Projektowanie algorytmów i metody sztucznej inteligencji
Prowadzący: Mgr inż. Marcin Ochman
Wt. 15:15-16:55

Projekt 4

Autor:

Sieracka Weronika

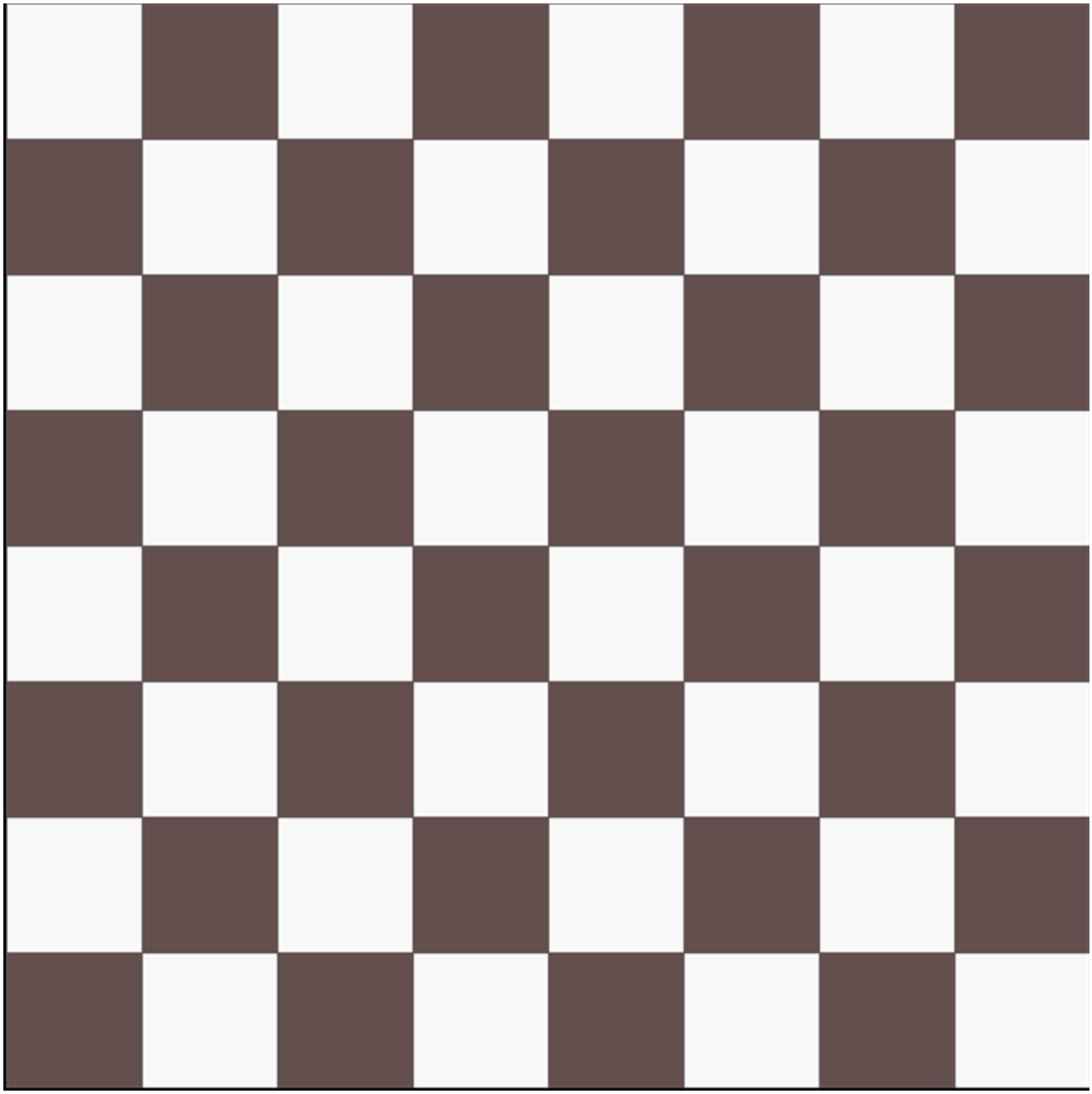
5 czerwca 2020

1 Wstęp:

Zadania polegało na stworzeniu gry, w naszym przypadku warcabów, która wykorzystuje sztuczną inteligencje. Graficzna część gry została stworzona dzięki wieloplatformowej bibliotece programistycznej zwanej Simple and Fast Multimedia Library, potocznie SFML.

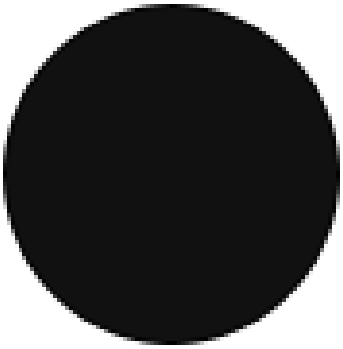
2 Warcaby

Wszystkie poniższe obrazy zawarte w sprawozdaniu zostały użyte do tworzenia gry.
Gra warcaby klasyczne rozgrywana jest na planszy o rozmiarze 8 x 8 pól pokolorowanych na przemian na kolor jasny i ciemny.

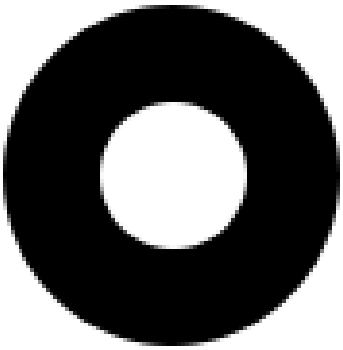


Posiada ona 4 rodzaje pionków:

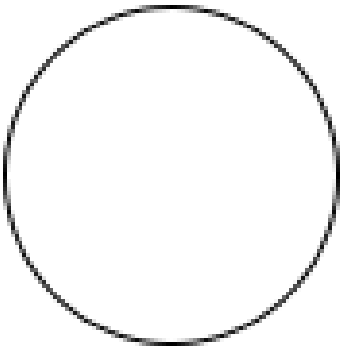
·Czarne



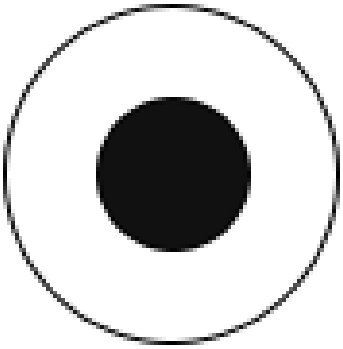
·Czarna Damka



·Białe



·Białe Damki



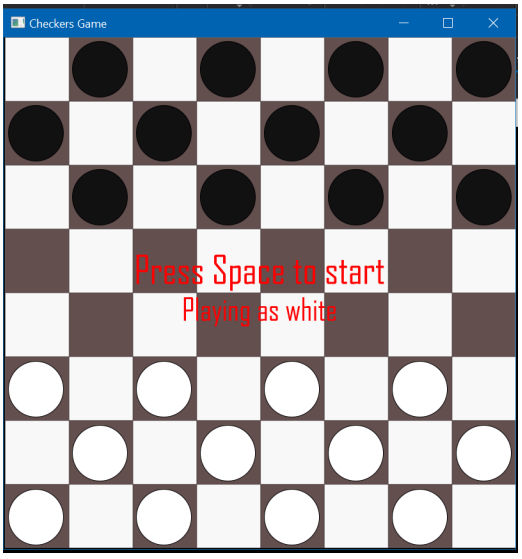
Plansza jest zaimplementowana za pomocą tablicy, która zawiera odpowiednia ilość pól oraz informację o tym co znajduję się na danym polu planszy. Klasa „Board" odpowiadająca planszy zawiera metody wykorzystywane do przesuwania się pionków i sposobów bicia. Do wykonania ruchu została użyta klasa zawierająca współrzędne pionka przed i po ruchu. Pionek może wykonać ruch jeśli znajduje się on na liście możliwych ruchów dla niego i jeśli na tej liście znajduję się bicie to musi ono zostać wykonane i nie ma możliwości użycia innego pionka w takiej sytuacji.

2.1 Zasady

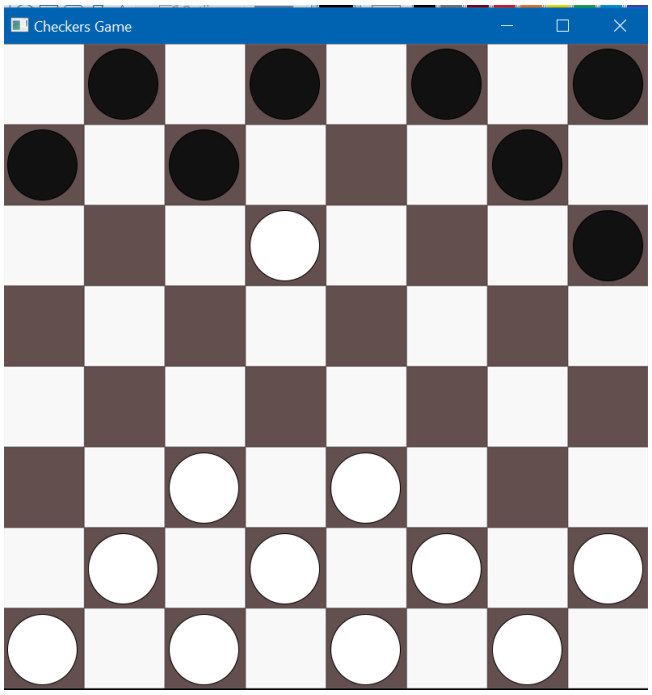
Gra kończy się, gdy jeden z graczy straci wszystkie swoje pionki lub nie będzie mógł wykonać ruchu, ponieważ będzie zablokowany. Gracze zremisuje jeśli obydwoje wykonają po 10 ruchów damkami bez żadnego zbicia. Jeśli pionek dojdzie do ostatniego rzędu (czyli najdalej oddalonego od niego) pionek staje się damką. Przyjęto zasadę bicia obowiązkowego. Pionki mogą poruszać się tylko do przodu, ale mogą bić do tyłu. Damka po zbiciu pionka może dalej się poruszać. Możliwe jest bicie kilkukrotne.

3 Wersja graficzna gry

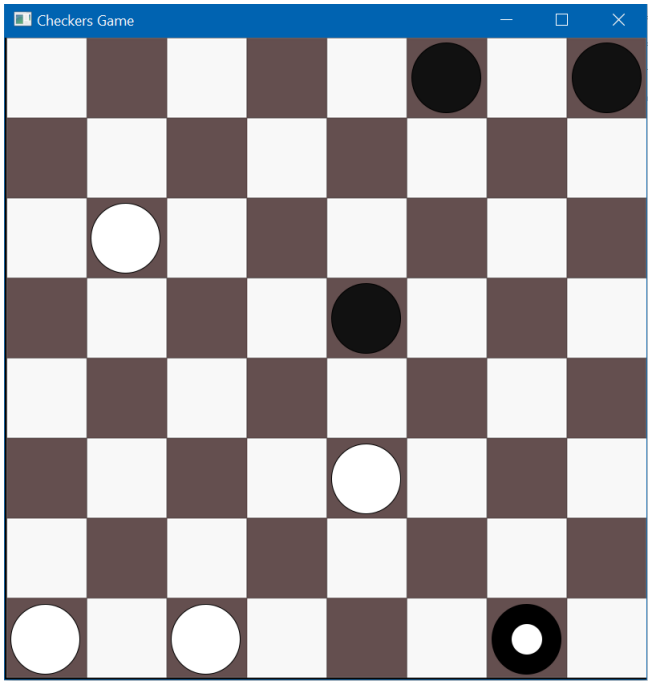
3.1 Początek gry



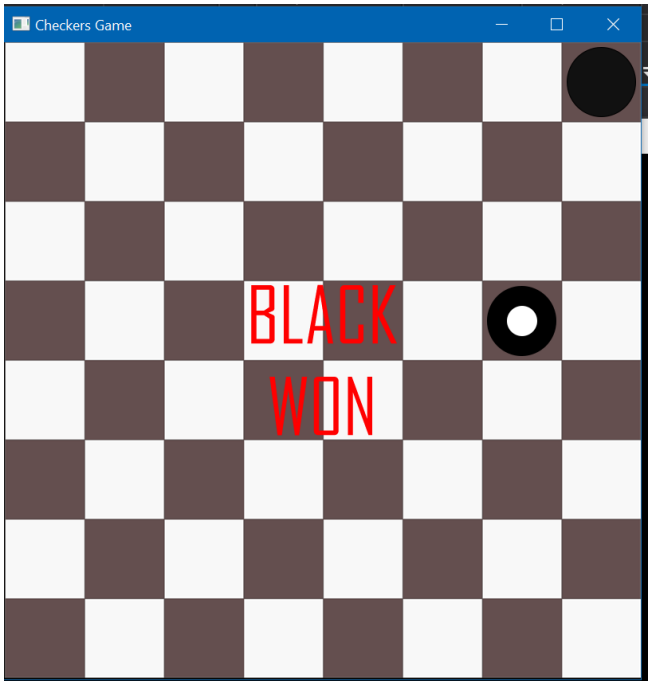
3.2 Podczas gry



3.3 Gra z damka



3.4 Koniec gry



4 Opis wykorzystania AI

Do implementacji AI wykorzystano algorytm Minimax. Polega on na wybieraniu ruchów, które polegają na minimalizowaniu strat oraz osiągnięciu największego zysku dla danej strony. Polega to na stworzeniu drzewa wszystkich możliwych sytuacji dla następnych X ruchów (które określane są jako głębokość). Każdej możliwej przewidzianej sytuacji przypisywane są odpowiednie wartości zysku lub straty w zależności od wyniku ruchów. Następnie algorytm wybiera sytuację z największym zyskiem i realizuje ruch który prowadzi do takowej sytuacji w grze. Algorytm polega na założeniu, że strona (gracz) dla której wywoływany jest algorytm stara się zdobyć największą ilość punktów, a drugi gracz stara się przeszkodzić w tym. Drzewo w każdym momencie dostaje informację czy aktualny gracz stara się minimalizować czy maksymalizować zyski i w zależności od tego wybiera odpowiednią sytuację. Zysk jest określany na podstawie punktów, które są przydzielane za określone ruchy i doprowadzenie do konkretnej sytuacji:
1. Wygrana +1000
2. Przegrana -1000
3. Remis 0
4. Pionek na bocznym polu +2
5. Pionek oddalony o jedno pola bocznego +1
6. Każdy pionek +10
7. Każda damka +30
8. Pionek z możliwością bicia +15

Do algorytmu Minimax wykorzystany został dodatkowo algorytm Alfa-Beta. Modyfikuje on algorytm Minimax poprzez wprowadzenie dodatkowych zmiennych oznaczających minimalną i maksymalnąwartość jaką algorytm Minimax może zapewnić na danej głębokości drzewa lub wyżej. Alfa oznacza najlepszą wartość jaką strona maksymalizująca może uzyskać a Beta najlepszą wartość jaką strona minimalizująca może uzyskać. Algorytm Alfa-Beta polega na przerwaniu oceniania danej sytuacji, gdy zostanie znaleziona co najmniej jedna inna możliwość, która dowodzi że ruch jest gorszy niż poprzednio badany. To oznacza, że taki ruch nie musi być dalej analizowany i można go porzucić. Minimax razem z dodatkowym algorytmem zwraca ten sam

wynik, który byłby zwrócony w standardowej wersji, ale jest szybszy, ponieważ przycina gałęzie, które nie mogą wpłynąć na ostateczną decyzje.

5 Złożoność obliczeniowa

W warcabach liczba mozliwych ruchów jest zmienna, wiec nie da sie dokładnie okreslic złozonosci obliczeniowej algorytmu. Jesli załozymy, że dla kazdego ruchu mozna wykonac n-nastepnych, to złozonosc obliczeniowa algorytmu minimax wyniesie O(nm), gdzie m to maksymalna głebokosc rekurencji.Natomiast złozonosc obliczeniowa algorytmu Alfa-Beta wynosi $O(n_{m/2})$

6 Wnioski

• Złożoność algorytmu jest dość duża, przy większej głębokości ruchy mogą wykonywać się bardzo powoli, więc dla nieuciążliwej rozgrywki trzeba zmniejszyć dopuszczalną maksymalną głębokość algorytmu w zależności od możliwości obliczeniowych komputera
• Algorytm nie jest niepokonany, ponieważ zależy od wartości przypisywanych danym ruchom, a te nie zawsze są idealne dla danej sytuacji na planszy

7 Literatura

https://pl.wikipedia.org/wiki/Algorytmmin-max
https://en.wikipedia.org/wiki/Minimax
https://kcir.pwr.edu.pl/ witold/aiarr/2009projekty/warcaby/
https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/
https://en.wikipedia.org/wiki/Alpha%E2%80%93betapruning
https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-4-alpha-beta-pruning/