Projekt 3

Table 1: Informacje o projekcie i uczestnikach

Kierunek	Informatyczne systemy automatyki
Przedmiot	Projektowanie i analiza algorytmów
Data oddania	2024-06-10
Imię i Nazwisko	Numer albumu
Kateryna Fedoruk	272609

1 Opis Tworzonej Gry

Gra opisana w podanym kodzie to komputerowa, zmieniona wersja warcabów rosyjskich. Warcaby rosyjskie, podobnie jak inne odmiany warcabów, są grą planszową dla dwóch graczy, rozgrywaną na planszy o rozmiarze 8×8, gdzie gracze na przemian wykonują ruchy swoimi pionkami. Celem gry jest zbicie wszystkich pionków przeciwnika lub zablokowanie ich, tak aby przeciwnik nie mógł wykonać ruchu.

1.1 Zasady Gry

- Gra odbywa się na planszy o rozmiarze 8×8 pól.
- Każdy gracz rozpoczyna grę z 12 pionkami rozmieszczonymi na ciemnych polach planszy, na pierwszych trzech rzędach najbliżej siebie. Grę rozpoczynają białe.
- Gracze wykonują ruchy na przemian, przesuwając pionki o jedno pole do przodu na ukos na puste pola. Pionki mogą również bić pionki przeciwnika, przeskakując nad nimi na puste pole za zbitym pionkiem. Zwykłe pionki mogą bić zarówno do przodu, jak i do tyłu.
- Gdy pionek dojdzie do ostatniego rzędu planszy, staje się damką. Damka może poruszać się po ukosie o dowolną liczbę pól w obu kierunkach. Damki mogą bić zarówno do przodu, jak i do tyłu.
- W tej wersji warcabów rosyjskich zbijanie nie jest obowiązkowe. W trakcie jednego ruchu można wykonać tylko jedno bicie (dotyczy to i pionków i damek).
- Gra kończy się, gdy jeden z graczy zbije wszystkie pionki przeciwnika lub zablokuje przeciwnika, tak że ten nie może wykonać żadnego ruchu. Do remisu dochodzi, gdy żaden z graczy nie ma szans na wygranie rundy.

1.2 Techniki Sztucznej Inteligencji

W grze zaimplementowano prostą sztuczną inteligencję (SI) korzystającą z algorytmu *minimax* z optymalizacją alfa-beta. Algorytm ten jest powszechnie używany w grach strategicznych do przewidywania najlepszych ruchów. Oto szczegółowy opis technik SI zastosowanych w grze:

1.2.1 Algorytm Minimax

Algorytm *minimax* jest wykorzystywany do symulacji różnych możliwych ruchów, oceny ich wartości i wyboru najlepszego możliwego ruchu dla gracza. Algorytm działa rekurencyjnie, na przemian maksymalizując i minimalizując wartość heurystyczną planszy dla gracza maksymalizującego i minimalizującego.

Rekursja Rekursja to technika, w której funkcja wywołuje samą siebie w celu rozwiązywania problemów złożonych na mniejsze, łatwiejsze do rozwiązania problemy. W algorytmie *minimax*, rekursja jest używana do przeszukiwania drzewa możliwych ruchów do określonej głębokości. W moim przypadku, głębokość rekursji została ustawiona na 3. Oznacza to, że algorytm przeszukuje trzy poziomy możliwych ruchów do przodu. Wybór głębokości 3 jest kompromisem między dokładnością oceny a czasem potrzebnym na przetworzenie wszystkich możliwych ruchów. Wpływa to też na poziom trudności gry(więcej w seksji Podsumowanie i Wnioski). Głębsze przeszukiwanie (większa głębokość) pozwala na dokładniejszą ocenę, ale zwiększa znacząco czas obliczeń.

Funkcja Oceny Heurystycznej Funkcja oceny heurystycznej ocenia wartość danego stanu planszy. W mojej grze, wartość ta jest oparta na liczbie i rodzaju pionków obu graczy. Pionki zwykłe mają wartość 3, a damki mają wartość 7. Funkcja ta dodaje wartości pionków gracza białego i odejmuje wartości pionków gracza czarnego. W ten sposób, pozytywny wynik oznacza przewagę gracza białego, a negatywny wynik oznacza przewagę gracza czarnego.

Alfa-Beta Pruning Alfa-beta pruning to technika optymalizacji algorytmu *minimax*, która pozwala na eliminowanie części drzewa wyszukiwania, które nie mają wpływu na ostateczną decyzję. Zmienne `alfa` i `beta` reprezentują najgorsze możliwe wyniki dla graczy maksymalizującego i minimalizującego, odpowiednio.

- **Alfa**: Najlepsza (największa) wartość, jaką gracz maksymalizujący (w naszym przypadku czarny) może zagwarantować sobie na danym poziomie drzewa wyszukiwania. Początkowo jest ustawiona na najniższą możliwą wartość (minus nieskończoność).
- **Beta**: Najlepsza (najmniejsza) wartość, jaką gracz minimalizujący (w naszym przypadku biały) może zagwarantować sobie na danym poziomie drzewa wyszukiwania. Początkowo jest ustawiona na najwyższą możliwą wartość (plus nieskończoność).

Jeśli w trakcie przeszukiwania drzewa algorytm znajdzie ruch, który ma wartość mniejszą niż `alfa` dla gracza maksymalizującego lub większą niż `beta` dla gracza minimalizującego, dalsze przeszukiwanie tej gałęzi jest zbędne. Oznacza to, że ten ruch nie będzie lepszy od wcześniej znalezionych i można go pominąć, co znacznie przyspiesza proces decyzyjny.

Gracz Maksymalizujący i Minimalizujący W algorytmie *minimax* mamy dwa typy graczy:

- **Gracz maksymalizujący**: Stara się maksymalizować wartość oceny heurystycznej planszy. W mojej grze, graczem maksymalizującym jest komputer (gra pionkami czarnymi), który chce uzyskać jak najwyższą wartość.
- **Gracz minimalizujący**: Stara się minimalizować wartość oceny heurystycznej planszy. W mojej grze, graczem minimalizującym jest użytkownik (gra pionkami białymi), który chce uzyskać jak najniższą wartość dla przeciwnika.

1.2.2 Znajdowanie Najlepszego Ruchu

Aby znaleźć najlepszy ruch, algorytm przeszukuje wszystkie możliwe ruchy gracza czarnego (komputera), symuluje każdy z nich i ocenia je za pomocą algorytmu *minimax* do głębokości 3. Wybiera ruch, który daje najniższą wartość oceny heurystycznej, ponieważ komputer gra jako gracz minimalizujący.Gdy algorytm znajdzie kilka ruchów o identycznej wartości oceny, wybierze ten, który pojawia się jako pierwszy w kolejności generowanych ruchów. W ten sposób komputer wybiera ruch, który jest najmniej korzystny dla przeciwnika (gracza białego).

2 Podsumowanie i Wnioski

Gra w warcaby rosyjskie, jak opisano powyżej, jest interesującym przykładem zastosowania technik sztucznej inteligencji w grach komputerowych. Dzięki zastosowaniu algorytmu minimax z optymalizacją alfa-beta, SI jest w stanie podejmować decyzje w oparciu o przewidywanie przyszłych ruchów, co czyni ją bardziej skuteczną w grze przeciwko ludzkiemu graczowi.

Postanowiłam zaimplementować wersję warcabów, w której bicie nie jest obowiązkowe w implementacji, co czasami prowadziło do wyboru przez algorytm ruchów innych niż bicie. W wielu wersjach warcabów bicie jest obowiązkowe, ale chciałam zobaczyć czy algorytm analizując wartości ruchów może uznać, że inny ruch jest bardziej korzystny strategicznie. Brak obowiązku bicia pozwala lepiej zauważyć, jak algorytm minimax ocenia różne opcje ruchu, nie zawsze wybierając te, które wydają się intuicyjnie najlepsze.

Przy zwiększaniu głębokości rekursji w algorytmie minimax można zauważyć znaczną poprawę jakości decyzji podejmowanych przez SI. W przypadku głębokości rekursji równej 3, algorytm jest już całkiem kompetentny, ale przy głębokości 4 staje się już trudnym przeciwnikiem do pokonania. Jest to wynikiem bardziej szczegółowej analizy przyszłych ruchów, co pozwala SI na lepsze przewidywanie i unikanie błędów.

Dalszym krokiem w rozwoju tej gry mogłoby być zaimplementowanie bardziej zaawansowanych technik SI, takich jak uczenie maszynowe, aby jeszcze bardziej poprawić zdolności przewidywania ruchów przeciwnika i dostosowywanie strategii w czasie rzeczywistym. Techniki uczenia maszynowego mogłyby umożliwić SI naukę na podstawie rozgrywanych partii i dostosowywanie swojego stylu gry do konkretnych przeciwników, co dodatkowo zwiększyłoby jej skuteczność i atrakcyjność rozgrywki.

3 Literatura

- ROSYJSKIE WARCABY - REGUŁY GRY
- Algorytm min-max
- Algorytm Minimax
- strategia minimaksowa
- Warcaby - problemy i porady
- Przykładowa implementacja
- Rady co do implementacji