



Други пројектни задатак из предмета Интелигентни системи школске 2023/24. године

Основне информације

Pyvezi4 је графичка симулација секвенцијалне игре „Connect4” написана на програмском језику Пајтон. Играчи наизменично спуштају жетоне своје боје у мрежу од шест редова и седам колона, заузимајући најнижи расположиви простор унутар одабране колоне. Циљ игре је бити први играч који је формирао хоризонталну, вертикалну или дијагоналну линију од четири сопствена жетона.

Начин коришћења апликације

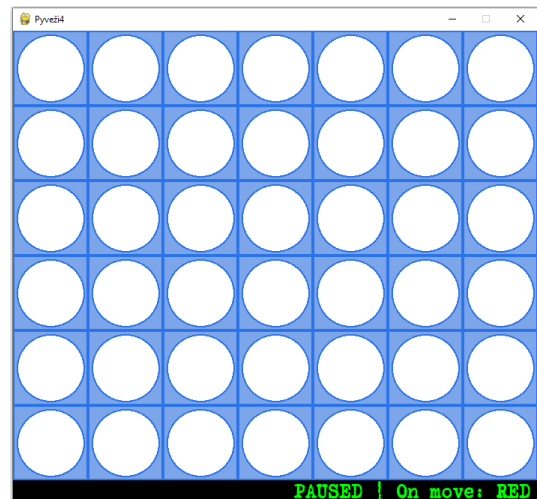
Програм се покреће из терминала навођењем следеће команде:

```
.\main.py agent_1 agent_2 max_depth max_think_time state
```

где су:

- `.\main.py` – путања до главног Пајтон фајла са изворним кодом
- `agent_1` – назив класе агента који се користи за играча који је први на потезу (подразумевано `Human`)
- `agent_2` – назив класе агента који се користи за играча који је други на потезу (подразумевано `ExampleAgent`)
- `max_depth` – целобројни параметар који представља максималну дубину развијања стабла игре (подразумевано 0, што означава игру без ограничења)
- `max_think_time` – целобројни параметар који представља максимално време у секундама за доношење одлуке (акције агента) (подразумевано 10 секунди)
- `state` – име текстуалног фајла са конфигурацијом стартног стања игре (подразумевано се креће од празне мреже)

Пре покретања потребно је инсталирати пакет `pygame` у оквиру Пајтон интерпретера. Након покретања приказује се главни прозор апликације. У оквиру прозора приказана је мрежа поља у које је могуће спустити жетон, испод које се налази секција са информацијама о тренутном потезу игре и протеклом времену размишљања агента који је на потезу. Уколико је један од играча човек (*Human*), притиском миша на одговарајућу колону, спушта се жетон. Притиском на дугме `ESC` прекида се рад апликације и затвара се њен главни прозор. Притиском на дугме `SPACE` привремено се зауставља или наставља рад апликације.



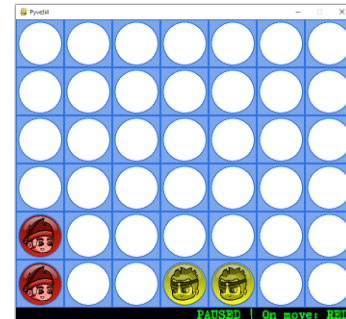


Конфигурација стартног стања

Стартно стање може да се конфигурише избором текстуалне датотеке која у произвољном броју редова садржи информације о одиграним потезима који доводе до тог стања (редом се наводе колоне у које су спуштани жетони):

Пример фајла:

0
3
0
4



Алгоритми

Потребно је имплементирати следеће агенте:

- MinimaxAgent. Представља реализацију *Minimax* алгоритма за два рационална играча уз коришћење алфа-бета одсецања.
- Negascout. Представља реализацију *Negascout* алгоритма за два рационална играча.

Агенти користе функцију процене стања која представља разлику две вредности. Прва вредност се рачуна као број места на којима је могуће остварити победу. Друга вредност се рачуна као број места на којима противник може да оствари победу. Такође, већим вредностима је потребно наградити победе остварене мањим бројем жетона.

Редослед експанзије чворова наследника текућег чвора радити према вредности функције процене стања опадајуће, а у случају два или више стања са истом вредношћу функције процене предност давати колонама ближим средини (3, 2, 4, 1, 5, 0, 6).

- CompetitiveAgent. Реализација агента је у потпуности препуштена студентима и није потребна за максимални број поена на овом домаћем задатку.

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_column`, чији параметри `state` и `max_depth` представљају тренутно стање игре за које се тражи следећа акција и максималну дубину развијања стабла, а повратна вредност је колона која је одабрана за спуштање жетона. Студентима је у оквиру кода дат пример класе (`ExampleAgent`) која проширује основну класу `Agent` и на једноставан начин имплементира функцију `get_chosen_column`.

Студентима се предлаже да провере решење са различитим почетним стањима, временима за доношење одлуке и максималним дубинама за развијање стабла. Имати у виду да се на одбрани домаћег задатка могу појавити нови агенти и функције процене. 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 21. децембра 2023. године до 21 часа, као ZIP архиву на следећем [линку](#). Одбрана домаћег задатка планирана је 22. децембра 2023. 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