

Dáma

Huu Quy Nguyen

BI-ZUM

nguyehu7@fit.cvut.cz

16. května 2023

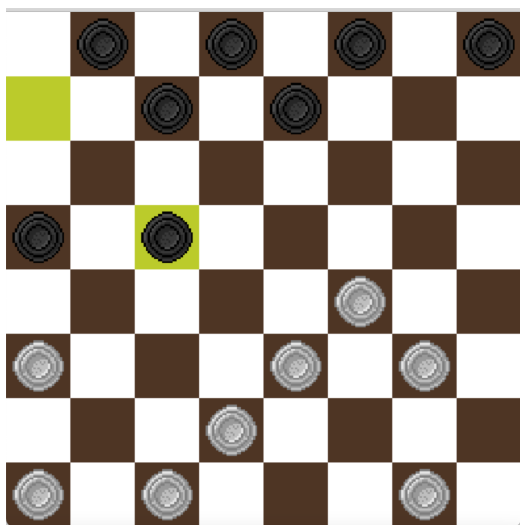
1 Úvod

Tato zpráva pojednává o semestrální práci pro předmět BI-ZUM.

Mým úkolem pro tuto semestrální práci bylo implementovat známou hru dáma a k tomu naprogramovat jednoduchou umělou inteligenci, která ji bude hrát. Umělá inteligence funguje na bázi MiniMax algoritmu a alpha-beta prořezáváním.

2 Pravidla hry

Obyčejná dáma se hraje na čtvercové desce o velikost 8x8. Cílem hry je donutit soupeře, aby ztratil všechny své pěšce nebo aby se dostal do pozice, kdy nemůže provést žádný legální tah.



Obrázek 1: Ukázka hry

3 Umělá inteligence

MiniMax algoritmus se ideálně hodí na hraní dámy. Algoritmus pracuje na základě vyhledávání v herním stromu, který reprezentuje možné tahy a jejich důsledky. Na začátku se projdou listy a spočítají se pro ně hodnoty evaluační funkce. V dalším kroku se vyhodnocuje stav hry pro každý uzel ve stromě. Pro hráče typu Maximizing se vybírá tah s nejvyšším ohodnocením, zatímco pro hráče typu Minimizing se vybírá tah s nejnižším ohodnocením. Dodatečně

alfa-beta prořezávání se využívá k odříznutí větví, které již nemá smysl prohledávat.

Označme si

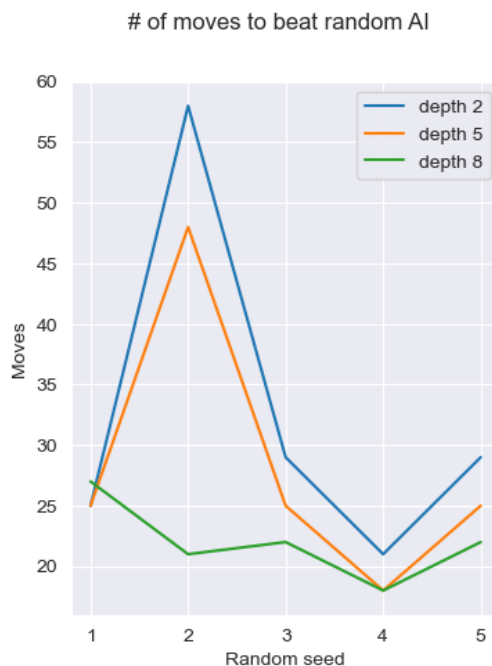
- p_w ... počet bílých pěšců
- p_b ... počet černých pěšců
- q_w ... počet bílých dam
- q_b ... počet černých dam

K evaluaci herního stavu jsem využíval vzorec:

$$eval = 10 \cdot p_w - 10 \cdot p_b + 15 \cdot q_w - 15 \cdot q_b$$

4 Experimenty

K otestování kvality umělé inteligence jsem využil toho, že protistrana hraje náhodné tahy a metrikou je počet tahů, který naše AI udělala. Zkusíme nejdříve testovat kvalitu podle hloubky stromu, který MiniMax algoritmus prohledá. Protihráče budeme držet konzistentně pro 5 různých seedů, které nám zaručí alespoň stejná zahájení jednotlivých her.



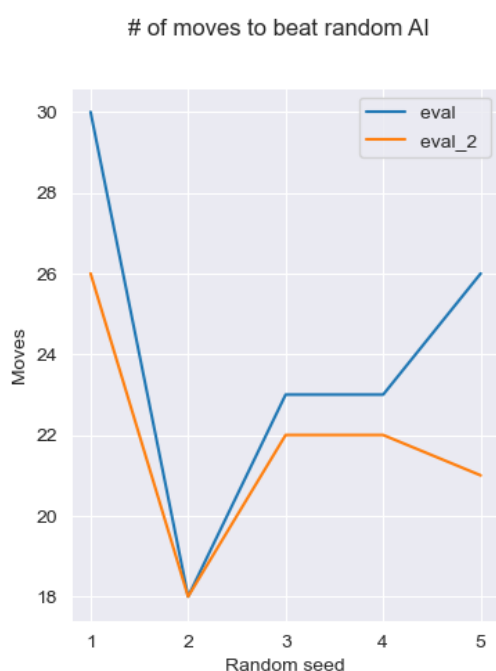
Obrázek 2: Vliv hloubky

Vidíme, že nižší hloubka má vskutku velký vliv na kvalitu modelu. Můžeme předpokládat, že by s rostoucí hloubkou klesal nutný počet tahů nadále.

Pokusme se nyní experimentovat jinou evaluační funkcí a pozorovat, zda se výsledky změní. Můžeme zkusit dávat vyšší hodnotu pěšcům, kteří jsou blíže k tomu, aby se stali dámou.

Označme si nyní $d(x)$... vzdálenost pěšce x od finálního řádku.

$$eval_2 = eval + \sum_{\forall \text{ bílý } p} (8 - d(p)) - \sum_{\forall \text{ černý } p} (8 - d(p))$$



Obrázek 3: Vliv evaluační funkce

Vidíme, že modifikací evaluační funkce jsme jemně vylepšili naši AI. Je pravděpodobné, že jiné možné modifikace by ji zlepšily ještě více.

5 Závěr

V semestrální práci jsem se naučil implementaci MiniMax algoritmu, což je velmi zajímavá zkušenost. Následné experimenty mě bavili, neboť bylo příjemné se kochat vlastním výtvořem. Jsem si však jistý, že by bylo možné implementaci AI vylepšit především modifikací evaluační funkce. Dalo by se vzít v potaz např. počet políček, které figurky kontrolují, počet možných tahů v dané pozici, či aktivita figurek.

Reference

- [1] Pete Shinnars. Pygame. <http://pygame.org/>, 2011.

- [2] W3Schools. W3schools online web tutorials. <https://www.w3schools.com/>.
- [3] Wikipedia contributors. Alpha-beta pruning. <https://en.wikipedia.org/wiki/Alpha>
- [4] Wikipedia contributors. Minimax. <https://en.wikipedia.org/wiki/Minimax>.