



Први пројектни задатак из предмета Интелигентни системи школске 2024/25. године

Основне информације

Lost in space је графичка симулација написана на програмском језику Пајтон која приказује рад основних алгоритама претраживања. Симулацију чини мапа поља у свемиру по којима се крећу свемирски бродови користећи унапред дефинисану претрагу. На појединим пољима на мапи се налазе паркинг места до којих је потребно спровести свемирске бродове. Циљ симулације је довести све свемирске бродове до паркинг места.

Свемирски бродови се могу померити у било ком смеру и тада они путују по проходним пољима док не ударе у поље препреке или у други свемирски брод где се и заустављају. У ту свхру потребно је имплементирати неколико алгоритама који решавају дати проблем.

Начин коришћења апликације

Програм се покреће из терминала навођењем следеће команде:

```
python .\main.py algorithm map timeout
```

где су:

- `.\main.py` – путања до главног Пајтон фајла са изворним кодом
- `algorithm` – назив алгорита који се користи (подразумевано `ExampleAlgorithm`)
- `map` – назив мапе која се користи (подразумевано `example_map.txt`)
- `timeout` – максимално време за извршавање (подразумевано 0; неограничено)

Пре покретања потребно је инсталирати пакет `pygame` у оквиру Пајтон интерпретера. Након покретања приказује се главни прозор апликације:



Слика 1 – Почетни екран

Приликом покретања програма извршава се одабрани алгоритам, а потом приказује прозор апликације. У оквиру прозора приказана је мапа свемирске станице по којој се свемирски бродови крећу. Притиском на дугме *SPACE* могуће је покренути или привремено зауставити приказ решења. Притиском на дугме *ENTER* могуће је приказати коначно решење. Притиском на дугме *ESC* прекида се рад апликације и затвара се главни прозор.



Мапа

Мапа је текстуална датотека која садржи матрични приказ поља, препрека, свемирских бродова и паркинг места у произвољном броју редова. Сматрати да је број знакова у сваком реду једнак.

Пример фајла мапе:

```
__ 0  
_S_  
_G_  
__ 0
```



На мапи се могу наћи поља дата у табели:

Назив	Слика	Ознака у мапи	Опис
Празно поље		—	Поље по којима се свемирски бродови могу кретати.
Препрека		O	Поље које зауставља кретање свемирског брода.
Свемирски брод		S	Свемирски брод.
Паркинг место		G	Циљ.



Алгоритми система

У систему постоји неколико типова свемирских бродова, сваки са својом стратегијом претраживања:

- *Blue*. Плави свемирски бродови користе стратегију претраге по дубини, а редослед обиласка суседа приликом претраге одређен је странама света: север, исток, југ, запад.
- *Red*. Црвени свемирски бродови користе стратегију претраге по ширини, а редослед обиласка суседа приликом претраге одређен је странама света: север, исток, југ, запад.
- *Black*. Црни свемирски бродови користе стратегију претраге са гранањем и ограничавањем, при чему је цена сваког померања свемирског брода до заустављања одређена бројем пређених поља, а редослед обиласка суседа у случају да два или више суседа имају једнаку евалуацију одређен је странама света: север, исток, југ, запад.
- *White*. Бели свемирски бродови користе A* стратегију претраживања са *Manhattan* дистанцом као хеуристичком функцијом, при чему је цена сваког померања свемирског брода до заустављања одређена бројем пређених поља, а редослед обиласка суседа у случају да два или више суседа имају једнаку евалуацију одређен је странама света: север, исток, југ, запад.



Кориснички захтеви

Студенти треба да имплементирају стратегије претраге проширивањем класе `Algorithm` (чија дефиниција треба да се нађе у датотеци `algorithms.py`) и редефинисањем њене функције `get_path`, чији параметар `state` представља стање система, а повратна вредност је коначна листа акција које доводе до циљног стања. Њу чини листа акција, где је свака акција представљена торком почетног и крајњег положаја свемирског брода (`(src_row, src_col), (dst_row, dst_col)`). Студентима је у оквиру кода дат пример класе (`ExampleAlgorithm`) која проширује основну класу `Algorithm` и на једноставан начин имплементира функцију `get_path`.

У наставку је дат пример покретања програма са датом пример класом:

```
python .\main.py ExampleAlgorithm example_map.txt 5
```

Имати у виду да се на одбрани домаћег задатка могу појавити нове мапе, алгоритми и хеуристике.

Имплементација се ради у програмском језику Пајтон уз дозвољено коришћење свих стандардних библиотека и структура података.



Универзитет у Београду - Електротехнички факултет
Катедра за рачунарску технику и информатику

Напомене

Електронску верзију решења овог домаћег задатка послати најкасније до 14. новембра 2024. године до 15 часова, као ZIP архиву на следећем [линку](#). Одбрана домаћег задатка планирана је 15. новембра 2024. године.

Домаћи задатак из предмета *Интелигентни системи* се ради самостално и није обавезан за полагање испита (на испиту се може заменити са теоријским питањима из целокупног градива). Овај домаћи задатак се може бранити само у за то предвиђеном термину. Домаћи задатак вреди максимално 10 поена.

На усменој одбрани кандидат мора самостално да покрене своје решење. Кандидат мора да поседује потребан ниво знања о задатку, мора да буде свестан недостатака приложеног решења и могућности да те недостатке реши. Кандидат мора тачно да одговори и на одређен број питања која се баве тематиком домаћег задатка.

Евентуална питања послати асистентима на мејл, али као једну поруку, а не две одвојене (другог асистента обавезно ставити у копију - СС поруке).

aki@etf.rs

jocke@etf.rs