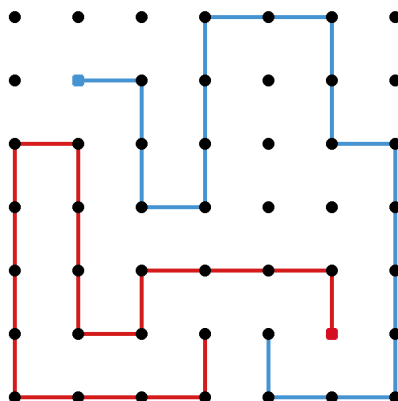


Umělá inteligence - domácí projekt - léto 2022

Cílem domácího projektu je naprogramovat agenta, který hraje hru Had proti oponentovi.

Popis hry je možné nalézt na <http://www.papg.com/show?3AE4>



Hraje se na mřížce fixní velikosti 7 x 7 (po zvážení zmenšeno). Indexujeme od 0, takže políčka mají po obou osách indexy (0..6). Levý horní roh má index [0, 0]. První index udává řádek, druhý sloupec. Hráč 1 začíná na pozici [1, 1], hráč 2 na pozici [5, 5]. Začíná hráč 1 a přidává nový segment z bodu [1, 1] k hadovi, pak následuje hráč 2 a hráči se pravidelně střídají, dokud jeden z nich nemůže táhnout. Hráč, který nemůže dále táhnout prohrál.

Cílem domácího projektu je využít některé umělo-inteligenční techniky představené na přednášce (nebo klidně všechny či jejich kombinaci). Nejde tedy o to popsat rozhodování agenta hrajícího hada pomocí deterministických if-then pravidel.

Snahou by mělo být, aby náš agent hrál přiměřeně inteligentně (není potřeba, aby hrál optimálně). Abychom dokázali vysvětlit naši implementaci a zdůvodnit použité techniky. I případně proč fungují či nefungují.

Není potřeba příliš řešit vykreslování, výstup může být klidně v textové formě.

Cílem je naprogramovat funkci `snake_move(my_snake, other_snake)`, kde `my_snake` je seznam (list) dvojic souřadnic pozic našeho hada tak, jak jsou postupně přidávány s přibývajícími tahy - poslední přidáný segment (nová hlava) je tedy na posledním místě v seznamu, ocas hada je v seznamu na prvním místě. Položka `other_snake` je stejně řešený seznam, akorát reprezentující hada protihráče. Funkce vrací pozici, kam se má rozšířit náš had o novou hlavu.

Výpočet počátečního tahu hráče 1 (pokud za něj hrajeme) tedy probíhá pomocí volání `snake_move([(1, 1)], [(5, 5)])`. Kdybychom hráli jako v příkladu na obrázku, odpovědí funkce by byla dvojice (1, 2).