Eulerov kôň

Zadanie 2 pre predmet Umelá Inteligencia

Zadanie

Úlohou je prejsť šachovnicu legálnymi ťahmi šachového koňa tak, aby každé políčko šachovnice bolo prejdené (navštívené) práve raz. Riešenie treba navrhnúť tak, aby bolo možné problém riešiť pre štvorcové šachovnice rôznych veľkostí (minimálne od veľkosti 5 x 5 do 20 x 20) a aby cestu po šachovnici bolo možné začať na ľubovoľnom východziom políčku.

Úloha (g)

... Implementujte túto heuristiku do algoritmu prehľadávania stromu do hĺbky a pre šachovnicu 8x8 nájdite pre 10 rôznych východzích bodov jedno (prvé) správne riešenie (pre každý východzí bod). Algoritmus s heuristikou treba navrhnúť a implementovať tak, aby bol spustiteľný aj pre šachovnice iných rozmerov než 8x8. ...

Riešenie

Stav

Stav reprezentuje šachovnicu, kde je určené na každom políčku či už bolo navštívené alebo nie (2D bool pole). Zároveň si pamätá veľkosť šachovnice (2xInteger) a aktuálnu pozíciu koňa (2xInteger).

Operátory

Každý typ pohybu, ktorý môže kôň spraviť na šachovnici je reprezentovaný operátorom, čiže dostupný počet operátorov je 8. To či sa dá operátor použiť vieme zistiť na základe aktuálnej pozície koňa, veľkosti šachovnice a už použitých políčok. Cena všetkých operátorov je 1.

Heuristická funkcia

Na vyhľadávanie cesty som použil Warnsdorfové pravidlo (H. C. von Warnsdorf, 1823). Princíp tejto heuristiky spočíva v tom, že cena stavu je rovná počtu dostupných operátorov (obr.č.1).

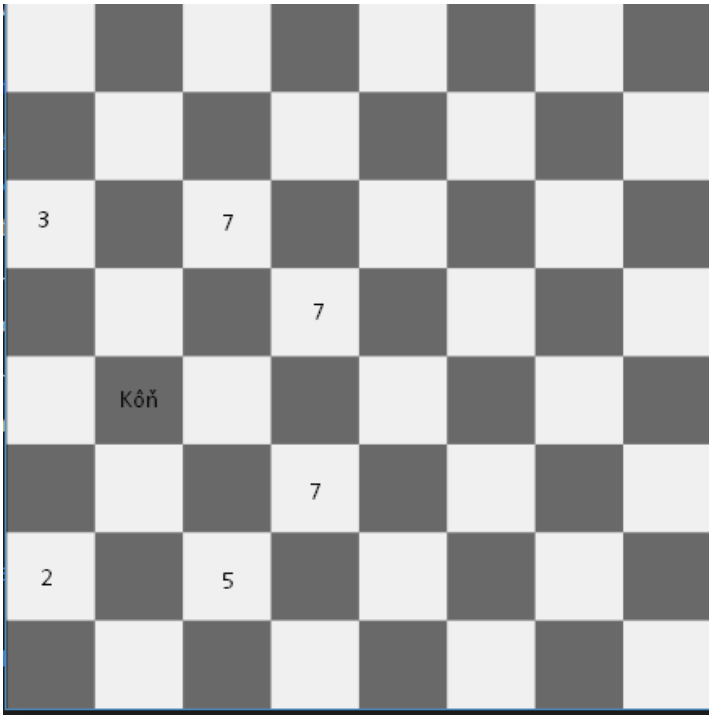
Uzol

Uzol si okrem aktuálne stavu pamätá ešte zoznam políčok (List<Tuple<int,int>>) z ktorých sa dostal do aktuálneho stavu (použité operátory) , počet políčok ktoré ešte treba navštíviť (Integer) a použiteľné operátory (List<Tuple<int,int>>), čo zároveň slúži ako cena.

Algoritmus

1. Vytvor počiatočný uzol a vložím ho do zásobníka
2. Ak je nepárna veľkosť šachovnice a začína sa na párnom políčku, skonči – riešenie neexistuje
3. Ak je zásobník prázdny, skonči – riešenie neexistuje
4. Vyber uzol zo zásobníka
5. Ak sa jedná o konečný uzol skonči a vráť cestu
6. Ak nemôžeš použiť žiaden operátor zahoď uzol – slepá ulička
7. Vygeneruj uzly(listy) pre každý dostupný operátor
8. Vlož vygenerované uzly od najdrahšieho po najlacnejší do zásobníka
9. Choď na krok 3

Matúš Tundér (74288)
17. 3. 2017



obr.č.1 – Cena nasledujúcich uzlov

Implementácia

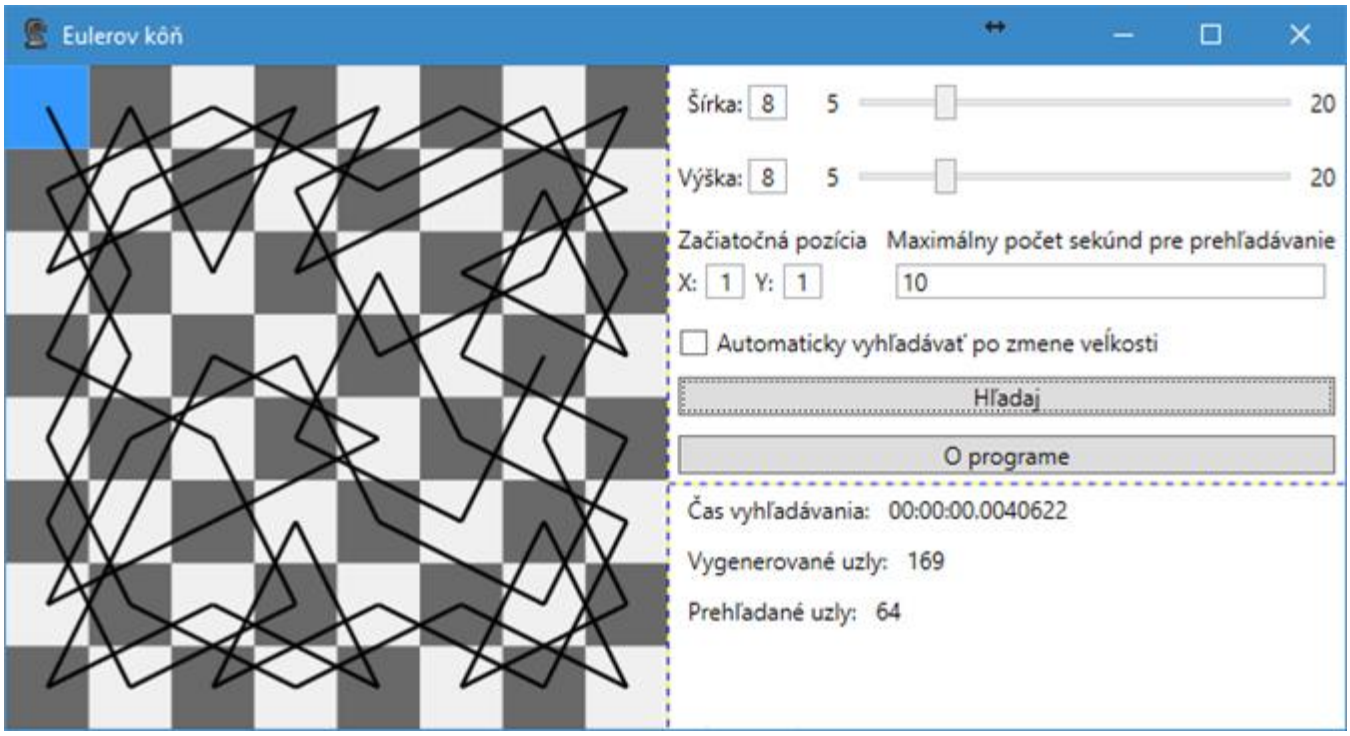
Pre vyriešenie tejto úlohy som si zvolil jazyk C# 6 na .Net Framework-u 4.5.2. Príklad aplikácie je možné vidieť na obrázku č.2. Hlavná logika je osamostatnená v Dll, v ktorej sa nachádza upravený algoritmus spomenutý vyššie. Hlavnou zmenou je pridanie časovača, ktorý má na starosti aby sa neprehľadávalo dlhšie než povolený čas. Taktiež si pamätá počet vygenerovaných/navštívených stavov a celkovú dĺžku vyhľadávania. Knižnica obsahuje statickú triedu Operatory, v ktorej sa nachádzajú metódy, ktoré vracajú dostupné pozície pre aktuálny stav. Internal triedu Uzol kde sa nachádzajú všetky údaje spomenuté v kapitole Uzol obohatený o metódy, ktoré dokážu vytvoriť nový uzol a k nemu priradiť dostupné operátory. Nakoniec je tam verejná trieda Search, ktorá má v sebe vyhľadávací algoritmus a atribúty určené pre štatistiku. Podobnejší popis je možné si pozrieť v Summary a komentároch zdrojového kódu.

Zhodnotenie

Warnsdorfové pravidlo dokáže výrazne zlepšiť vyhľadávanie. Pre veľkosť šachovnice 20x20 vyhľadávanie trvá v priemere menej než 0.03 sekundy. Po testovaní na rôznych vstupoch sa mi podarilo zistiť, že stále existujú prípady kedy aj za pomoci použitia tejto heuristiky sa dĺžka prehľadávania predĺži aj 100-násobne. Pri úzkych a vysokých rozmeroch šachovnice alebo v niektorých prípadoch kedy je začiatok blízko kraja môžeme pozorovať spomalenie, keďže vyhľadávanie trvá dlhšie a môže aj presiahnuť časový limit. Jeden s pomalých vstupov je začiatok 5x5 pre rozmery 5x8 alebo začiatok 5x2 pre 8x8.

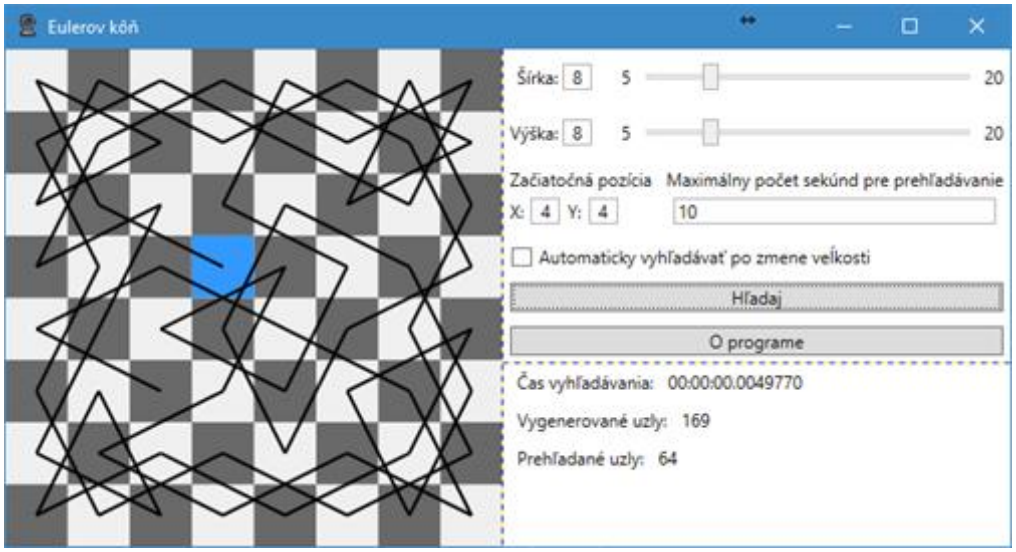
Program využíva približne 50 MB pamäte (GUI). Algoritmus si v pamäti uchováva iba nespracované uzly. Jeden uzol má bez zapamätaných použitých operátorov priemerne 3 KB. Keďže sa prehľadáva do hĺbky je lepšie pamätať si zoznam použitých operátorov pre daný uzol namiesto všetkých uzlov pre určenie cesty. Takže každý uzol má ešte o cca 20 bajtov za úroveň (hĺbku v strome) navyše. Pri veľkosti 20x20 v hĺbke 400 je zapamätaných 969 uzlov (čo je zároveň maximum) čo je menej ako 10 MB pre najväčšiu šachovnicu.

Matúš Tundér (74288)
17. 3. 2017

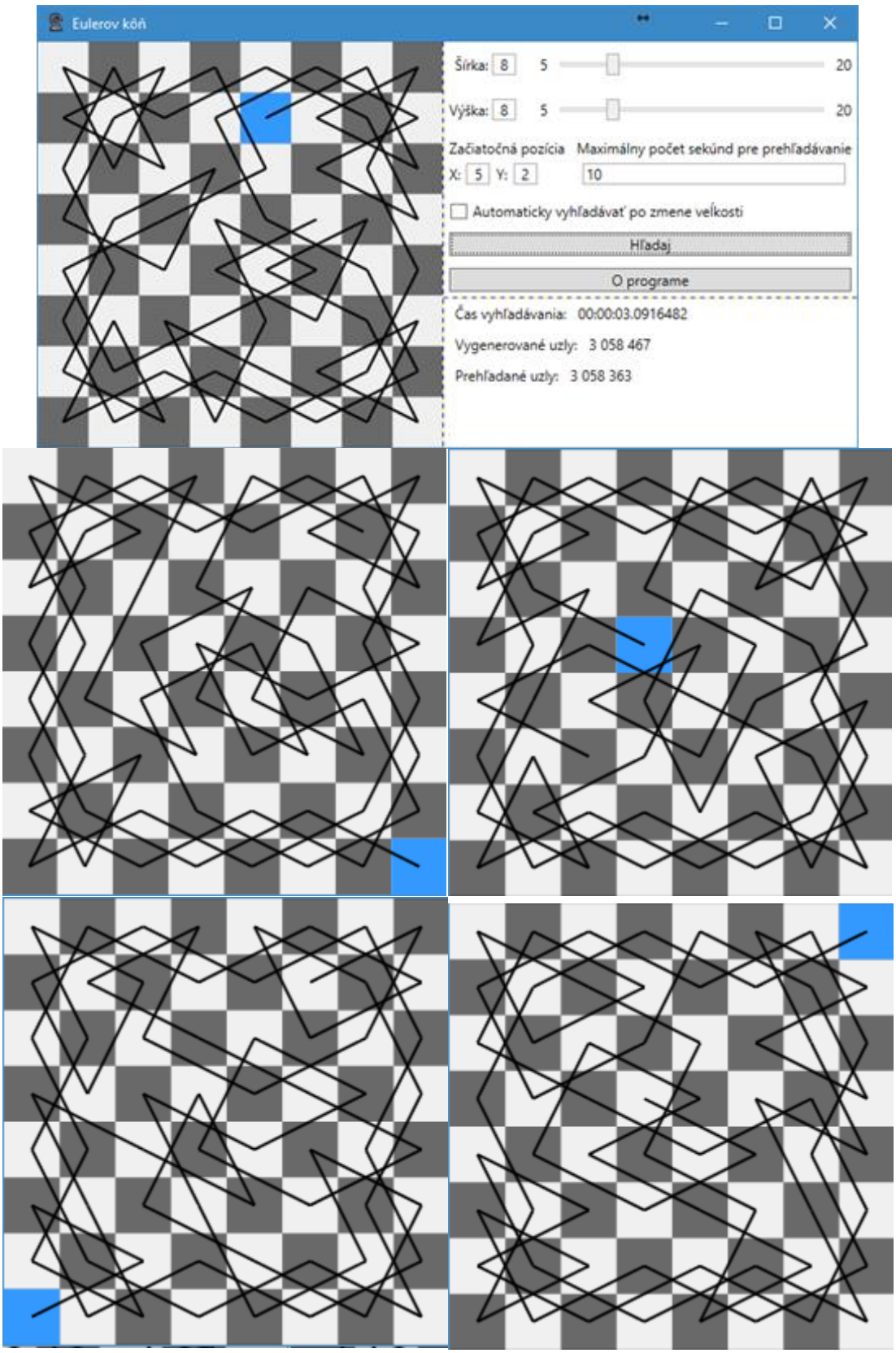


obr.č.2 – Vzhľad aplikácie

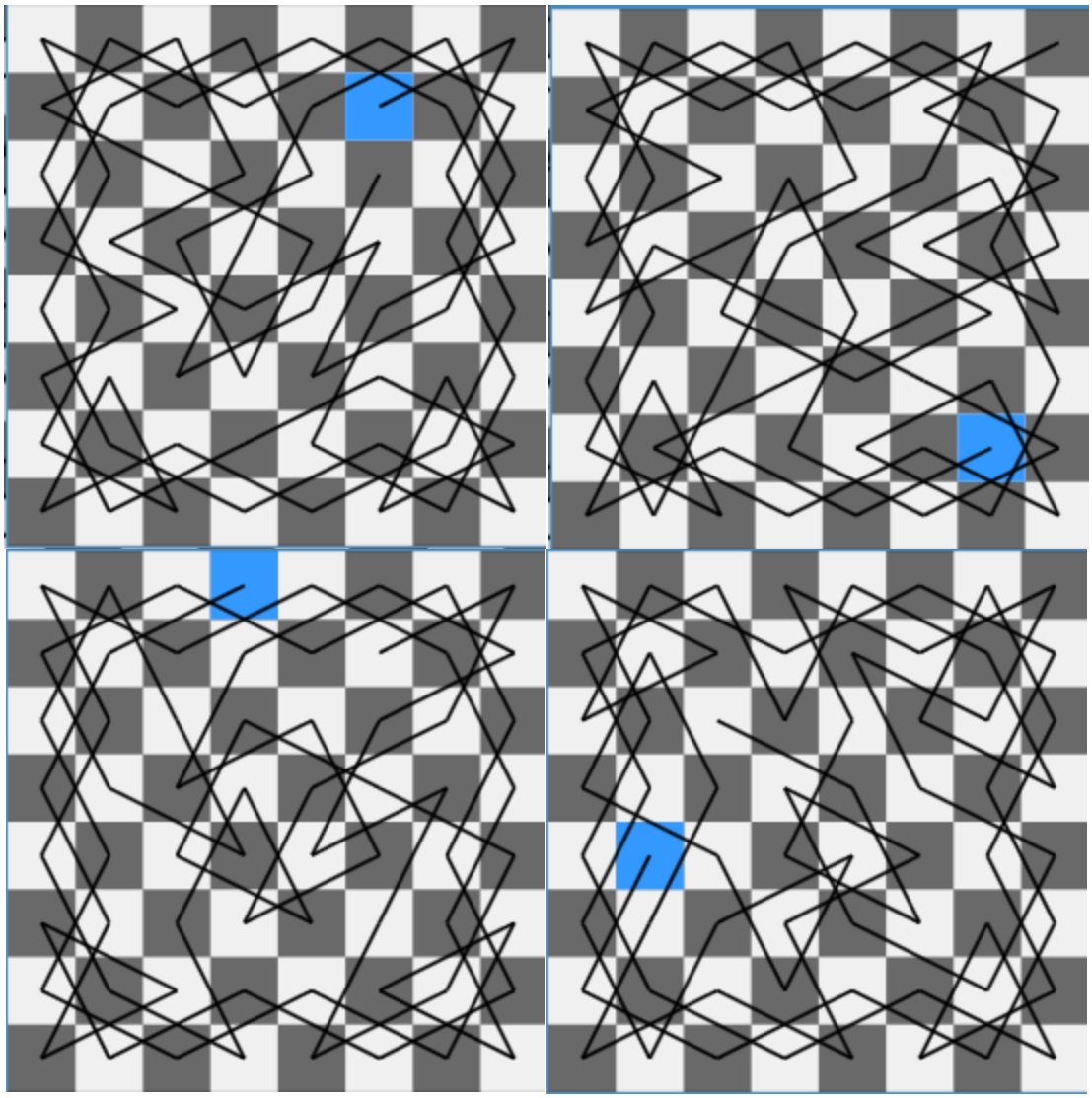
Nájdené riešenia pre šachovnicu 8x8



Matúš Tundér (74288)
17. 3. 2017



Matúš Tundér (74288)
17. 3. 2017



Matúš Tundér (74288)
17. 3. 2017