



## Домаћи задатак из предмета Интелигентни системи за јануарски рок школске 2018/19. године

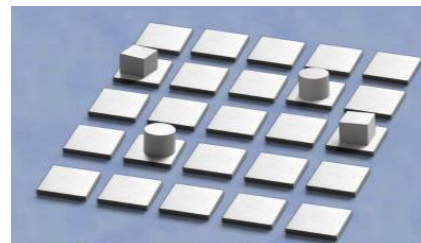
### Основне информације

**Санторини** је друштвена игра на табли за два играча који наизменично повлаче потезе. Сваки играч има две фигуре и неограничен број плочица и сферних купола. Циљ игре је довести једну од сопствених фигура до поља са три плочице, или довести противника у ситуацију да не може да помери ниједну од својих фигура.

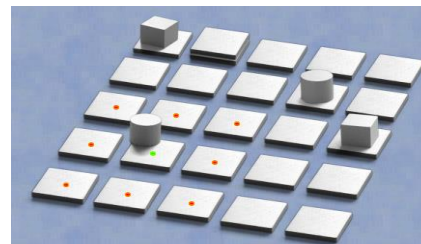


### Правила игре

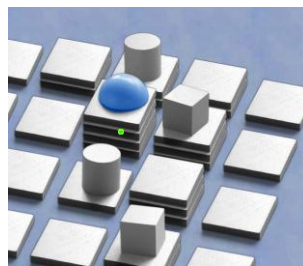
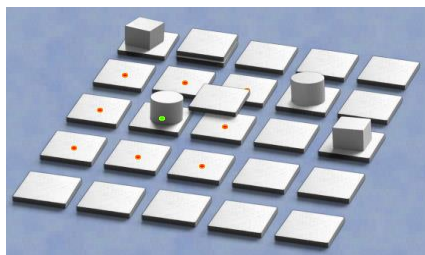
- Игра се игра на табли 5x5, која се иницијално састоји из 25 празних поља. Игру играју два играча, који повлаче потезе наизменично. На почетку игре, први, а затим и други играч постављају своје фигуре на било које од слободних поља. Након постављања фигура, играчи наизменично играју повлачећи потезе.



- Потез играча се састоји из померања фигуре и градње. Најпре се једна одабрана фигура помера на неко од слободних суседних поља (укључујући дијагоналне суседе, при чему одредишно поље мора бити највише једну плочицу изнад изворишног поља. Уколико играч не може да помери ниједну од својих фигура према овим правилима, губи игру.

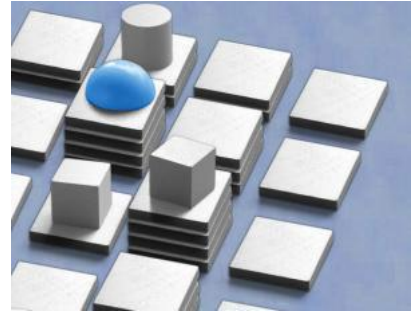


- Други део потеза је градња. Ова фаза подразумева додавање плочице на једно од слободних поља суседних одредишном пољу померене фигуре (укључујући дијагоналне суседе). Плочица која се додаје на поље нивоа 3 је кружна купола. Поља нивоа 4 (3 плочице и кружна купола на врху) су „блокирана“ у наставку игра. Није дозвољено постављање фигура на таква поља, нити даља изградња над тим пољима.





- Игра се може завршити на два начина:
  - када играч који је на потезу помери једну од фигура на поље нивоа 3 (изграђене три плочице на том пољу) – победник је играч који је померио своју фигуру на поље нивоа 3;
  - када играч који је на потезу не може да одигра потез према правилима игре – победник је играч који није на потезу.



### *Кориснички захтеви*

- Изглед корисничког интерфејса

У сваком тренутку треба да буде назначено који играч је на потезу и приказана табла за игру, са јасним информацијама о стању појединачних поља и позицијама фигура.

- Покретање игре

Пре почетка игре неопходно је:

- одабрати ко игра игру (човек против човека, човек против рачунара или рачунар против рачунара),
- омогућити читање стања игре из фајла, у виду низа потеза који се одигравају до посматраног тренутка (пример улазног фајла дат је у једној од наредних секција),
- одабрати тежину игре – дубину развијања стабла игре, у случају да је бар један од играча рачунар.

- Режији рада

За игру коју игра рачунар против рачунара, омогућити два режима рада:

- корак по корак, где корисник може да испрати комплетан рад алгоритма, уз приказане вредности функције процене за релевантна поља;
- извршавање имплементираних стратегија до краја.

Без обзира на одабран режим рада, решење додатно записати у виду излазног фајла са евиденцијом о низу одиграних потеза.



- Евиденција одиграних потеза

Излазни фајл са евиденцијом о низу одиграних потеза треба да има исти формат као улазни фајл из ког се чита стање игре. Редови и колоне табле за игру нумерисани су словима А-Е и бројевима 1-5 ради означавања позиција фигура. Прве две линије фајла садрже информације о почетном распореду фигура за првог и другог играча. Остатак фајла садржи информације о потезима, сваки потез у засебном реду. У наставку је дат пример оваквог фајла:

```
C3 B2 //постављају се фигуре првог играча на означена поља
B2 D2 //постављају се фигуре другог играча на означена поља
C3 B3 A3 //помера се фигура са поља C3 на поље B3, уз изградњу плочице на пољу A3
C2 D3 E3 //помера се фигура са поља C2 на поље D3, уз изградњу плочице на пољу E3
```

- Решавање проблема

Проблем је потребно решити *Minimax* алгоритмом.

Потребно је имплементирати једноставног играча. Овај играч треба да користи једноставну статичку функцију процене  $f$ , која се рачуна као  $f = m + l$ , где је  $m$  број плочица одредишног поља, а  $l$  број нивоа на које се додаје плочица помножен разликом растојања сопствених и противничких играча од тог поља.

Након креирање једноставног играча, потребно је имплементирати једноставног алфа-бета играча, који побољшава једноставног играча тако што приликом претраживања користи алфа-бета одсецање.

Након креирања једноставног алфа-бета играча, потребно је направити такмичарског алфа-бета играча, који треба да оптимизује перформансе претходног. Можете укључити било које технике које желите (изузев нити, GPU програмирања или других сличних ствари) за реализацију овог играча. Почните од надоградње постојећег алфа-бета играча, коришћењем боље статичке функције процене, која ће користити више информација из стања игре.

- Имплементација игре

Препоручено је да се имплементација ради у програмском језику Јава или C++/C#, у развојном окружењу *Eclipse IDE* или *VisualStudio*, уз дозвољено коришћење свих стандардних библиотека и структура података. Дозвољено је имплементирати систем и у неком другом програмском језику по жељи (*Android*, *Swift for iOS*, ...) уз одобрење предметних асистената.



Главни програм, алгоритме, основне структуре и режиме рада симулације, одвојити у засебне класе. Сваки студент треба да направи свој пакет (или *namespace*) `etf.santorini.<username>` где је `<username>` Ваше корисничко име за портал е-Студент (у формату *piGGBBBBx*, где су *pi* иницијали - презиме и име, ознака *GG* представља последње две цифре године уписа факултета, ознака *BBBB* представља четвороцифрени број индекса, проширен водећим нулама, а ознака *x* представља ниво студија *d* или *m*). Студенти су обавезни да у том пакету имплементирају комплетан код, а не ван њега.

- Корисници интерфејс треба да буде интуитиван и довољно детаљан приликом решавања проблема.

### ***Захтеви домаћег задатка***

Ваш задатак је да реализуете игру Санторини коришћењем алгоритама теорије игара. За максималан број поена, програм мора да испуни све услове пројектног задатка и да ради поуздано.

Поред тога, треба написати извештај, који описује како је проблем решен. Извештај треба да садржи:

- Кратак приказ имплементираних алгоритама и њихових функција;
- Кратке описе свих реализованих класа, са потписима и описима метода.

Извештај приложити уз код, у фолдеру `/documentation` и именовати га као: `Prezime_Ime_IS_Domaci_Jan2019.pdf`.

### **Напомене**

Електронску верзију решења овог домаћег задатка предати најкасније до 11. јануара 2019. године у 23:59, као ZIP архиву, на [сајту за предају домаћег](#). Термин одбране домаћег задатка биће накнадно објављен, највероватније у недељи пре одржавања испита.

Домаћи задатак из предмета *Интелигентни системи* се ради самостално и није обавезан за полагање испита (на испиту се може заменити теоријским питањима из целокупног градива). Овај домаћи задатак се може бранити **само у јануарском испитном року**. Домаћи задатак вреди максимално до 20 поена (уз могућност додатних поена за радове који се посебно истакну).

На усменој обрани кандидат мора самостално да покрене своје решење, које је предао до задатог рока за израду. 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.

Пре започињања реализације проблема или тражења помоћи задатак прочитати у целини и пажљиво. Уколико у задатку нешто није довољно прецизно дефинисано, од кандидата се очекује да уведу разумне претпоставке и темељно их образложе.



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

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

[drazen.draskovic@etf.bg.ac.rs](mailto:drazen.draskovic@etf.bg.ac.rs),

[dragana.m@etf.bg.ac.rs](mailto:dragana.m@etf.bg.ac.rs)

**Додатни материјали**

- [https://en.wikipedia.org/wiki/Santorini\\_\(game\)](https://en.wikipedia.org/wiki/Santorini_(game))
- <http://www.boardspace.net/santorini/english/santorini-rules.html>