

Gra - sprawozdanie

Nazwisko, Imię, Indeks	Kajetan Zdanowicz, 248933
Prowadzący kurs	Mgr inż. Marcin Ochman
Termin zajęć	WT 15:15
Data oddania sprawozdania	05.06.2020r.

1 Wstęp

Celem ćwiczenia było zaimplementowanie gry umożliwiającej rywalizację ze sztuczną inteligencją. Nie udało się dokończyć zadania i program jest sprowadzony do wersji prototypowej, posiadającej wszystkie funkcjonalności szachów, poza funkcją sprawdzającą czy król jest szachowany.

2 Zaimplementowane techniki sztucznej inteligencji

Program z założenia miał zawierać algorytm minimax, odpowiadający za działanie sztucznej inteligencji. Algorytm nie został w pełni napisany, ponieważ drzewo nie może zostać zbudowane, ze względu na popełnione wcześniej błędy w implementacji. Program jednak posiada fundamenty umożliwiające jego rozbudowę do gry z komputerem:

- Funkcja oceniająca planszę - algorytm miał na celu minimalizację zbitych własnych figur przy maksymalizacji zbitych figur przeciwnika. Każna figura ma przydzielone punkty, które przysługują przeciwnikowi po zbiciu
 - Goniec - 60
 - Król - 1800
 - Koń - 60
 - Królowa - 180
 - Pion - 20
 - Wieża - 100
- Drzewo możliwych plansz zadaną głębokością rekurencji wywoływania konstruktora, które nie może zostać zbudowane w grze, przez wcześniejsze błędy
- Funkcja zwracająca możliwe ruchy dla każdej ze stron

3 Planowane ulepszenia programu

- Dodanie algorytmu Minimax

Na podstawie poprawnie zbudowanego drzewa wszystkich możliwych ruchów, zostanie zaimplementowany powyższy algorytm. Rozpocznie się on od oceny każdej z wygenerowanych plansz od dołu drzewa, w zależności od gracza wykonującego ruch. Na każdym piętrze na przemian będzie wybierana największa i największa wartość oceny odpowiadająca najlepszym ruchom komputera i najgorszym dla komputera ruchom przeciwnika. Algorytm będzie zwracał najlepszy możliwy do wykonania ruch przez komputer w danym momencie - zakładając inteligencję przeciwnika.

- Dodanie cięć alfa-beta

Przy przeliczaniu wartości poszczególnych węzłów na danym piętrze, dochodzi do sytuacji, kiedy komputer niepotrzebnie jest zmuszony zliczać węzły przynależące pod węzeł nie mający wpływu na rozgrywkę. Utrzymywane są dwie zmienne: alfa oraz beta. Pierwsza przechowuje minimalny wynik gracza maksymalizującego, a druga maksymalny wynik gracza minimalizującego. Początkowo alfa jest -nieskończonością, a beta +nieskończonością. W miarę postępowania rekurencji, zmniejsza się przedział (alfa; beta). W sytuacji, kiedy wartość alfa przekracza wartość beta, nie ma potrzeby przeszukiwania głębszej, ponieważ dalsze rozwiązania nie będą pożądane przez żadnego z graczy. Cięcia alfa-beta około dwukrotnie zmniejszają współczynnik rozgałęzienia drzewa.

- Dopeczętowanie głębokości rekurencji drzewa

Przy założeniu stałej ilości możliwych do wykonania ruchów w danej turze (a), możliwa jest do oszacowania liczba wygenerowanych plansz. Zakładając przykładową głębokość rekurencji (d), uzyskujemy a^d stworzonych węzłów drzewa. Można zauważyć bardzo szybki przyrost obliczeń, które musi wykonać komputer. Ustalenie odpowiedniej głębokości rekurencji zoptymalizuje poziom komputera w stosunku do czasu jego myślenia. Szacowana głębokość rekurencji wynosi 5-6.

- Urozmaicenie funkcji oceniającej planszę

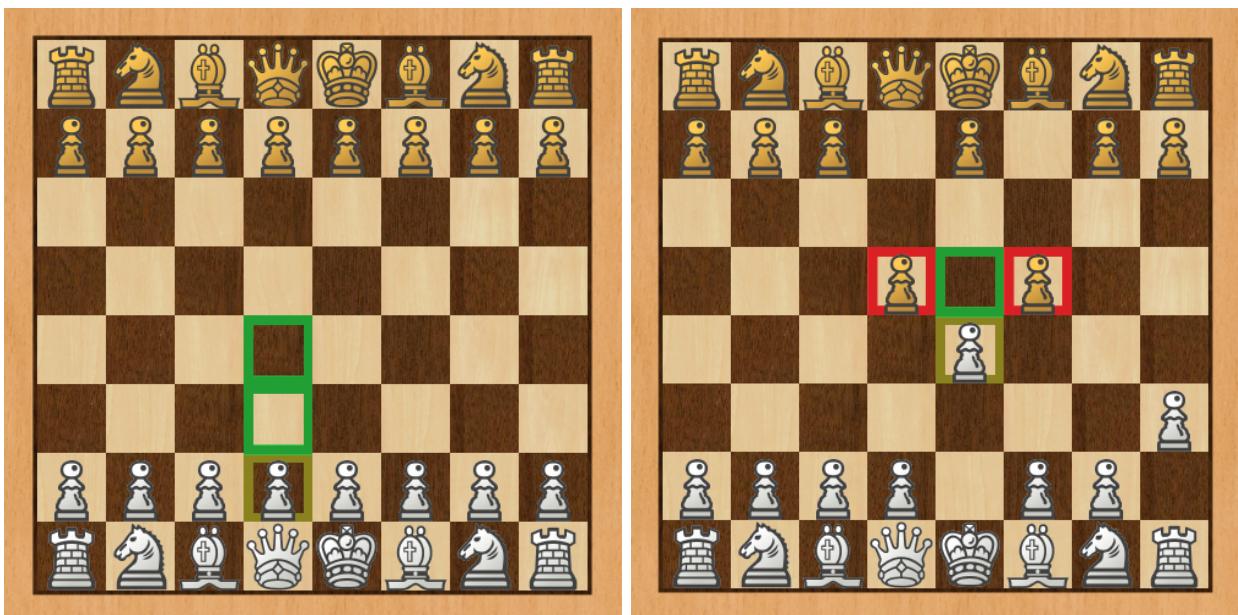
Dodana zostanie tablica dwuwymiarowa - dla każdego typu figury, premiująca pola na planszy inną liczbą punktów, zmuszając figury do objęcia swoim zasięgiem większej ilości pól, premiowanie zbliżania się pionów do promocji. Oprócz tego zostanie dodana premia za roszady i promocje piona.

- Funkcja sprawdzająca, czy jest szach

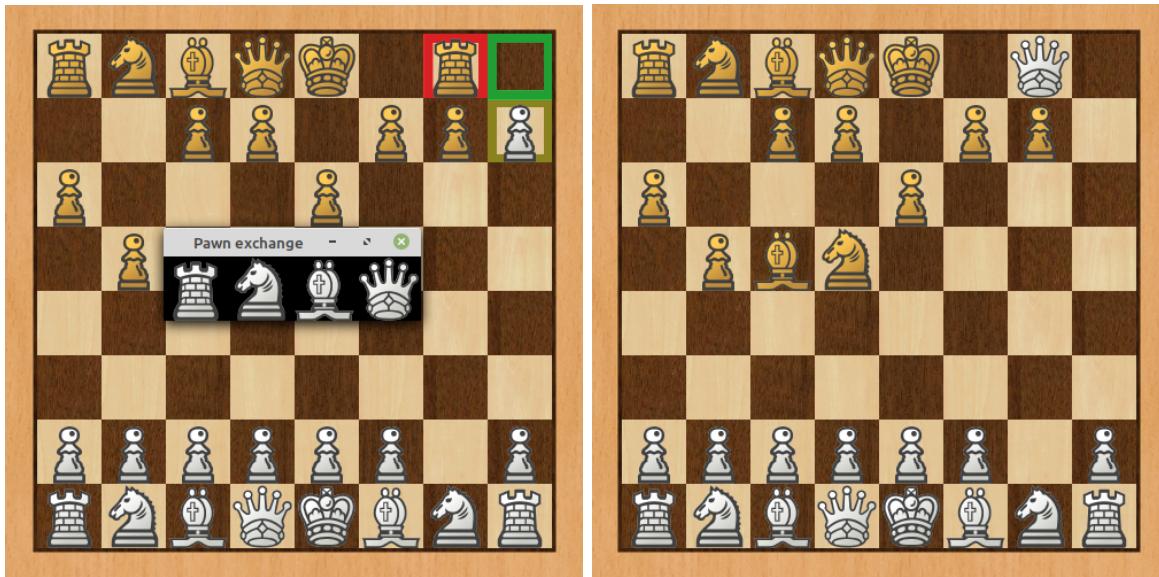
- Reguły precyzuujące koniec gry

4 Zaimplementowane ruchy figur oraz GUI

4.1 Pion



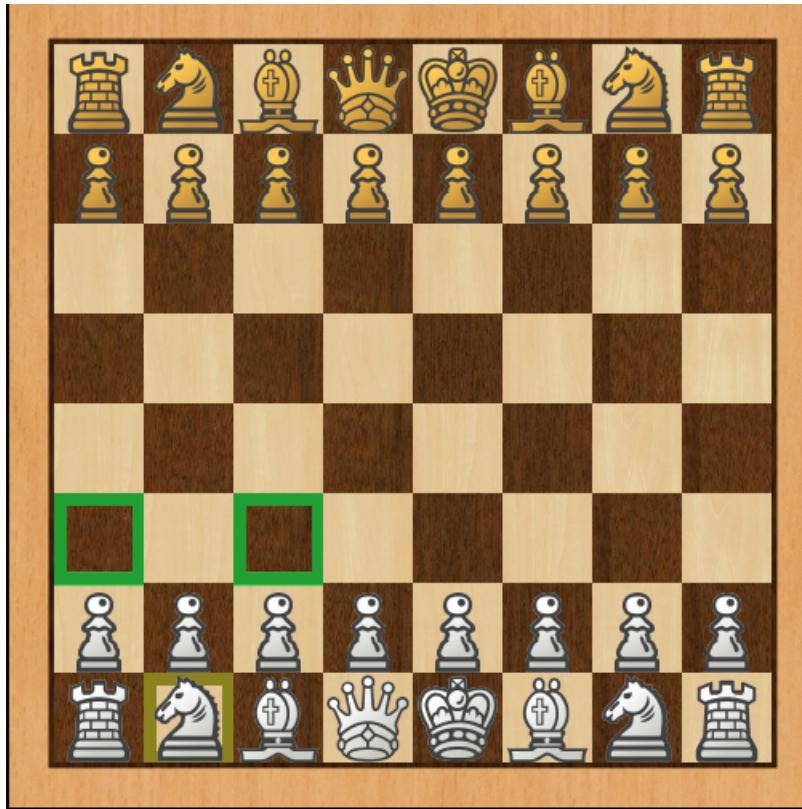
Po dojściu pionkiem na ostatnie pole, możliwa jest promocja. Wybrana figura z wyświetlnego okna zamieni pionka na miejsce. W przypadku wyłączenia okna lub innego nieprzewidzianego zachowania, pionek domyślnie jest zamieniany na królową.



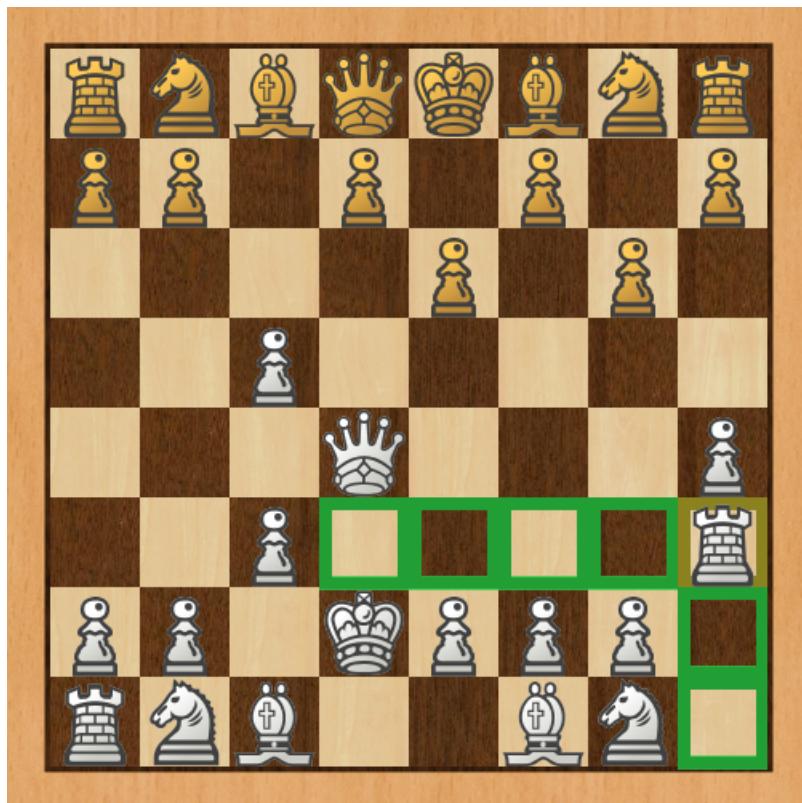
4.2 Goniec



4.3 Koń



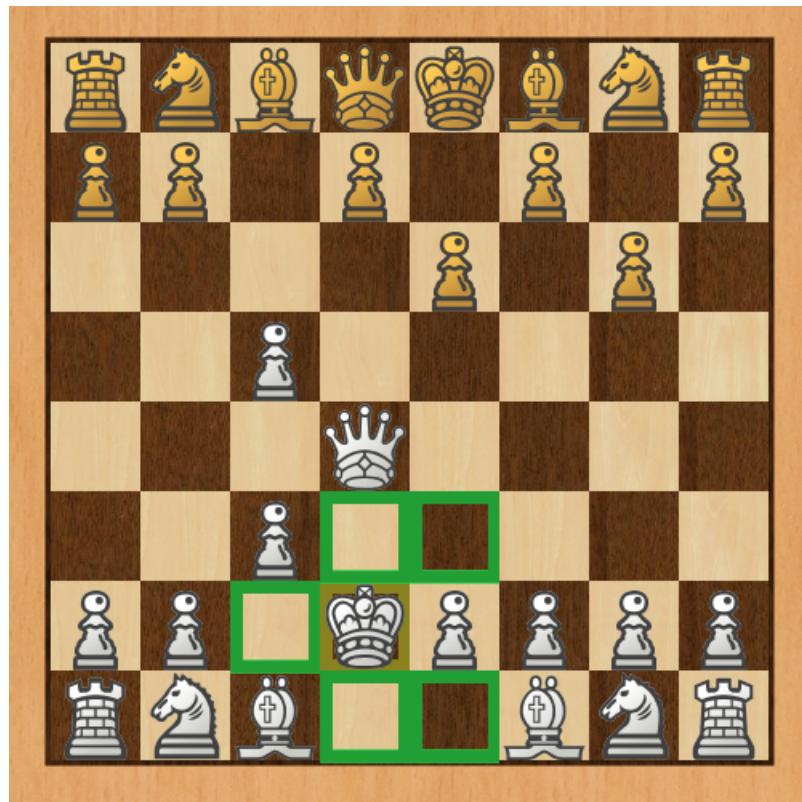
4.4 Wieża



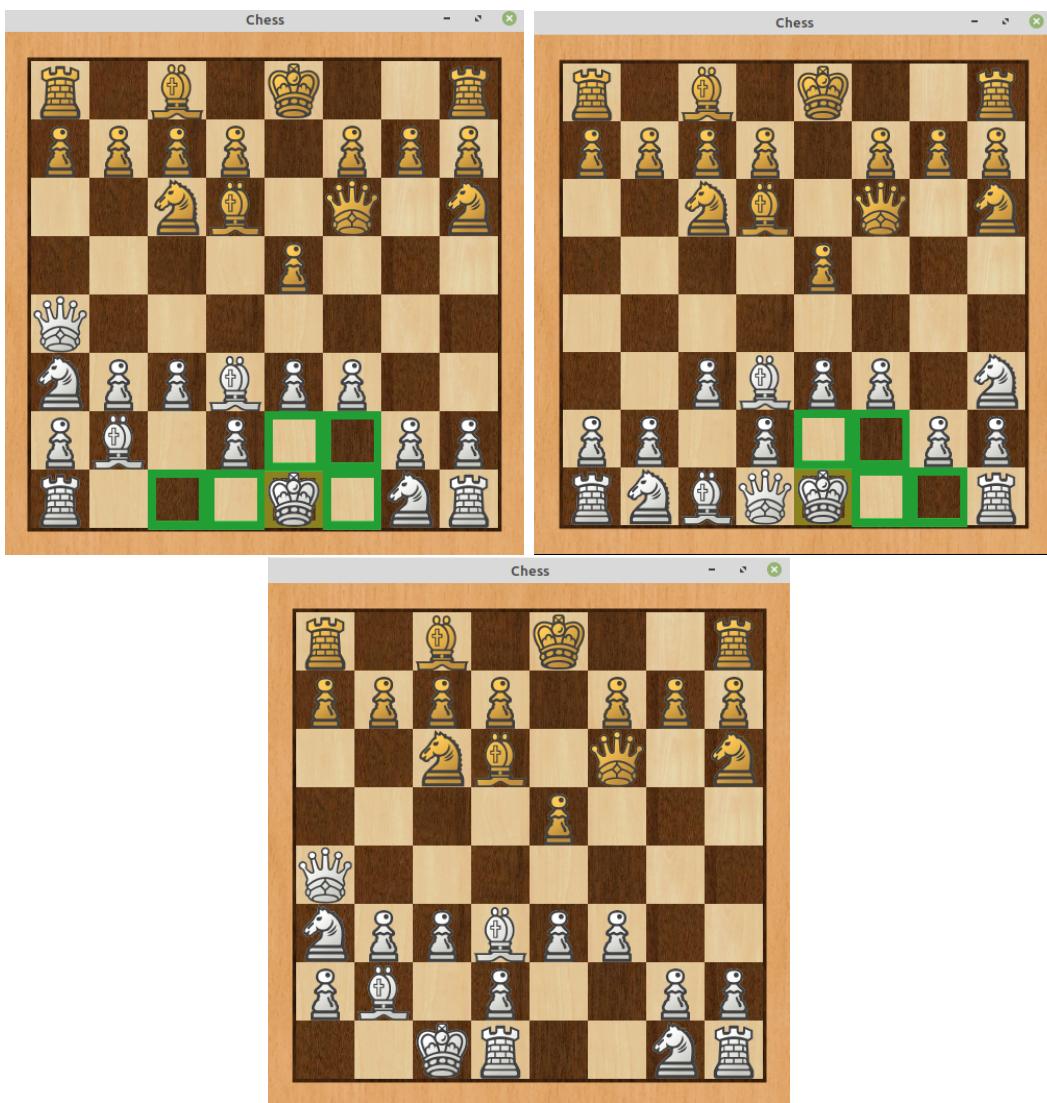
4.5 Królowa



4.6 Król



Została zaimplementowana również roszada królewska. Jest ona możliwa do wykonania, gdy król oraz wieża nie były jeszcze używane w grze oraz pola pomiędzy wieżą i królem są puste.



5 Wnioski

- Program będzie użyteczny po udoskonaleniu błędów. Przy poprawie należy zwrócić szczególną uwagę na operacje związane z pamięcią dynamiczną
- Przed implementacją sztucznej inteligencji należy upewnić się o możliwości wykonania przez program wszystkich prawidłowych ruchów
- Program w prototypowej wersji umożliwia wzajemne zbijanie figur

6 Bibliografia

- <https://www.sfml-dev.org/>
- <https://www.geeksforgeeks.org/design-a-chess-game/>
- https://pl.wikipedia.org/wiki/Algorytm_alfa-beta