

Inteligența Artificială

Proiect 1

Radulescu Mihai-Alexandru

Descrierea proiectului

În acest proiect este implementată rezolvarea problemei “Micul Vrajitor”. Cerința descrie cum un vrajitor intră într-o peșteră, găsește piatra magică și părăsește peștera.

Implementarea a fost realizată în limbajul *Python 3.9.4*, utilizând 3 fișiere:

- `main.py` : Conține funcția principală care porneste execuția și funcții ajutoare pentru citirea unui folder, citirea unui fișier, crearea unui folder cu soluții (dacă nu există deja) și scrierea rezultatelor în fișiere.
- `alg.py` : Conține funcții pentru algoritmi implementați și pentru euristici.
- `state.py` : Conține clasa `State` cu metodele sale și o funcție care setează datele inițiale.

Structura datelor de intrare

Datele de intrare se vor pune într-un folder sub forma de fișiere text. În acel fișier se vor scrie 2 matrici. Prima reprezentând culoarea pantofilor care trebuie utilizați pentru a se muța pe acea poziție și a doua reprezentând intrarea în peșteră, locația pietrei și pantofi de rezervă

```

a a a a a a a
a b b b b b a
a b a a a b a
a b a b a b a
a a a b a a a

a a a a a a a
* a a a a a @
a a a a a a a
a a a a a a a
a a a a a a a

```

Mod de utilizare

Programul poate fi pornit lansand in Python fisierul *main.py* si urmand instructiunile din consola.

Programul se mai poate rula modificand variabilele din fisierul *test.py* si lansand fisierul *test.py* in Python.

```

66  def main():
67      # Program start
68
69      # Input folder
70      folder = "input"
71      output_folder = "output"
72      id_alg = 3
73      id_fn = 1
74      nr_solutions = 3

```

Reprezentarea posibilelor stari ale unui jucator

Pentru reprezentare starilor se utilizeaza o clasa State, gasita in fisierul *state.py*. O stare pastreaza urmatoarele date:

- pozitia actuala pe harta
- pantofii purtati de vrajitor si numarul de utilizari ai acestora
- pantofii din rucsac si numarul de utilizari ai acestora
- starea precedenta
- daca vrajitorul se afla in posesia pietrei magice sau nu
- o lista cu mutarile realizate pana in acel moment

Creearea unui obiect de tip State se realizeaza prin apelarea constructorului de initializare:

```

states = [state.State(state.start, None, 0,
(state.map_colors[state.start[0]][state.start[1]], 1), None, False,
["Start"])]

```

Generarea unei stari succesori

Pentru generarea unui succesori se apeleaza metoda succesori() din clasa State. Aceasta metoda ia cele 4 posibile directii unde se poate duce vrajitorul si verifica toate posibilitatile.

Pentru a determina daca vrajitorul a iesit din pestera utilizam metoda endNode(). Aceasta verifica daca pozitia curenta este intrarea in pestera si daca vrajitorul are piatra magica.

Statistici

Timp / Noduri	A*	IDA*	A* Optimizat	UCS
basic.txt	1.327 / 2384	0.818 / 28595	0.034 / 152	0.139 / 2384
sol.txt	0.001 / 60	0.008 / 437	0.001 / 39	0.001 / 60