

Na začátku programu jsem vytvořil třídu constants. V této třídě jsou hodnoty, které jsou poté používány v celém programu. Zjistil, že je to docela praktické řešení a nemusím díky tomu hledat proměnné, pokud bych je chtěl upravit.

```
public final class constants {
    public static final int[][] englishVersion = new int[][]{
            {0, 0, 1, 1, 1, 0, 0},
            {0, 0, 1, 1, 1, 0, 0},
            {1, 1, 1, 1, 1, 1, 1},
            {1, 1, 1, 2, 1, 1, 1},
            {1, 1, 1, 1, 1, 1, 1},
            {0, 0, 1, 1, 1, 0, 0},
            {0, 0, 1, 1, 1, 0, 0}
    };
    public static final int[][] frenchVersion = new int[][]{
            {0, 0, 2, 1, 1, 0, 0},
            {0, 1, 1, 1, 1, 1, 0},
            {1, 1, 1, 1, 1, 1, 1},
            {1, 1, 1, 1, 1, 1, 1},
            {1, 1, 1, 1, 1, 1, 1},
            {0, 1, 1, 1, 1, 1, 0},
            {0, 0, 1, 1, 1, 0, 0}
    };
```

Obrázek 6 - číselná interpretace desky

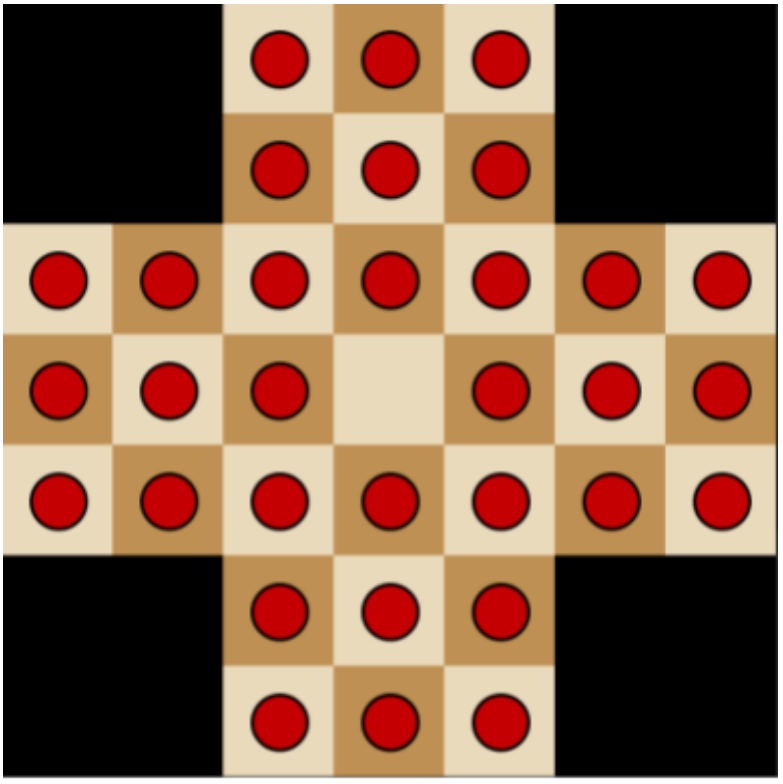
Asi nejzajímavější část této třídy jsou 2 pole (viz. obrázek 6). Fungují jako číselná interpretace herní desky. 0 značí mrtvé dlaždice, které se nedají využít. 1 značí dlaždice s žetonem. 2 značí samostatnou dlaždici bez žetonu.

3.2. Metoda placeTiles

metoda placeTiles (viz. obrázek 7) využívá hodnot z výše zmíněných polí a vytváří pomocí těchto hodnot výchozí herní plochu (viz. obrázek 8)

```java
private void placeTiles(int[][] gameLayout) {
    for (int y = 0; y < height; ++y) {
        for (int x = 0; x < width; ++x) {
            Tile tile = null;
            Peg peg = null;
            // 0 jsou mrtvé dlaždice, 2 jsou prázdné dlaždice
            if (gameLayout[y][x] == 0 || gameLayout[y][x] == 2) {
                tile = new Tile(gameLayout[y][x], x, y);
                // 1 jsou dlaždice s žetony
            } else if (gameLayout[y][x] == 1) {
                tile = new Tile(gameLayout[y][x], x, y);
                peg = pohyb(x, y);
            }
            board[x][y] = tile;
            tileGroup.getChildren().add(tile);
            if (peg != null) {
                tile.setPeg(peg);
                pegGroup.getChildren().add(peg);
            }
        }
    }
```

Obrázek 7 - metoda placeTiles



Obrázek 8 - výchozí deska

3.3. Pohyb žetonů

Nejdůležitější část pohybu je metoda legalMove(viz Obrázek 9). Tato metoda kontroluje podle souřadnic, zda je pohyb, o který se hráč snaží, legální. Využívá spousta podmínek, které kontrolují, zda hráč nepřetahuje žeton mimo hrací pole, zda nepřetahuje žeton na dlaždici s jiným žetonem, zda nepřetahuje žeton moc daleko, kontroluje směr přetažení žetonu a zda vůbec přeskakujeme nějaký žeton.

```
// kontroluje jakýkoliv pohyb zda je legální
private boolean legalMove(int x0, int y0, int x1, int y1) {

    // zkontroluje, zda hráč nepřetahuje žeton mimo hrací pole pomocí booleanu OB
    if (OB(x1, y1)){
        System.out.println("Out of Bounds");
        return false;
    }
    // zkontroluje, zda dlaždice, na kterou chceme žeton přetáhnout, už nemá žeton
    if (board[x1][y1].hasPeg()) {
        System.out.println("Dlaždice na souřadnicích " + x1 + " " + y1 + "je zabraná");
        return false;
    }
    // zkontroluje, zda hráč nepřetahuje žeton moc daleko
    if (Math.abs(x0 - x1) > 2 || Math.abs(y0 - y1) > 2) {
        System.out.println("Moc daleko");
        return false;
    }

    // Zkontroluje, zda hráč přetahuje žeton ve správném směru (horizontální, vertikální nebo diagonální)
    if (Math.abs(x0 - x1) == 1 || Math.abs(y0 - y1) == 1) {
        System.out.println("Špatný směr");
        return false;
    }
    // Kontroluje, zda hráč přeskakuje nějaký žeton
    System.out.println("Přeskakuje žeton = " + board[(x0 + x1) / 2][(y0 + y1) / 2].hasPeg());
    return board[(x0 + x1) / 2][(y0 + y1) / 2].hasPeg();
}
```

Obrázek 9 - metoda legalMove

Metodu legalMove poté využívám v metodě pohyb, která vytváří žetony, získává staré a nové souřadnice přetahovaného žetonu. Dělá samotný pohyb (pokud je metoda legalMove splněna).

Závěr

V tomto ročníkovém projektu jsem vytvořil deskovou hru Solitér. Nakonec jsem rád, že jsem musel ročníkovou práci předělat, protože za těch pár hodin předělávání jsem se o programování v jazyce Java naučil mnohé věci, které se mi budou hodit do příštích let. S předělaným kódem jsem velmi spokojen a jsem přesvědčen, že mu dostatečně rozumím.

Citace

1. Peg solitaire - Wikipedia. [online]. Dostupné z: https://en.wikipedia.org/wiki/Peg_solitaire

Seznam obrázků