



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

Основне информације

Pynter је графичка симулација написана на програмском језику Пајтон која приказује рад основних алгоритама теорије секвенцијалних игара. Симулацију чини мапа поља у свемиру по којима се крећу свемирски бродови. Преласком преко поља, свемирски брод осваја поље и боји га у своју боју. Игра се завршава када се сва поља обоје или када прође задат број рунди. Циљ игре је имати највише обојених поља.

У свом потезу, свемирски бродови могу да одлуче да се не помере, да се помере тачно једно поље у било ком смеру или да започну померање у било ком смеру које се прекида када дођу до провалије или другог свемирског брода.

Начин коришћења апликације

Програм се покреће из терминала навођењем следеће команде:

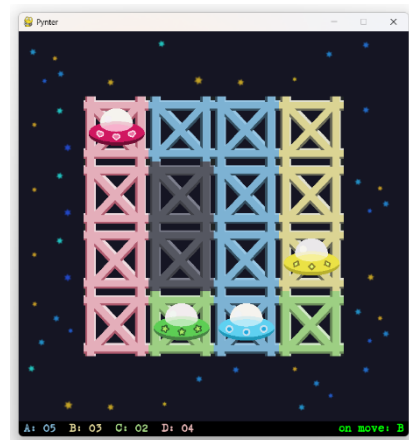
```
python .\main.py agents map rounds timeout max_depth
```

где су:

- `.\main.py` – путања до главног Пајтон фајла са изворним кодом
- `agents` – називи агената који се користе, одвојени зарезима (подразумевано сви агенти су `RandomAgent`)
- `map` – назив мапе која се користи (подразумевано `example_map.txt`)
- `rounds` – максималан број рунди игре (подразумевано 5)
- `timeout` – максимално време за извршавање (подразумевано 0; неограничено)
- `max_depth` – максимална дубина развијања стабла (подразумевано 5)

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

У оквиру прозора приказана је мапа свемирске станице по којој се свемирски бродови крећу, испод које се налази секција са информацијама о тренутном резултату и тренутном играчу на потезу. Притиском на дугме *SPACE* могуће је покренути или привремено зауставити игру. Притиском на дугме *ESC* прекида се рад апликације и затвара се главни прозор.



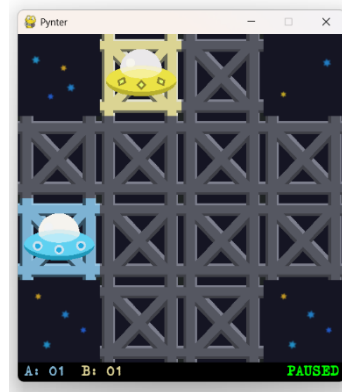


Mana

Мапа је текстуална датотека која садржи матрични приказ поља, провалија, свемирских бродова и обојених поља. Сматрати да је број знакова у сваком реду једнак.

Пример фајла мапе:

```
0B_0
----
A___
0__0
```



На мапи се могу наћи поља дата у табели:

Назив	Слика	Ознака у мапи	Опис
Провалија		0 (нула)	Поље по ком се свемирски бродови не могу кретати.
Слободно поље		—	Поље по ком се свемирски бродови могу кретати.
Обојено поље		a, b, c, d	Поље по ком се свемирски бродови могу кретати и које је иницијално обојено.
Свемирски брод		A, B, C, D	Свемирски брод.



Агенти система

У систему постоје имплементирани следећи агенти:

- RandomAgent. Агент насумично повлачи потезе.
- GreedyAgent. Агент повлачи потезе који му локално доносе највећи број поена.

Потребно је имплементирати следеће агенте:

- MaxNAgent. Агент користи *MaxN* алгоритам за више рационалних играча.
- MinimaxAgent. Агент користи *Minimax* алгоритам за два рационална играча.
- MinimaxABAgent. Агент користи *Minimax* алгоритам за два рационална играча уз коришћење алфа-бета одсецања.

У првом алгоритму функција процене стања представља торку резултата свих агената. У следећа два алгоритма функција процене стања се рачуна као разлика резултата агента и противника.

Кориснички захтеви

Студенти треба да имплементирају алгоритме секвенцијалних игара проширивањем класе *Agent* (чија дефиниција треба да се нађе у датотеци *agents.py*) и редефинисањем њене функције *get_chosen_action*, чији параметри *state* и *max_depth* представљају стање игре и максималну дубину развијања стабла, а повратна вредност је акција која представља следећи потез играча. Акција је представљена торком почетног и крајњег положаја свемирског брода ((*src_row*, *src_col*), (*dst_row*, *dst_col*)). Студентима је у оквиру кода дат пример две класе (*RandomAgent*, *GreedyAgent*) које проширују основну класу *Agent* и на једноставан начин имплементирају функцију *get_chosen_action*.

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

```
python .\main.py RandomAgent,RandomAgent example_map.txt 10 0 5
```

Студентима се предлаже да провере решење на различитим мапама, временима за доношење одлуке и максималним дубинама за развијање стабла. 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 као ZIP архиву на следећем [линку](#). Одбрана домаћег задатка планирана је у сваком испитном року ове школске 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 или рачунским задацима из области машинског учења). Овај домаћи задатак се може бранити само у за то предвиђеном термину. Домаћи задатак вреди максимално 15 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