

Drugi domaći zadatak iz predmeta Vještačka inteligencija
02.11.2020.

Ovaj domaći zadatak odnosi se na definisanje i rješavanje različitih problema traženja na primjeru igrice Pac-Man. Za rješavanje ovog zadatka biće Vam potreban kod koji možete preuzeti na adresi [Berkeley AI Materials](#). Materijali koji se nalaze na ovom sajtu su dostupni svima i mogu se koristiti u edukativne svrhe.

1. U fajlu **search.py** implementirati A* algoritam u funkciji **aStarSearch**, koja kao argumente uzima heurističku funkciju i problem traženja. U ovom zadatku nije potrebno definisati sopstvenu heuristiku. Sve potrebne strukture podataka su implementirane u fajlu **util.py**. Rješenje možete testirati na problemu traženja putanje kroz lavirint (problem sa vježbi), koristeći već definisanu Manhattan distance heuristiku:

```
python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
```

Cijena putanje, u ovom slučaju, treba da bude 210 sa oko 549 proširenih čvorova.

2. U fajlu **searchAgents.py** implementirati klasu **CornersProblem** kojom se definiše problem traženja najkraće putanje koja prolazi kroz sva četiri ugla datog lavirinta. Ova klasa treba da bude implementirana tako da se prethodno implementirani algoritmi traženja (algoritmi sa vježbi i algoritam iz prvog zadatka) mogu, bez ikakvih izmjena, pokrenuti i na ovom problemu, komandama:

```
python pacman.py -l tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem  
python pacman.py -l mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
```

Veličina definisanog prostora stanja biće uzimana u obzir prilikom bodovanja.

3. U fajlu **searchAgents.py** definisati netrivialnu, konzistentnu heuristiku u funkciji **cornersHeuristic** za problem traženja iz prethodnog zadatka (**CornersProblem**). Rješenje možete testirati komandom:

```
python pacman.py -l mediumCorners -p AStarCornersAgent -z 0.5
```

Bodovanje će se vršiti u skladu sa brojem proširenih čvorova na datom primjeru i to:

- > 2000 proširenih čvorova – 0% poena
- < 2000 proširenih čvorova – 30% poena
- < 1500 proširenih čvorova – 60% poena
- < 1000 proširenih čvorova – 100% poena

4. U ovom zadatku rješava se problem u kojem Pacman treba da pojede svu hranu, koja se nalazi u lavirintu, u što je moguće manje koraka. Klasa **FoodSearchProblem**, koja odgovara ovom problemu traženja, već je implementirana u fajlu **searchAgents.py**. Vaš

zadatak je da implementirate netrivialnu, konzistentnu heuristiku za ovaj problem u funkciji **foodHeuristic** koja se takođe nalazi u fajlu **searchAgents.py**. Rješenje možete testirati komandom:

```
python pacman.py -l trickySearch -p AStarFoodSearchAgent
```

Cijena putanje u ovom primjeru je 60. Bodovanje će se vršiti u skladu sa brojem proširenih čvorova na datom primjeru i to:

- > 12000 proširenih čvorova – 0% poena
- < 12000 proširenih čvorova – 50% poena
- < 9000 proširenih čvorova – 100% poena
- < 7000 proširenih čvorova – bonus poeni
- < 2000 proširenih čvorova – bonus bonus poeni

Hint: Za heuristiku koja proširuje manje od 2000 čvorova možete iskoristiti funkciju **mazeDistance** koja je implementirana u fajlu **searchAgents.py**, ali da bi osvojili bonus poene morate je pozvati konstantan broj puta.

5. Implementirati agenta koji se ponaša pohlepno tako što uvijek pojede tačku koja mu je najbliža. **ClosestDotSearchAgent** je implementiran u fajlu **searchAgents.py**, ali mu nedostaje funkcija koja pronalazi putanju do najbliže tačke **findPathToClosestDot**. Pored ove funkcije, potrebno je implementirati i funkciju **isGoalState** u klasi **AnyFoodSearchProblem**. Rješenje možete testirati komandom:

```
python pacman.py -l bigSearch -p ClosestDotSearchAgent -z .5
```

Ovaj agent nalazi putanju dužine 350, koja očigledno nije optimalna.

Rok za predaju rada je 22. novembar. Predaje se RAR ili ZIP datoteka u kojoj treba da se nalaze fajlovi **search.py** i **searchAgents.py**. Ime datoteke mora biti oblika Ime_Prezime_AI_Domaci02.rar i šalje se kao attachment na mejl kosta@ucg.ac.me sa temom (subjectom) poruke "Ime_Prezime_AI_Domaci02".

Napomena: Svi radovi koji su po procjeni nastavnika prepisani biće ocijenjeni sa 0 bodova. Odbrana domaćih radova biće naknadno zakazana.