

SPRAWOZDANIE		
Michał Filipowicz	226393	PAMSI
Data oddania: 14.06.2017	Kółko i krzyżyk – min-max	dr inż. Łukasz Jeleń

I. Wstęp

Zadaniem do wykonania była implementacja algorytmu min-max na wybranej z listy możliwych (do samodzielnej implementacji) gier.

Mój wybór padł na kółko i krzyżyk, w którym jak można się domyślić algorytm ma za zadanie wybranie optymalnego w danej sytuacji na planszy ruchu.

Sama gra została napisana z użyciem biblioteki graficznej SFML. Łatwa w obsłudze biblioteka pozwoliła na szybką implementację przyzwoicie wyglądającego GUI gry, a następnie skupienie na implementacji samych jej zasad.

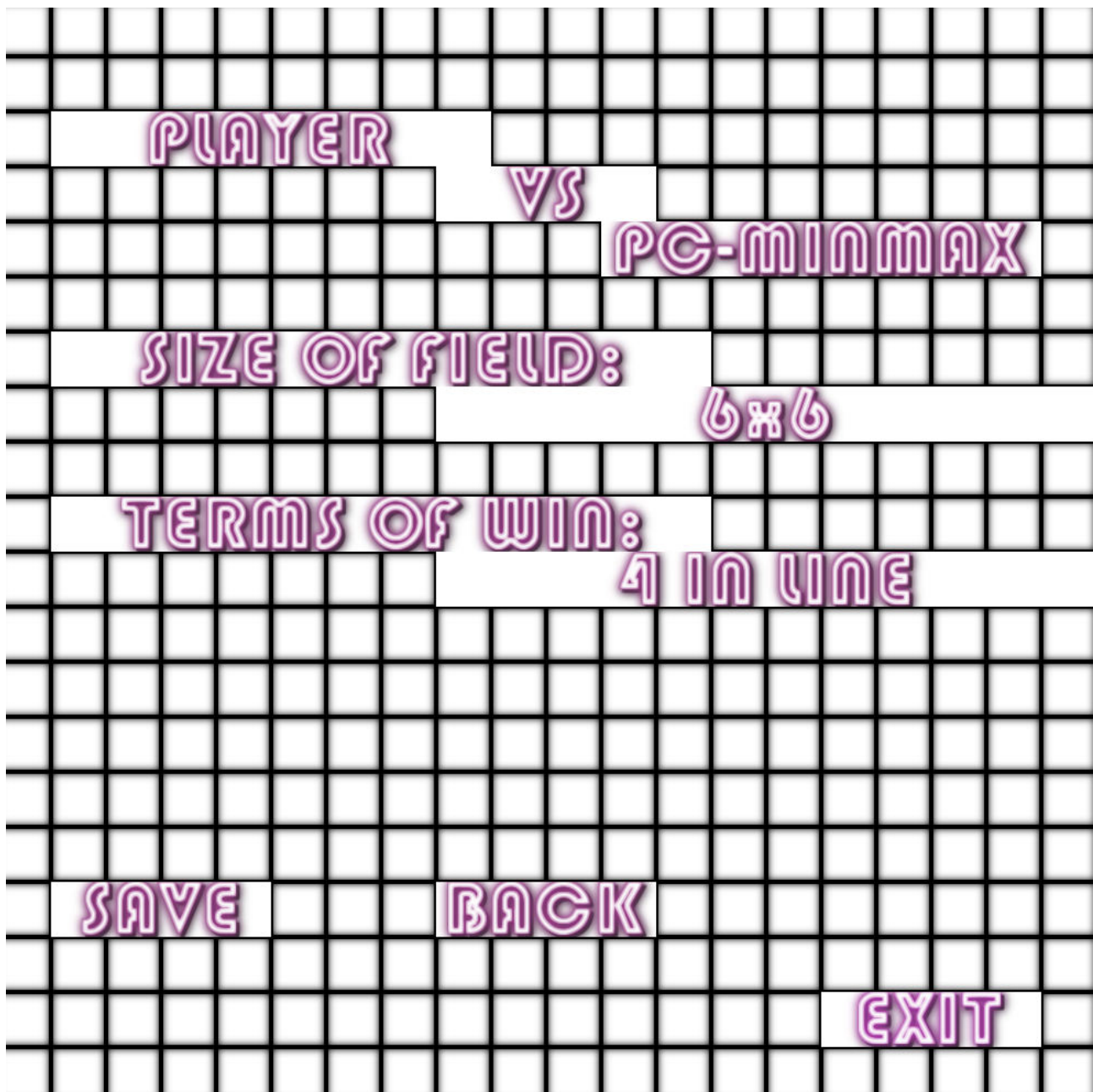
W stworzonej przeze mnie wersji „kółka i krzyżyk” mamy do wyboru rozmiar pola na którym toczy się rozgrywka:

a) 3x3 b) 4x4 c) 5x5 d) 6x6

rozszerzenie tych możliwości o większe nie byłoby problemem dla samej gry, problemem jest jednak napisane do niej GUI, w którym nie przewidziałem wielkości okna potrzebnego do komfortowej gry w kółko i krzyżyk. Dlatego predefiniowana wielkość okna 1000x1000 pozwala na wyraźną rozgrywkę z maksymalnym rozmiarem pola 6x6.

Ponadto istnieje możliwość wyboru zasad panujących podczas rozgrywki. Rozpoczynając od „3 w rzędzie”, mamy możliwość wyboru ile ustawionych obok siebie symboli ma powodować zwycięstwo (6 to maksimum, ze względu na maksymalny rozmiar planszy). Oczywiście dla planszy o wielkości 3x3 nie można wybrać warunku zwycięstwa „4 w rzędzie”, ponieważ byłoby to fizycznie niemożliwe.

Ustawienia można wybrać w zakładce „settings” w samej grze:



ustawienia zostały rozszerzone o możliwość ich zapamiętania. Opcja zapisu polega na prostym zapisaniu ustawień w pliku „config.ini”, z którego gra podczas uruchomienia pobiera informacje o ustawieniach. Jeśli podczas uruchomienia gry config nie znajduje się w folderze z grą, jest on automatycznie tworzony z domyślnymi ustawieniami.

Jak na powyżej zamieszczonym screenie również widać, prócz wymienionych wcześniej ustawień istnieje również możliwość wybrania graczy. Dzięki temu możemy grać zarówno kontra drugiej osobie, algorytmowi, jak i obserwować wyniki rozgrywki algorytm kontra algorytm.

II. Algorytm

Zastosowany algorytm min-max rekurencyjnie sprawdza który ruch jest w danej sytuacji najlepszy (na podstawie funkcji zwracającej wagę danego posunięcia).

Sytuacja komplikuje się już od powiększenia planszy do rozmiaru 4x4. Możliwości pojawia się wtedy zbyt wiele, dlatego należało użyć kontroli głębokości, którą metodą prób i błędów postarałem dobrać się optymalnie dla każdego przypadku wybranych ustawień.

III. Wnioski

Algorytm min-max okazał się dobrą metodą implementacji SI w klasycznym „kółku i krzyżyk”, jednak jak wszystko ma on swoje wady.

Głównym problemem algorytmu jest wspomniana już wcześniej złożoność obliczeniowa wynikająca z rekurencyjnej natury zastosowanej w moim przypadku implementacji.

Problem ten został rozwiązany poprzez kontrolę głębokości, która zapewnia poprawne działanie algorytmu w rozsądnym czasie, jednocześnie jednak nie gwarantując wybrania przez algorytm zawsze najlepszego ruchu.

Teoretycznie głębokość mogłaby zostać bez kontroli, co zapewniłoby przewidzenie przez algorytm zawsze najlepszego z możliwych ruchów, jednak sprzęt którym dysponuję i czas potrzebny na wykonanie pojedynczego ruchu przez algorytm fizycznie uniemożliwiają przeprowadzenie takich testów.

Źródła:

https://pl.wikipedia.org/wiki/Algorytm_min-max
<http://neverstopbuilding.com/minimax>