

# Projektowanie algorytmów i metody sztucznej inteligencji

## Projekt 2 - Grafy - sprawozdanie

Igor Zieliński 248944  
Prowadzący: mgr inż. Marcin Ochma

Wtorek 15<sup>15</sup> – 16<sup>55</sup>

### 1 Wprowadzenie

Celem ćwiczenia było napisanie gry dwuosobowej oraz zaimplementowanie algorytmu sztucznej inteligencji, która umożliwi użytkownikowi grę z komputerem.

### 2 Wybór gry i algorytmu

Zdecydowałem się na implementację szachów z użyciem algorytmu MINMAX wzbogaconego o cięcia alfa-beta.

Efekt końcowy jest uproszczoną wersją szachów. Nie ma mechanizmu szach-mat, gra kończy się, gdy zbity zostanie król przeciwnika. Nie licząc tej różnicy, gra zachowuje się jak zwykłe szachy, dostępne są wszystkie typowe ruchy włącznie z roszadami i promocją piona. W zaimplementowanej wersji szachów gracz posiada zawsze figury w kolorze białym, a komputer - czarnym.

### 3 Funkcja oceniająca algorytmu MINMAX

Funkcja ta zdefiniowana jest w taki sposób, że gracz czarny jest graczem minimalizującym(komputer), natomiast gracz biały maksymalizującym. Każdemu rodzajowi figury przyporządkowano punkty za posiadanie tejże figury na planszy, są to kolejno.

- pionek - 100pkt.
- wieża - 500pkt.
- skoczek - 320pkt.
- gонец 330pkt.
- hetman - 900pkt.
- król - 20000pkt.

Ponadto, dla każdego rodzaju figury zdefiniowano ilość punktów za położenie na planszy.

```
int array_pawn[8][8] = { { 0, 0, 0, 0, 0, 0, 0, 0},  
                          {50, 50, 50, 50, 50, 50, 50, 50},  
                          {10, 10, 20, 30, 30, 20, 10, 10},  
                          { 5, 5, 10, 25, 25, 10, 5, 5},  
                          { 0, 0, 0, 20, 20, 0, 0, 0},  
                          { 5, -5, -10, 0, 0, -10, -5, 5},  
                          { 5, 10, 10, -20, -20, 10, 10, 5},  
                          { 0, 0, 0, 0, 0, 0, 0, 0} };
```

Rysunek 1: Przykładowa definicja punktów za położenie dla pionka - zachęca to zmierzania w kierunku pola, które gwarantuje promocję.

Funkcja dla każdej figury na planszy, jeśli jest koloru czarnego, odejmuje od końcowej wartości funkcji ilość punktów za posiadanie figury na planszy oraz za położenie figury na planszy. Jeśli jest to figura koloru białego, wymienione wartości są dodawane.

## 4 Głębokość drzewa MINMAX i czas działania

Ponieważ zastosowano cięcia alfa-beta, czas obliczeń jest nierówny i zależy od sytuacji panującej na planszy. Z testów, które przeprowadziłem wynika, że maksymalna głębokość, która gwarantuje komfort gry to 4 i na takiej głębokości skompilowałem program. Maksymalny czas oczekiwania na ruch w takiej konfiguracji to około dwie sekundy, przy czym w zdecydowanej większości przypadków, ruch zajmuje mniej, niż pół sekundy.

## 5 Interfejs graficzny i sterowanie

Okno aplikacji zawiera wyłącznie planszę z ustawionymi figurami



Rysunek 2: Początkowy układ planszy



Rysunek 3: Plansza w trakcie rozgrywki

Sterowanie odbywa się w następujący sposób: Klikamy na pole z figurą, którą chcielibyśmy się ruszyć, następnie klikamy na pole, na które chcielibyśmy się ruszyć. W przypadku roszady, odtwarzamy ruch króla. Jeśli chcemy ruszyć się inną figurą, niż zaznaczyliśmy, prawy przycisk myszy anuluje wybór.