

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 východizom 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. Zároveň si pamätá veľkosť šachovnice a aktuálnu pozíciu koňa.

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 políček. Cena všetkých operátorov je 1.

Heuristická funkcia

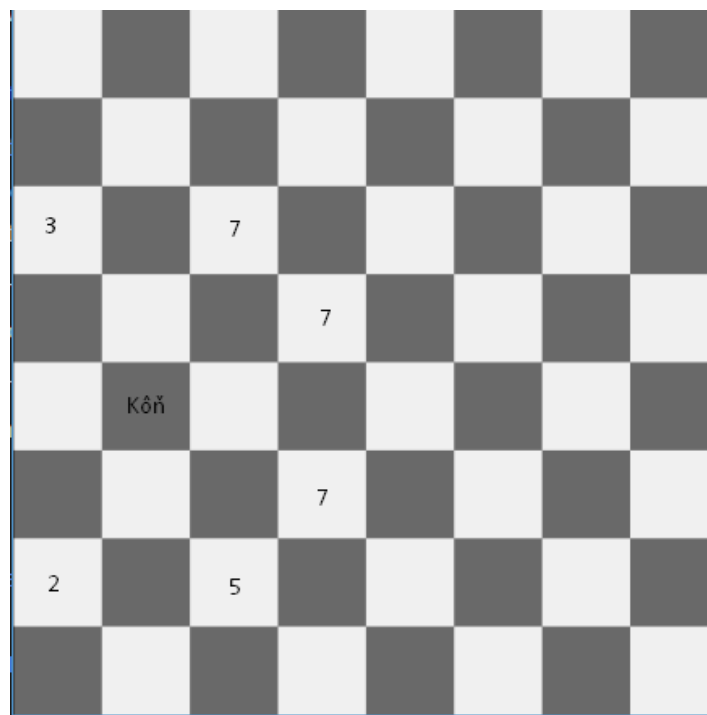
Na vyhľadávanie cesty som použil Warnsdorfové pravidlo (H. C. von Warnsdorf v roku 1823). Princíp tejto heuristiky spočíva v tom, že cena stavu je rovná počtu dostupných operátorov (obr.č.1).

Uzol

Uzol si okrem aktuálne stavu pamätá ešte zoznam políček ,z ktorých sa dostal do aktuálneho stavu (použité operátory) , počet políček ktoré ešte treba navštíviť a použiteľné operátory, čo zároveň slúži ako cena.

Algoritmus

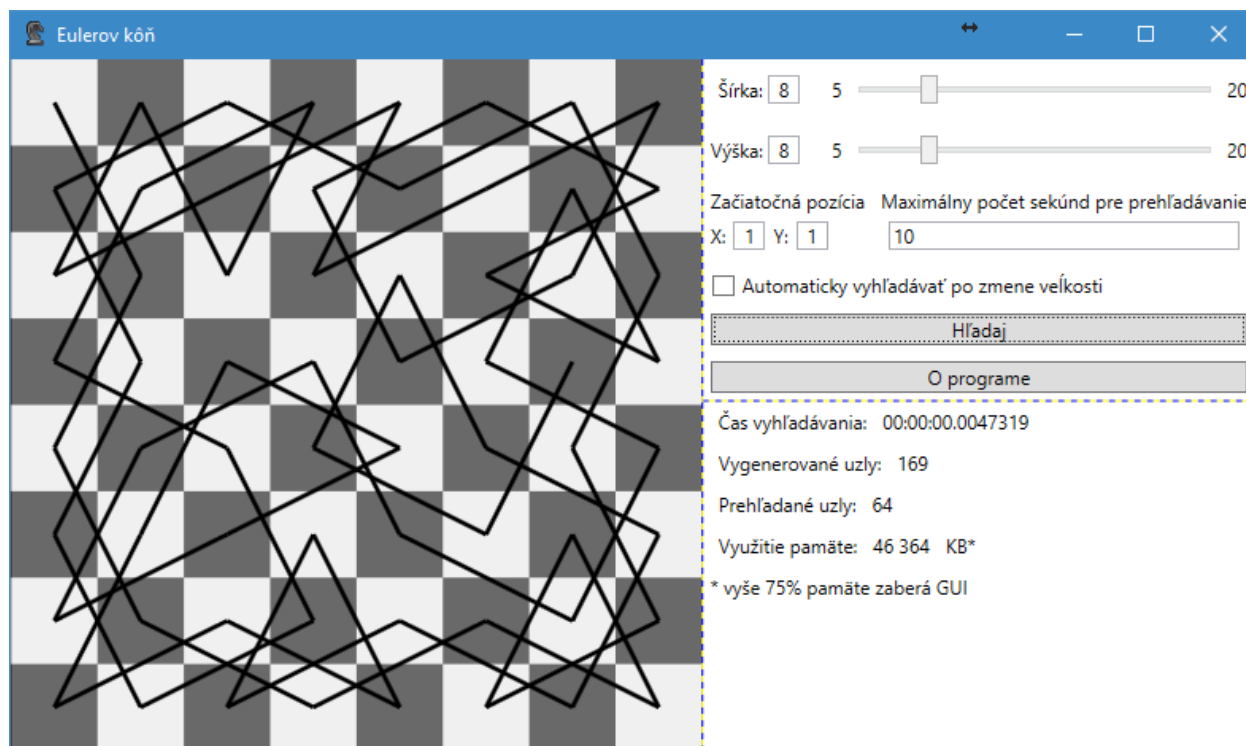
1. Vytvor počiatočný uzol a vložíme ho do zásobníka
2. Ak je zásobník prázdny, skonči – riešenie neexistuje
3. Vyber uzol zo zásobníka
4. Ak sa jedná o konečný uzol vráť použité operátory
5. Ak nemôžeš použiť žiaden operátor zahod' uzol – slepá ulička
6. Vygeneruj uzly(listy) pre každý dostupný operátor
7. Vlož vygenerované uzly od najdrahšieho po najlacnejší do zásobníka
8. Chod' na krok 2



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.



obr.č.2 – Vzhľad aplikácie

Nájdeneé riešenia pre šachovnicu 8x8

