

Reversi

Jakub Kulhan

1 Zadání

Hra pro dva hráče na jednom počítači, program zobrazí hrací pole s obsazenými pozicemi (rozdílitelnými podle hráče), v každém tahu hráč na výzvu zadá pozici, kterou chce obsadit (např. kartézskými souřadnicemi).

Program kontroluje dodržování pravidel¹. Nedovolí nepřipustné tahy. Podle pravidel také provádí automaticky některé akce.

Možnost uložení a načtení rozehrané hry. Také možnost hry proti počítači, který bude hrát náhodnými tahy nebo jednoduchou umělou inteligencí.

2 Řešení

2.1 Pravidla hry Reversi

Reversi se hraje na čtvercové hrací desce 8×8 barevně nerozlišených políček. Obvykle se políčka označují písmeny A-H pro sloupce a čísla 1-8 pro řádky. Hraje se s 64 plochými kameny, které jsou z jedné strany bílé a z druhé černé. V počáteční pozici jsou na políčkách E4 a D5 černé kameny a na políčkách D4 a E5 bílé kameny. Hru začíná hráč hrající černými kameny.

Umístěním kamene hráč otočí soupeřovy kameny ve všech osmi směrech (horizontálně, vertikálně i po obou diagonálách) od umístěného kamene, pokud je řada kamenů ukončena hráčovým kamenem. Hráč na tahu musí vždy svůj kámen umístit tak, aby otočil alespoň jeden soupeřův kámen. Neexistuje-li prázdné políčko, kam by hráč mohl umístit svůj kámen, aby tuto podmínku splnil, dává pas a hraje protihráč. Existuje-li však takové pole, hráč tam svůj kámen musí umístit, i kdyby to pro něj bylo nevýhodné.

Hra končí, byla-li celá plocha zaplněna kameny anebo nemůže-li ani jeden z hráčů umístit svůj kámen. Vyhrává ten hráč, který má na ploše více kamenů své barvy.

2.2 Model-View-Controller

Pro daný typ zadání se přímo nabízí využít spíše grafické rozhraní než prostředí terminálu. Proto jsem i tak zvolil. Existuje více ověřených postupů řešení (návrhových vzorů) při programování grafických aplikací. Asi nejpoužívanější je Model-View-Controller² (dále jen MVC).

MVC rozděluje aplikaci na tři části – model, view a controller. Model, to jsou data aplikace. View se stará o zobrazování těchto dat. A controller obsluhuje vstup uživatele a podle toho mění model anebo posílá view příkazy, jak a co má vykreslovat. Také existuje nepřímé spojení mezi modelem a view realizované podle vzoru Observer³ - view se zaregistruje u modelu a ten mu při změně pošle zprávu, view se si vytáhne data z modelu a podle nich se aktualizuje.

2.3 Implementace

Aplikace je rozdělena do tří balíčků:

- `info.kulhan.reversi.model`,
- `info.kulhan.reversi.view`
- a `info.kulhan.reversi.controller`.

¹ <http://en.wikipedia.org/wiki/Reversi>

² <http://martinfowler.com/eaCatalog/modelViewController.html>

³ http://en.wikipedia.org/wiki/Observer_pattern

Každý balíček obsahuje třídy specifické pro danou část MVC triády.

Hlavní třídou modelu je **GameState**. Ta obsahuje celý stav hry:

```
public class GameState extends Observable implements Serializable, Observer {
    private Player currentPlayer;

    private Board board;

    public enum State { INTERACTIVE, ENDED, WAITING }

    private State state;

    public enum Type { ONE_PLAYER, TWO_PLAYER }

    private Type type;

    // ...
}

public enum Player implements Serializable { NONE, BLACK, WHITE }

public class Board extends Observable implements Serializable, Iterable<BoardSquare> {
    private Player data[][];

    // ...
}

public class BoardSquare {
    private int row;

    private int column;

    private Player player;

    // ...
}
```

Jednotlivé vlastnosti objektů mají své gettery, popř. i settery. Pokud je vytvořen nový prázdný stav hry, hrací pole (**Board**) je inicializováno do stavu nové hry (tj. všude je **NONE** krom polí uprostřed, které jsou nastaveny na **BLACK**, resp. **WHITE**) a **currentPlayer** je **BLACK**. Celý **GameState** je **Serializable** pro jednoduché ukládání/načítání hry.

View využívá grafické knihovny Java Swing⁴. Vše se odehrává v jednom okně (v jednom **JFrame**) , ale to je tvořeno z více panelů (**JPanel**):

- **MenuPanel** obsahuje tlačítka pro novou hru, nahrání uložené pozice a její uložení;
- **BoardPanel** se stará o vykreslení hrací plochy
- a **MovePanel** zobrazuje informace o současném tahu.

Všechny akce jsou delegovány controlleru (třída **Controller**), např. po stisknutí tlačítka *Nová hra* v **MenuPanelu** se zavolá metoda **newGame()** controlleru.

⁴ <http://docs.oracle.com/javase/7/docs/api/javafx/swing/package-summary.html>

3 Ovládání programu

3.1 Kompilace

Ke spuštění programu je potřeba vybuildovat soubor `dist/Reversi.jar`. Aplikace využívá sestavovací systém Ant⁵, stačí tedy v adresáři aplikace spustit:

```
$ ant
```

Popř. lze adresář aplikace otevřít jako NetBeans⁶ projekt a použít integrovanou funkci pro build.

3.2 Spouštění

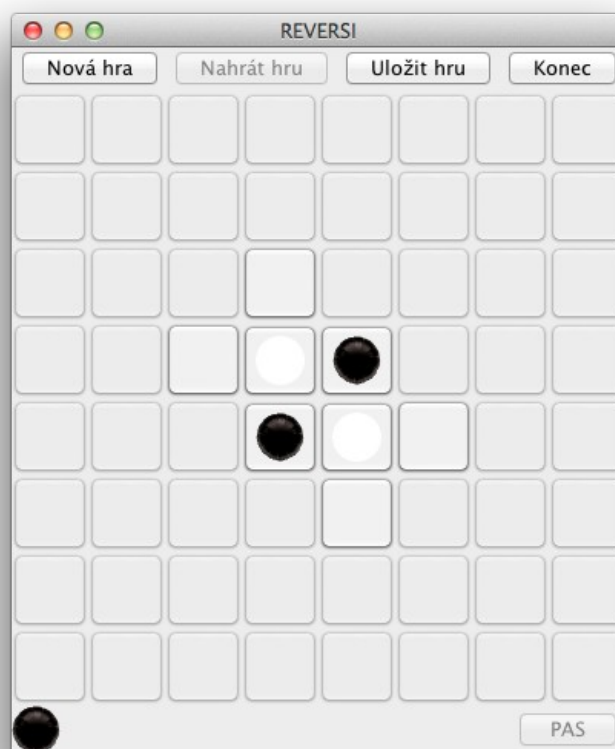
Spustit lze v terminálu v adresáři projektu příkazem:

```
$ java -jar dist/Reversi.jar
```

Nebo lze adresář aplikace otevřít jako NetBeans projekt a spustit pomocí Run.

3.3 Aplikace

Po spuštění aplikace začne novou hru pro dva hráče na jednom počítači (viz screenshot 1).

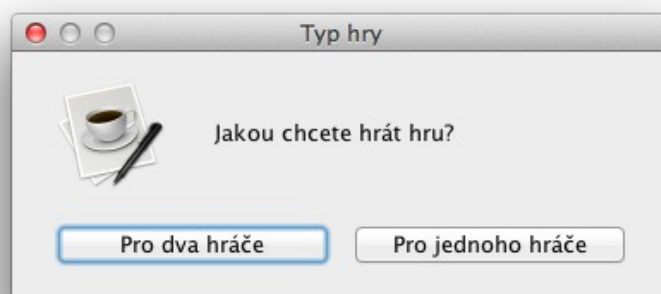


Screenshot 1: Po prvním spuštění

⁵ <http://ant.apache.org/>

⁶ <http://netbeans.org/>

Kliknutí na tlačítko *Nová hra* vyvolá dialog pro výběr hry (viz screenshot 2).



Screenshot 2: Nová hra

Při výběru *Pro dva hráče* se vytvoří nová hra pro dva hráče na jednom počítači. *Pro jednoho hráče* spustí hru proti počítačovému protivníkovi.

Pokud není k dispozici uložená pozice, tlačítko *Nahrát hru* je neaktivní. Jinak se po kliknutí na něj zahodí současná rozehraná hra a nahraje se uložená pozice.

Naopak tlačítko *Uložit hru* současnou hru uloží do souboru (konkrétně do domovského adresáře uživatele pod názvem `.reversi.sav`).

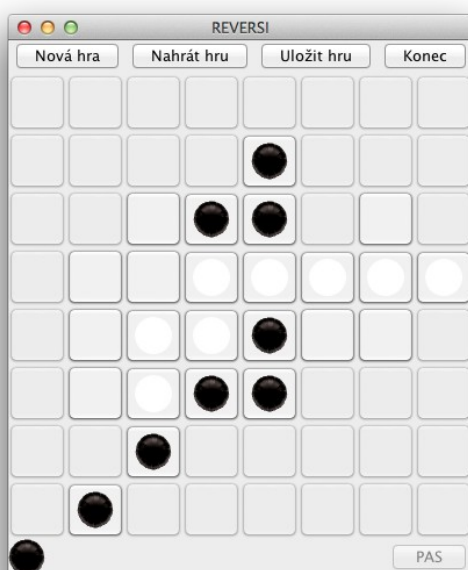
Tlačítko *Konec* ukončí program.

Při hře se možné tahy zobrazují jako aktivní tlačítka na hracím poli (viz screenshot 3).

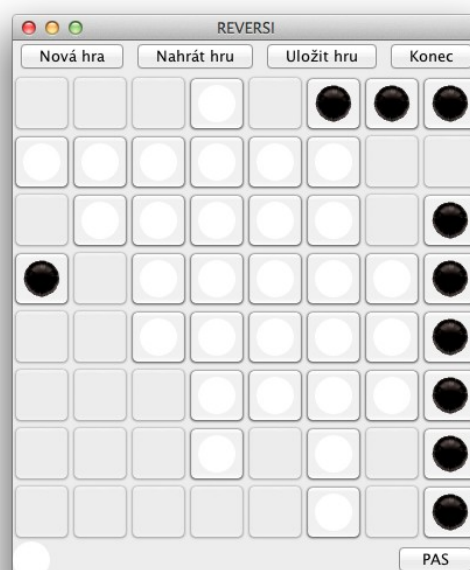
Dole vlevo se zobrazuje, který hráč je zrovna na tahu.

Pokud hráč nemůže hrát, tlačítko *PAS* se stane aktivní (viz screenshot 4).

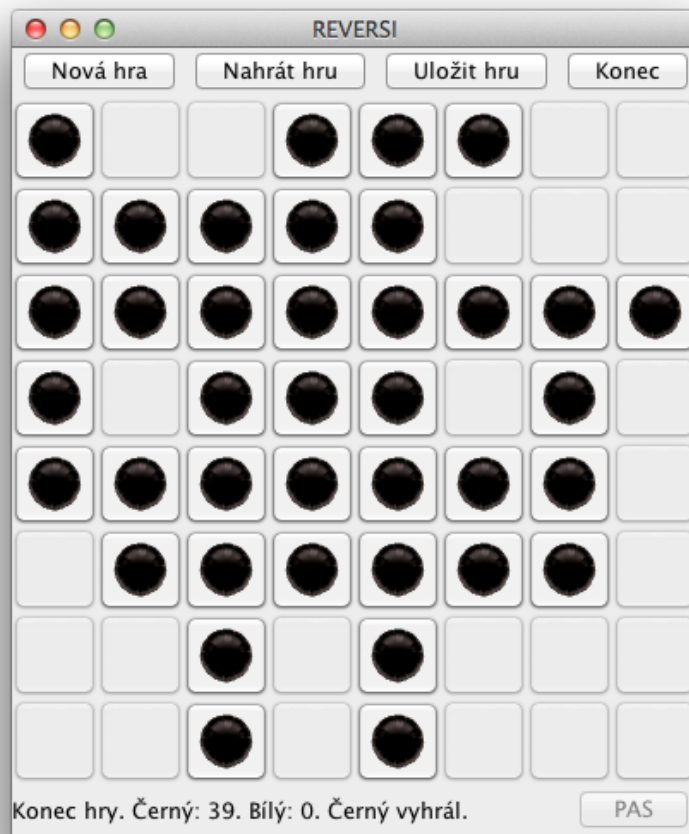
Na konci hry se dole vlevo zobrazí text se skórem (viz screenshot 5).



Screenshot 3: Rozehraná hra



Screenshot 4: Bílý musí dát pas



Screenshot 5: Drtivé vítězství černého