Tomáš Čelko

**Zápočtový program: THE CHESS**

**Anotácia:**

Šach je strategická hra s úplnou informáciou pre dvoch hráčov, v ktorej o výsledku rozhodujú taktické a strategické schopnosti hráčov a ich pozornosť a sústredenie. Cieľom hry je dať súperovi mat. Má v sebe prvky umenia, vedy a športu. Šach rozvíja logické myslenie a cvičí pamäť. V dnešnej verzii šachových kameňov možno badať alegóriu stredovekej spoločnosti.

**Figúrky:**

Šach sa hrá na doske zvanej šachovnica, rozdelenej na 8×8 políčok. Každý hráč má na začiatku partie k dispozícii nasledujúce figúrky:

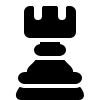
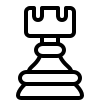
* 1 kráľa



* 1 dámu (nesprávne kráľovná)



* 2 veže



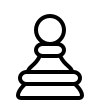
* 2 strelcov



* 2 jazdcov (nesprávne kôň)



* 8 pešiakov (nesprávne sedliak



**Výhra**

* vyhráva hráč, ktorý dá súperovi mat. Mat je situácia, keď je možnosť brať kráľa a súper nemá možnosť tomu zabrániť. Pri mate sa iba výnimočne berie kráľ, k výhre stačí mat zahlásiť.
* častejšie sa končí partia tým, že jeden súper uzná prevahu súpera, vzdá beznádejnú pozíciu.

**Remíza**

* po trojnásobnom dosiahnutí rovnakej pozície - hráč na ťahu musí reklamovať

(implementované vo verzii dosiahnutie 3x rovnakej pozície v posledných 6 ťahoch)

* ani jeden hráč nemá materiál na dosiahnutie matovej situácie
* je pat, teda situácia, keď strana, ktorá je na ťahu, nemôže spraviť žiaden legálny ťah podľa pravidiel, pričom nie je mat.

**Program: TheChess**

**Triedy:**

* **Move**

Uchováva dáta o ťahu – teda je to usporiadaná štvorica súradníc.

* **MyPictureBox**

Upravená verzia pictureboxu, ktorá má naviac atribút dvojicu súradníc, ktoré označujú, kde sa picturebox nachádza.

* **Coordinates**

Usporiadaná dvojica súradníc, veľakrát používaná na vyjadrenie polohy objektov na šachovnici.

* **ChessPiece**

Figúrka, ktorá ma atribúty ako napríklad Coordinates, vie o zozname jej možných ťahov, uložených v .possible\_moves. A tiež má ID – pre čierne figúrky záporné, pre biele kladné, čo je veľmi dobre využiteľné v situácii, keď chceme napríklad zistiť, či je možné figúrkou ťahať (využijeme súčin premennej turn (hodnoty 1 a -1) a ID (hodnoty +- 1 až +-6)

* **Castling**

Opisuje stav veží a kráľov a obsahuje potrebné informácie, ktoré vyjadrujú, či a kde je možné previesť rošádu.

* **AI**

Obsahuje atribúty a metódy potrebné pre oponenta vo verzii singleplayer (jeden hráč) alebo AI vs. AI (počítač proti počítaču).

* **EvalData**

Spracováva data z textových súborov pre pozičné ohodnotenie jednotlivých figúrok vo verzii singleplayer (jeden hráč) alebo AI vs. AI (počítač proti počítaču).

* **Judge**

Má na starosti počítanie ohodnocovacej funkcie založenej na materiály a pozícii figúrok.

* **Intro**

Úlohou tejto triedy je zabezpečiť dialóg s hráčom o tom, aký typ hry by chcel hrať a nastaviť potrebné atribúty podľa toho.

* **TheChess**

Hlavná trieda, ktorá spracováva používateľské rozhranie, striedanie ťahov ako aj počítanie možných ťahov pre figúrky.

**AlgoritmusAI:**

Algoritmus, ktorý umožňuje počítať ťahy pre AI, som zvolil minimax s alfa-beta orezávaním, ktoré má obmedzenú hĺbku, nastaviteľnú v úvode hry. V prípade väčšej hĺbky algoritmus najprv spočíta čo možno najlepší ťah v o jedna menšej hĺbke a potom výsledok tohto výpočtu použije ako prvú pozíciu v minimaxe s danou hĺbkou, kde je predpoklad, že sa hodnoty alfa a beta vhodne nastavia a väčšia časť výpočtu sa oreže a program tak zrýchli.

**Módy hry:**

* 1. **Singleplayer** = hra jedného hráča proti počítaču
  2. **Multiplayer** = hra dvoch hráčov, ktorí sa striedajú v ťahoch
  3. **AI vs AI** = hra ‘‘dvoch počítačov‘‘ proti sebe

**Dôležité Metódy:**

* **InitStart** – Inicializuje premenné a volá funkcie, ktoré zabezpečia štart hry
* **FillArray** – Rozostaví figúrky do poľa 8x8 podľa pôvodného rozostavenia šachovnice
* **Redraw** – Podľa stavu poľa šachovnice vykreslí na obrazovku zodpovedajúci stav pictureboxov
* **AddPossibleMoves** – Pre figúrku, na ktorú užívateľ klikol vypočíta možné ťahy
* **DeleteCheckMoves** – Odstráni ťahy, ktoré by priviedli tvojho kráľa do šachu (zakázané ťahy)
* **FindCheck** – Zistí, či je daný kráľ v šachu
* **ISDraw** – Metóda, ktorá vyhodnotí stav bufferu a množstvo materiálu na ploche a rozhodne, či nastala remíza
* **CountMoves** – Spočíta počet možných ťahov pre hráča na ťahu
* **PicClicked** – Handler pre event kliknutia na políčko (picturebox)
* **MakeMove** – Vykoná ťah figúrky na základe zadaných parametrov
* **FindBestMove** – Pomocou Minimaxu nájde čo najlepší ťah pre AI oponenta
* **Evaluate** – Ohodnotí situáciu na šachovnici na základe pozície a materiál

Hodnoty: pešiak = 100, jazdec = 320, strelec = 330, veža = 500, dáma = 900

**Notácia:**

V programe som sa snažil rozlíšiť Metódy ktoré sa v prípade viacslovného názvu písali spolu s veľkými písmenami na začiatku slov (NejakáMetóda) a atribúty s malými písmenami a podtržníkom medzi slovami (nejaký\_atribút).

**Ovládanie:**

Po úspešnom zvolení módu hry a výbere ďalších dát (farba vašich figúrok, prípadne obtiažnosť AI) sa šach ovláda jednoduchým klikom na vašu figúrku počas vášho ťahu a následným kliknutím na jedno z modrých políčok (na ktoré figúrka môže ťahať). Po presune pešiaka na druhú stranu šachovnice sa objaví možnosť vybrať figúrku, na ktorú chcete pešiaka transformovať, kde stačí opäť kliknúť na jednu z možností. Hra automaticky ohlasuje stav šachu, prípadne konca hry, tam stačí kliknúť na tlačidlo „OK“.

**Vstupné dáta:**

Program využíva ako vstupné dáta obrázok, požitý ako pozadie úvodnej plochy, rovnako tak využíva obrázky znázorňujúce jednotlivé figúrky. Program využíva aj textové súbory, v ktorých sú uložené dáta o hodnotení pozícii figúrok pri Minimaxe v špecifickom formáte a to: 8 riadkov po 8 celých (4B) čísel, oddelených bielou medzerou a na konci každého riadku je za posledným číslom čiarka.

**Práca na projekte:**

Spočiatku išla práca na projekte veľmi pomaly, lebo som presne nevedel, ako chcem aby boli jednotlivé dáta reprezentované. Viac návrhov som zahodil a začal odznovu. Postupne som začal implementovať grafické rozhranie a ťahy rôznych figúrok. Táto časť tvorí pomerne veľkú časť programu, ale je nutno si priznať, že figúrky v šachu sa vo všeobecnosti hýbu pomerne netriviálne. Keď už toto fungovalo, dokončil som mód multiplayer a začal pracovať na móde singleplayer. Implementácia minimaxu prebehla podľa plánu a alfa-beta orezávanie výpočet výrazne zrýchlilo. Nakoniec zostávalo už len vytvoriť úvod a dialóg s užívateľom o výbere typu hry a podobne. A samozrejme ešte napísať dokumentáciu, aby sa v mojom programe ešte niekto niekedy vedel orientovať (o čo sa momentálne snažím ;) ). Nakoniec by som sa chcel ešte poďakovať Ivane Piačkovej za testovanie hry a spätnú väzbu, vďaka ktorej sa mi podarilo dorobiť tento program do (dúfam) hrateľnej verzie.

**Záverečný povzdych**

Šach bola moja prvá skúsenosť s projektom takéhoto rozmeru. Práca na ňom ma veľmi bavila, keďže som fanúšikom tejto hry. Fakt, že si môžem zahrať šach proti počítaču, ktorý som sám naprogramoval, ma naozaj teší. Čo sa týka obtiažnosti AI, nepodarilo sa mi dosiahnuť takú výpočetnú hĺbku, aby moje AI mohlo súťažiť so svetovou konkurenciou, za to je ale dôstojným súperom pre amatérskych hráčov šachu, akým som aj ja. Verím teda, že sa vám bude hra páčiť a užijete si pri nej veľa zábavy. ☺