**Sztuczna inteligencja i inżynieria wiedzy**

Algorytmy rozwiązywania gier

Maciej Kopiński

254578

**Zbadanie wyboru algorytmu, a czasu i liczby ruchów**

Tabela dla przeciwnika: minmax – głębokość 3 (gracz biały)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ustawienia algorytmu | Liczba ruchów wygranego | Czas dla gracza białego | Czas dla gracza czarnego | Wygrany |
| minmax – głębokość 2 | 36 | 2,7s | 0,39s | biały |
| minmax – głębokość 3 | 43 | 1,88s | 2,13s | remis |
| minmax – głębokość 4 | 25 | 1,10s | 6,17s | czarny |
| alphabeta – głębokość 2 | 47 | 0,97s | 0,25s | remis |
| alphabeta – głębokość 3 | 27 | 1,26s | 0,96s | czarny |
| alphabeta – głębokość 4 | 24 | 1,04s | 2,4s | czarny |

Wnioski:

Pierwszą rzeczą, którą możemy zauważyć jest różnica w czasie wykonania dla algorytmu minmax i alphabeta, co jest oczywiście spowodowane „obcinaniem” gałęzi w drzewie poszukiwań.

Drugą rzeczą, którą zauważyłem jest spadek średniej liczby ruchów wraz ze wzrostem głębokości drzewa poszukiwań, co również wydaje się zrozumiałe.

**Zbadanie wpływu heurystyk na czas i liczbę ruchów**

Tabela dla przeciwnika: minmax – głębokość 3 i połączone heurystyki (gracz biały)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pionki białe – pionki czarne | | Różnica pionków + pozycja | |
|  | Liczba ruchów | Czas | Liczba ruchów | Czas |
| minmax – 2 | 30 | 0,39 | 30 | 0,37s |
| minmax – 3 | 27 | 1,52s | 30 | 1,6s |
| minmax – 4 | 40 | 7,8s | 47 | 14,8s/7s |
| alphabeta – 2 | 32 | 0,38s | 44 | 0,4s |
| alphabeta – 3 | 29 | 0,89s | 30 | 0,75s |
| alphabeta - 4 | 34 | 2,07s | 27 | 2,5s |

Wnioski:

Gra bardzo często kończyła się poprzez brak ruchów do wykonania przez gracza białego. Przy niższych głębokościach przeszukiwania drzewa, liczba ruchów mogła się od siebie bardzo dużo różnić, natomiast dla większych głębokości wyniki są zbliżone do siebie.