

Semestrální úloha X36TJC — Hra šachy

Pavel Tvrđík <tvrdipa1@fel.cvut.cz>, ZS 2010-2011

1 Úvod

Navázal jsem na mojí semestrální práci do předcházejícího předmětu o jazyku C/C++ (X36PJC). Původní práce je přiložená v souboru tvrdipa1_x36pjp.zip. Implementoval jsem nové techniky, které jsem se naučil během semestru. Opravil jsem nedostatky, které vyplynuly z hodnocení původní semestrální práce, které jsem uložil do souboru hodnoceni_x36pjp.txt. Vylepšil jsem pak ještě pár drobností (oznamování šachu, správný přechod na další hru a ohodnocovací funkci).

2 Zadání

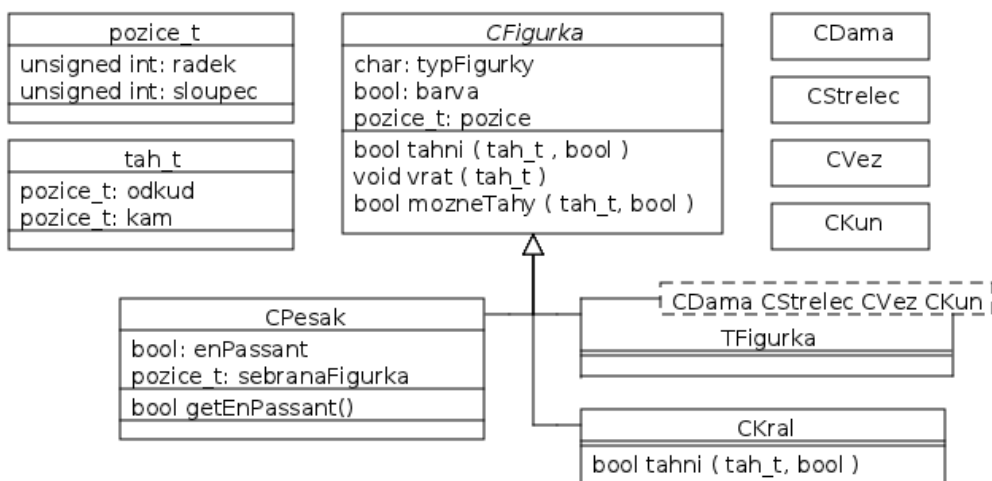
2.1 X36PJC

Realizujte hru „šachy“ podle obvyklých pravidel. Program bude umět zobrazovat hrací plochu s figurkami, bude reagovat na požadavky o tahy (kontroluje správnost tahu) a bude obsahovat umělou inteligenci, která umožní hru člověka proti počítači. Pro zobrazení stačí textový výstup, zájemci mohou použít ncurses, rozhraní Windows konzole nebo grafický výstup.

2.2 X36TJC

Při implementaci použijte STL, šablony a polymorfismus.

3 Nástin zvoleného řešení



Zjednodušený diagram tříd

Šachovnice je reprezentována dvourozměrným polem ukazatelů na jednotlivé figurky (objekty). Každé šachové políčko nese informaci, která figurka na něm stojí (jinak ukazuje na NULL) a má místní seznam dosahů resp. krytí všech ostatních figurek na toto políčko. Třídy jednotlivých typů figurek jsou vytvořené děděním z třídy **CFigurka**. Figurky obsahují generátor tahů, možnost táhnout figurkou a táhnout zpět.

Umělá 'intelligence' funguje na zjednodušeném principu minimaxu [2]. Program vyzkouší všechny tahy figurkami dané barvy a každý tah (pozici) nějak ohodnotí. Pracuje se vzájemným dosahem, krytím figurek s jejich váhou, kdo pak bude na tahu atp. Čím nižší zápornější skóre, tím lépe pro černého a čím vyšší kladné, tím lépe pro bílého hráče.

4 Poznámky o implementaci

Celý program jsem napsal v editoru (g)vim [3] a pomocí programu valgrind [4]. K zobrazení výstupu jsem se rozhodl pro ncurses [5].

4.1 Polymorfismus

Je použit při manipulaci s figurkami, které byly vytvořeny zděděním z třídy CFigurka.

4.2 Šablony

Nenašel jsem bohužel v programu místo, kde by se šablony přirozeně hodily použít. Implementoval jsem šablonu na vytvoření figurky dáma, stálec, věž nebo kůň podle dané třídy. Pěšák a král mají svá specifika, ty jsem nechal zvlášť.

4.3 STL

Využil jsem kontejner vector na seznam krytí a dosahů figur po šachovnici.

4.4 Soubory

- dat_struct.h — datové struktury pozice_t, tah_t
- figurky.h — třídy všech figurek
- sach_global.h — všechny globální proměnné, naštěstí jich ale není moc
- chess.h — vstup / výstup, umělá intelligence a vše, co se jinam nevešlo

4.5 Kompilace

Semestrálku jsem přepsal pod Archlinuxem. Verze g++ 4.5.2. Kompilace a spuštění doufám proběhne v pořádku pouze spuštěním příkazu.

```
make&&./chess.out
```

Pokud se program spustí s parametrem -v nebo --verbose, zobrazí se vám po straně šachovnice všechny přípustné tahy a jejich ohodnocení a dole vyznačení nejlepšího tahu dle programu.

```
./chess.out -v
```

4.6 Ovládání

Tahy se zadávají jako políčko, odkud se má táhnout a políčko, kam se má táhnout bez mezery např. e2e4, b1c3,.. Rošáda je tah krále na pozici věže, se kterou se má vyměnit. Při brání mimochodem je cílové políčko to, kde pěšák po tahu zůstane.

5 Závěr

Tahy s figurkami jsou snad zcela podle pravidel včetně rošád (před tím se nesmí táhnout ani králem ani věží), proměny pěšáka na konci šachovnice v dámu (ale není možnost si vybrat jinou figurku) a brání mimochodem u pěšáků. Program nedovolí táhnout figurkou

tak, aby hráč třeba nechal svého krále v šachu nebo dokonce ho dostal do šachu. Program rozpoznává šach mat, šach pat a tzv. mrtvou pozici, kdy už žádná posloupnost tahů nevede k šach matu.

Při implementaci jsem využil STL, šablony a polymorfismus.

6 Zdroje

- (1) <http://www.chess.cz/www/assets/files/informace/legislativa/PravidlaSachuFIDE2009.pdf>
- (2) http://www.linuxsoft.cz/article.php?id_article=1109
- (3) <http://www.vim.org/>
- (4) <http://valgrind.org/>
- (5) <http://www.gnu.org/software/ncurses/>
- (6) <http://www.usenet.com/newsgroups/rec.arts.ascii/msg00005.html>