Vývojová dokumentace programu Damebot

Václav Stupka

28. května 2025

Obsah

1 Úvod

Damebot je program, který umožňuje hrát dámu proti počítači. Tento projekt vznikl jako zápočtový program pro předměty Programování v jazyce C# a Pokročilé programování v jazyce C# na Matematicko-fyzikální fakultě Univerzity Karlovy (obor Informatika).

Cílem projektu bylo vytvořit program, který si bude moci uživatel jednoduše stáhnout a následně ho používat bez nutnosti složité a dlouho trvající instalace. Program by měl být také jednoduchý na správu a otevřený budoucímu rozšiřování. Samozřejmostí je také jednoduché ovládání a (přiměřeně) pěkné uživatelské rozhraní.

Tento dokument představuje vývojovou dokumentaci celého projektu. Dočtete se zde, které technologie jsem použil pro správu projektu, jak probíhá sestavování programu a také popis nejdůležitějších částí zdrojového kódu.

2 Struktura řešení

Řešení je rozděleno na dva samostatné projekty. Projekt *damebot_engine* definuje obecné rozhraní pro psaní herního enginu a také výchozí implementaci potřebných tříd pro hraní dámy.

Projekt *damebot* implementuje grafické uživatelské rozhraní pro hraní dámy. Používá k tomu výchozí implementaci z projektu *damebot_engine*.

Pro napsání programu jsem použil programovací jazyk C# a grafický framework *Windows Forms*. Jako verzovací systém jsem použil *Git* spolu se vzdáleným repozitářem *GitHubem*. Jako vývojové prostředí pro psaní kódu jsem použil *Visual Studio*, grafiku (obrázky hracích kamenů) jsem vytvořil pomocí programu *Inkscape* a tuto dokumentaci píšu v *TexStudiu*.

3 Získání zdrojového kódu a sestavení projektu

Zdrojový kód lze najít na adrese `https://github.com/basvas-jkj/dame_bot`. Vytvořit lokální repozitář s kopií zdrojového kódu lze pomocí Gitu příkazem `$git clone https://github.com/basvas-jkj/dame_bot.git`.

Projekt je nejsnazší sestavit ve Visual Studiu, je možné také použít příkaz `dotnet.exe build damebot/damebot.csproj` z kořenové složky projektu. Díky závislosti *damebot* na *damebot_engine* tento příkaz sestaví oba projekty.

4 Stručný přehled souborů

damebot/DamebotGui.cs Vykreslení uživatelského rozhraní a ovládání aplikace.

damebor_engine/img/ Složka obsahující obrázky hracích kamenů ve formátu PNG.

damebor_engine/SQUARE.cs Pomocná struktura `SQUARE`, která ukládá souřadnice jednoho hracího pole.

damebor_engine/MOVE.cs Pomocná struktura `MOVE`, která ukládá jeden tah a kameny vyhozené v daném tahu.

damebor_engine/Piece.cs Třídy reprezentující kameny na hracím plánu.

damebor_engine/DamePlayer.cs Třída `DamePlayer` reprezentující hráče.

damebor_engine/Board.cs Třída `DameBoard` reprezentující hrací plochu.

damebor_engine/DameEngine.cs Třída `DameEngine` reprezentující herní engine.

5 Popis vybraných částí zdrojového kódu

V této části se pokusím podrobně popsat vybrané části zdrojového kódu. Omezuji se převážně na ty části, které jsou významné pro budoucí rozšiřování.

5.1 Reprezentace herních kamenů

Základním prvkem pro implementaci je abstraktní třída `Piece`. Jejím oddděním a implementací abstraktních metod lze reprezentovat libovolný hrací kámen. Třída je zamýšlena jako immutable, proto přesun kamene (metoda `Move()`) a případná proměna v dámu (`Promote()`) nemění stav kamene, ale vrací novou instanci.

Kromě přesunutí a proměny třída zajišťuje také enumerování všech tahů (`EnumerateMoves()` a `EnumerateJumps()`), které daný kámen může vykonat, validace zvoleného tahu (`GetMoveInfo()`), testování, zda daný kámen může skákat `CanCapture()`, a porovnání barvy s jiným kamenem (`HasDifferentColour()`).

Projekt dále obsahuje čtyři třídy reprezentující hrací kameny české dámy: `WhiteMan`, `BlackMan`, `WhiteKing` a `BlackKing`. Implementace většiny výše zmíněných metod je společná nebo velmi podobná pro kameny stejného typu (běžný kámen × dáma) bez ohledu na barvu, proto v typové hierarchii mezi zmíněnými třídami a třídou `Piece` stojí třídy `ManBase` a `KingBase`.

Struktura `MOVE_INFO` ukládá typ jednokrokového tahu (prostý přesun nebo jedna část vícenásobného skoku), souřadnici cílového pole a případný vyhozený kámen. Používá se pro uložení výsledku testování platnosti tahu a pomáhá také při enumerování možných tahů.

5.2 Reprezentace hrací plochy

Hrací plocha je zodpovědná za ukládání hracích kamenů na příslušných pozicích, vykonání tahu včetně případné proměny, statické vyhodnocení pozice a také vygenerování výchozí pozice. Tomu odpovídají po řadě readonly

indexer a metody `PerformMove()`, `EvaluatePosition()`, `GenerateInitialPieces()` deklarované v interfacu `IBoard`. Třídy implementující toto rozhraní nemusí být immutable, pro potřeby hledání dalších tahů je rozhraní doplněno metodou `SimulateMove()`, která vykoná tah na kopii hrací plochy.

Toto rozhraní je dostačující pro implementaci libovolné čtvercové hrací desky, na které se hraje dáma nebo podobné hry. Třída `DameBoard` představuje výchozí implementaci pro hrací desku české dámy.

5.3 Reprezentace hráče

Instance typu hráč (interface `IPlayer`) ukládá seznam kamenů, které má hráč na hrací desce. Reprezetuje také algoritmus pro automatické hledání příštího tahu, tomu odpovídá asynchronní metoda `FindNextMove()`.

Výchozí implementace hráče, třída `DamePlayer`, používá k hledání dalšího tahu minimaxový algoritmus. Konkrétně vyenumeruje možné tahy pro všechny svoje figurky, a pro každý hledá rekurzivně odpovědi pro svého protivníka. Díky paralelizaci je možné tento postup opakovat až do hloubky 6 (tedy tři tahy hráče a tři reakce protivníka). Vyšší hloubka by vyžadovala další optimalizace, např. cachování a prořezávání. Pozice po těchto šesti přesunech je ohodnocena na základě počtu a typu kamenů, které oba hráči drží. Podle výsledku ohodnocení je poté na každé úrovni vybrán nejlepší tah pro daného hráče a je předán do nižší hloubky.

Ačkoli se výchozí implementace jmenuje `DamePlayer`, ve skutečnosti nezávisí na implementačních detailech dámy: generování tahů zajišťuje třída `Piece` a ohodnocení pozice interface `IBoard`. `DamePlayer` je díky tomu fakticky pouze implementací minimaxového algoritmu pro libovolnou hru, nikoli pouze pro (českou) dámu.

Největším omezením rozhraní `IPlayer` je požadavek na hru dvou hráčů a to i v situaci, že používáme jiný algoritmus než minimax. Ačkoli existují modifikace dámy, které lze hrát ve více než dvou hráčích, jedná se o velmi neobvyklé hry, které navíc často vyžadují nečtvercovou hrací plochu. Tudíž toto omezení nepovažuji za příliš zásadní.

5.4 Herní engine

Herní engine reprezentuje interface `IEngine` a jeho výchozí implementace `DameEngine`. Zprostředkovává komunikaci s uživatelským rozhraním, kontroluje zvolený tah fyzického hráče a iniciuje hledání a vykonávání tahu automatického hráče. Na rozdíl od IPlayer rozhraní `IEngine` nevyžaduje existenci dvou hráčů.

Validaci tahů zajišťují metody `IsOnMovePiece()` (hráč označil svůj vlastní kámen) a `ValidateMove()` (zvolený tah je v souladu s pravidly). Vlastní vykonání tahu zajišťují metody `PerformMove()` a `PerformAutomaticMove()` – druhou zmíněnou bylo nutné přidat kvůli možnosti, že zahajující hráč je reprezentovaný počítačem.

Komunikace s uživatelským rozhraním je zajištěna pomocí událostí. Konkrétně událost `OnMove` oznamuje vykonání tahu a dále předává informaci, který další hráč je právě na tahu. Událost `OnMark` předává tah, který chce vykonat automatický hráč a tím umožňuje uživatelskému rozhraní tento tah oznámit fyzickému hráči. Poslední událost `OnGameOver` oznamuje konec hry a současně předává hráče, který zvítězil.

6 Možná budoucí rozšíření

Damebot je otevřený budoucímu rozšiřování. Snažil jsem se ho navrhnout tak, aby případné změny a vylepšování nevyžadovaly začít programovat úplně od začátku. Ačkoli damebot implementuje pouze českou dámu, rozhraní hracího enginu je otevřené i jiné modifikace dámy. Ačkoli k tomu nebylo rozhraní navrženo, nemělo by být problém vytvořit ani engine pro hry výrazně odlišné od dámy, např. pro šachy.

Pokud bychom chtěli přidat nový kámen, stačí vytvořit nového potomka třídy `Piece`. Jiné velikosti hrací plochy nebo jiné výchozí pozice lze docílit vlastní implementací interfacu `IBoard`. Pokud bychom chtěli implementovat nějaký jiný algoritmus pro hledání dalšího tahu (např. efektivnější minimax nebo naopak hloupější algoritmus vhodný pro slabší hráče), stačí implementovat třídu pro rozhraní `IPlayer`. Pokud bychom potřebovali změnit např. pravidla ohledně výhry jednotlivých hráčů nebo pořadí, v jakém se hráči střídají, stačí napsat kód třídy implementující `IEngine`.

Tyto změny lze provádět nezávisle na sobě, tudíž je možné např. použít výchozí minimaxový algoritmus a výchozí hrací plochu spolu s vlastním typem hracích kamenů nebo naopak použít výchozí kameny i hrací plochu a vlastní engine, který např. umožňuje zahrát mimořádný tah při proměně běžného kamene v dámu.

Pokud jde o uživatelské rozhraní, jediná změna, kterou by bylo nutné provést, je inicializace příslušného enginu. Přirozeným rozšířením by tedy byla možnost volby mezi různými enginy, případně možnost nahrát engine jako plugin.