Peter Kúdela

UI

Zadanie 2

8-hlavolam

A* algoritmus

28.10.2020

Používanie

stateGen.py

V adresári sa nachádza program stateGen.py ktorý slúži na vygenerovanie predpripravených vstupov (vstupy sú generované vo forme txt súborov v adresári kde bol program spustený).

main.py

Po spustení bude užívateľ vyzvaný vložiť názvy súborov reprezentujúcich začiatočný a konečný stav hlavolamu, nasledovne výber heuristiky. Po týchto voľbách program vypíše graficky kroky postupu, trvanie priebehu, počet spracovaných a vytvorených uzlov a postupnosť krokov riešenia.

Štruktúra programu

Node

Node je reprezentácia uzlu, ktorá si pamätá stav hlavolamu ako aj nasledovné informácie:
Grid – reprezentácia stavu v tvare 2D poľa
Parent – odkaz na rodičovský uzol
Depth – hĺbka uzla potrebná pre A* (hodnota q)
H – výsledok heuristickej funkcie
F – depth + h potrebná pre A*
lastOp – operácia ktorou sme sa na tento stav dostali (zmenšuje počet vytvorených potomkov čo znižuje čas hľadania)

A taktiež pre node špecifické funkcie:
Shifts – mená funkcií korešpondujú so smerom posunu
showGrid – vypíše 2D pole
heur – odkaz na užívateľom zvolenú heuristiku
heur1 – implementácia heuristiky 1
heur2 – implementácia heuristiky 2

Dictionaries opened a closed

Opened a closed slúžia na ukladanie vygenerovaných nespracovaných a spracovaných stavov. Tvoria ich kombinácie uzlov a ich kľúčových hodnôt (za kľúčové hodnoty som vybral výsledok heuristickej funkcie čo mi pomáha znížiť čas pri hľadaní výskytu stavov v denníkoch)

Pomocné funkcie

Okrem funkcii objektu node a samotného algoritmu sa v programe nachádzajú nasledovné pomocné funkcie:
findNum – ak sa v 2D poli hľadané číslo nachádza, vráti súradnice x a y
showPath – vykreslí na výstup cestu k danému uzlu
isIn – vráti True ak sa v denníku nachádza uzol s rovnakým stavom ktorého hodnota f je menšia

A* algoritmus

Algoritmus prebieha v cykle ktorý sa vykonáva kým sú uzli v denníku opened. Na začiatku každej iterácie vyberieme z denníku opened uzol s najmenšou hodnotou f, tento uzol nasledovne z denníku odstránime. Ďalej vytvoríme 3-och potomkov súčasného uzla (okrem spätného, môžu byť 4 ak ide o počiatočný stav), potomkov ktorým sa nepodarí vykonať shift operácia, alebo sa ich stav nachádza v denníkoch opened alebo closed s menšou hodnotou f ignorujeme. Na konci iterácie pridáme súčasný uzol do denníku closed. Kontrolu či sme našli cieľový stav vykonávame pri tvorbe potomkov, ak ho nájdeme vypíšeme cestu. Ak nám neostanú žiadne uzli v opened, funkcia končí a vypisuje, že hlavolam nemá riešenie.

Testovanie

Program bol testovaný na 8-mich vstupoch, jeden z nich je nesplniteľný (2x2, hlavolamy väčších rozmerov by trvali príliš dlho) všetky s použitím oboch heuristík. Výsledky vypadali nasledovne:

	Processed nodes	Nodes left in queue	Reverse time	Reverse proc nodes	Reverse left in que
2x2 Unsolvable:					
Almost imediatly	13	0			
3x2 Depth 21:					
H1: 0.0292s	545	51	0.0528s	545	51
H2: 0.0215s	343	63	0.0436s	343	63
3x3 Depth 20:					
H1: 2.7567s	6,368	4,013	2.9105s	6,305	3,997
H2: 0.0687s	703	477	0.0905s	609	431
3x3 Depth 27:					
H1: Too long	?	?	?	?	?
H2: 1.7066s	6,050	3,700	4.8836s	10,586	6,412
3x3 Depth 31:					
H1: Too long	?	?	?	?	?
H2: 59.6089s	33,772	19,403	189.319s	54,023	28,019
4x2 Depth 20:					
H1: 0.2890	2,121	843	0.2932s	2,146	803
H2: 0.0413	307	145	0.0366s	194	99
4x2 Depth 30:					
H1: 52.0539s	30,393	6,070	62.3449s	30,346	6,071
H2: 1.2606s	6,256	2,282	1.2827s	6,412	2,248
4x2 Depth 36					
H1: Too long	?	?	?	?	?
H2: 26.8295s	36,454	5,571	29.6550s	36,454	5571

H1 je heuristika, súčet kameňov na nesprávnom mieste, H2 reprezentuje súčet vzdialeností od cieľových pozícií. Všetky tri zaznamenané údaje kontrolujeme aj pri hľadaní opačným smerom.

Analýza testovania

V mojej implementácií je druhá heuristika omnoho rýchlejšia nie len na základe toho, že je presnejšia, ale aj kvôli spôsobu adresovania v programe, kde výsledok heuristiky slúži ako kľúčová hodnota v denníkoch. Pri vytváraní programu som originálne používal obyčajné polia,

pri porovnaní heuristík bola druhá stále rýchlejšia no rozdiel neboľ až tak výrazný (starú verziu programu som nechal v adresári misc).

Pri prehodení začiatočných a konečných stavov sa výsledky zväčša zhoršili.