# Projektowanie Algorytmów i Metody Sztucznej Inteligencji Projekt 4 - GraAI

Szymon Zajda 248917 Prowadzący: mgr.inż Marcin Ochman W<br/>T $15^{15}\,$ 

05.06.2020

#### 1 Wprowadzenie

Celem tego projektu było napisanie jednej z wymienionych gier w oparciu o sztuczną inteligencję. Stworzenie komputerowego przeciwnika miało odbyć się poprzez zaimplementowanie algorytmu "minimax".

### 2 Opis programu

Opisywana gra to "kółko i krzyżyk". Całe ciało projektu znajduje się w pliku game.cpp, gdzie wykonywane sa metody klasy Game. Oprócz wspomnianej struktury algorytm wykorzystuje strukturę Move w której znajdują się współrzędne pola(x,y). Dla przypisania kolejności ruchu wykorzystano wyliczeniowy typ danych - enum. Konstruktor Game() inicjuje wartości pola oraz alokuje pamięć przed rozpoczęciem rozgrywki.

Gra rozpoczyna się od wyświetlenia pola 3x3. Program prosi o podanie współrzędnych pola przez użytkownika w formacie "cyfra cyfra". Za tą czynność odpowiada metoda GetHumanMove(), która jednocześnie sprawdza czy podane dane nie wykraczają poza zakrez pola oraz czy podane pole nie jest już zajęte. Metoda ifWin() sprawdza czy 3 sąsiednie pola nie są zajęte przez jednego gracza. W takim przypadku funkcja gameOver() zwraca odpowiedniego zwycięzcę a metoda score() przyznaje odpowiednią puntktację właściwemu graczowi. Natomiast jeśli takiego nie ma zwraca informację o pustym polu, dzięki czemu rozgrywka trwa dalej. Sama rozgrywka wykonuje się w funkcji play().

#### 3 Budowa algorytmu

Jeśli chodzi o realizację algorytmu "minimax" to jest on podzielony na trzy części.

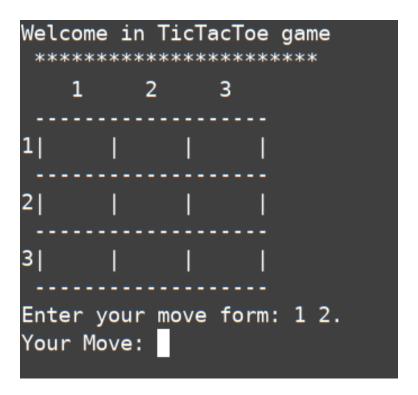
- Metoda min pomaga w znalezieniu kolejnego ruchu dla komputera.
- metoda max pomaga w znalezieniu kolejnego ruchu dla użytkownika
- Metoda minimax() zwraca współrzędne najlepszego ruchu dla komputera CPU

Algorytm dąży do maksymalizacji zysków, czyli znalezienia najlepszego pola dla "ataku", jednocześnie starając się zminmalizować straty czyli "obrony" przez zakończeniem rozgrywki. Ze względu na prosta budowę zaimplementowanego algorytmu program wykonuje się dla maksymalnej głębokości rekurencji, tj. ilości pól.

#### 4 Wnioski

Program działa prawidłowo. Często można odczuć powtarzalnośc ruchów ze strony komputera.

## 5 Zrzuty ekranów podczas rozgrywki



You	r Mo 1	ve	2		3			
1	Х	ı		I		ı		
2		ı		I		Ī		
3				I		١		
Comp	oute 1	r N	1ove 2		3			
1	Χ	l		l		I		
			^			1		
2		l	0			<u> </u>		

```
Computer Move:
   1 2 3
1| X |
2 0 0 0
3| X |
Enter your move form: 1 2.
Your Move: 2 3
   1
       2
1| X |
  0 | 0 | X
3| X |
Computer Move:
       2
            3
1 | X |
2| 0 | 0 | X
3| X | | 0 |
Enter your move form: 1 2.
Your Move:
```

```
Your Move: 1 3
   1 2 3
1| X | | X |
2| 0 | 0 | X
3| X |
Computer Move:
  1 2 3
1| X | 0 | X |
2| 0 | 0 | X
3| X | | 0 |
Enter your move form: 1 2.
Your Move: 3 2
  1 2 3
1| X | 0 | X
2| 0 | 0 | X
3| X | X | 0 |
It's a draw
```