

WSI - ćwiczenie 2.

Dwuosobowe gry deterministyczne

26 października 2025

1 Sprawy organizacyjne

1. Ćwiczenie realizowane jest samodzielnie.
2. Ćwiczenie wykonywane jest w języku Python.
3. Ćwiczenie powinno zostać oddane najpóźniej do 13.11.2025. W ramach oddawania ćwiczenia należy zademonstrować prowadzącemu działanie kodu oraz utworzyć pull request (z kodem oraz raportem) który prowadzący będzie mógł komentować.
4. Kod oraz raport powinien być umieszczony na repozytorium `gitlab-stud.elka.pw.edu.pl`, a pull request utworzony do konta prowadzącego: `@djagodzi` (dla nazwisk A-N) i `@pzawist2` (dla nazwisk O-Z)
5. Raport powinien być w postaci pliku `.pdf`, `.html` albo być częścią notebooka jupyterowego. Powinien zawierać opis eksperymentów, uzyskane wyniki wraz z komentarzem oraz wnioski.
6. Na ocenę wpływa poprawność oraz jakość kodu i raportu.
7. Można korzystać z pakietów do obliczeń numerycznych, takich jak *numpy*
8. Implementacja algorytmu powinna być ogólna. W szczególności powinna być możliwa do zastosowania dla dowolnej deterministycznej gry dwuosobowej.
9. Ćwiczenie **nie wymaga** używania interfejsu *Solver*.

2 Ćwiczenie

Celem ćwiczenia jest implementacja algorytmu minimax z obcinaniem $\alpha - \beta$. Dla różnych ruchów o tej samej jakości, algorytm powinien zwracać losowy z nich z równym prawdopodobieństwem.

Następnie należy wykorzystać implementację do porównania jakości dla różnych głębokości przeszukiwania (tj. sprawdzenia, jak komputerowy gracz grający z głębokością N poradzi sobie z komputerowym graczem grającym z głębokością M dla różnych wartości N i M) dla gry Szewc (ang. Dots and Boxes, https://en.wikipedia.org/wiki/Dots_and_boxes).

Można skorzystać z implementacji gry dostępnej w repozytorium <https://github.com/lychan1/two-player-games>.

Uwaga - w grze Szewc jeden gracz może w niektórych sytuacjach wykonać kilka ruchów pod rząd.

2.1 Dla chętnych

1: Dodaj do algorytmu możliwość sprawdzania maksymalnie pewnej liczby ruchów w każdym stanie, wybieranych losowo spośród wszystkich dostępnych.

Porównaj ten wariant z bazowym algorytmem pod względem uzyskiwanych wyników oraz potrzebnego czasu.

2: Przygotuj program pozwalający użytkownikowi grać z komputerem z możliwością wyboru poziomu trudności.