Kamil Breczko

Algorytm do gry great escape

1 listopada 2017

Spis treści

1. Krótki opis gry . . . 3
2. Wykorzystany algorytm . . . 3
3. Szczegółowy opis rozwiązania . . . 3
3.0.1. Ruch na planszy . . . 3
3.1. Postawnie ściany . . . 3
3.1.1. Ustawienie . . . 3
4. Wnioski z testowania . . . 4
5. Możliwe ulepszenia . . . 4

1. Krótki opis gry

Great escape jest to turowa gra, która jest rozgrywana na planszy o wymiarach 9x9 kwadratowych. Początkowo każdy z graczy zaczyna od osobnej ścianki. Celem gry jest dotarcie na drugą stronę planszy przed innymi graczami. Każdy z graczy może wykonywać tylko 4 podstawowe ruchy: w prawo, w lewo, do góry, w dół. Oprócz podanych ruchów, gracz może utudnić przejście ścieżki przeciwnikowi, stawiając ścianę. Ściany mogą być postawione w pozycji pionowej i poziomej na planszy. Żadna z ścian nie może się krzyżować. Gracz przegrywa za niepoprawne postawienie ściany lub niedozwolone poruszenie się na planszy.

2. Wykorzystany algorytm

Rozwiązanie do gry "great escape"wykorzystuje algorytm minimax (metoda minimalizowania maksymalnych możliwych strat) o głębokości 1, a poszukiwanie najkrótszej ścieżki do celu, obliczana jest za pomocą algorytmu A*.

3. Szczegółowy opis rozwiązania

Algorytm należy podzielić na 2 części: algorytm do ustawienia poprawnie ścian oraz algorytm do wybierania najkorzystniejszego ruchu w kierunku celu. Podczas podejmowania decyzji, największy koszt ponosi gdy jest na pozycji przegranej. To znaczy że rozwiązaniem jest nie przegranie danej rozgrywki. Aby wyznaczyć aktualną statystyke, początkowo sprawdzamy stan gry, a dokładnie algorytm analizuje każdego z graczy prostą heurystyką. Heurystyka polega na obliczeniu bezpośredniej ścieżki do celu, nie zwracając uwagi na przeszkody. Z otrzymanych danych można już stwierdzić prawdopodobieństwo wygranej i przegranej. Jeśli nasz gracz jest na pozycji przegranej to zostaje uruchomiony algorytm stawiający ścianę dla gracza, który ma duże prawdopodobieństwo wygranej, w przeciwnym przypadku wykonywany jest ruch na planszy.

3.0.1. Ruch na planszy

Dany przypadek jest wykonywany tylko wtedy gdy nasz gracz jest na pozycji wygranej. Aby wykonać najkorzystniejszy dla gracza ruch, należy obliczyć najkrótszą ścieżka do celu. Program w tym przypadku używa algorytmu A*, korzystający z prostej heurystyki. Heurystyka polega na obliczeniu bezpośredniej odległości od danej pozycji do celu. W najlepszym przypadku algorytm A* przejdzie przez n wierzchołków. Z obserwacji wyników testów, w najgorszym przypadku, jeśli w grze zostały umieszczone ściany to może przeanalizować nawet wszystkie wierzchołki (w przypadku planszy 9x9 jest to 72 wierzchołków). Ze względu na ograniczenia czasowe w sprawdzarce, które wymagają aby podjąć decyzje w danej turze w mniej niż 100ms, algorytm A* analizuje maksymalnie 30 wierzchołków. Algorytm kończy się, kiedy dojdzie do oczekiwanej krawędzi. W celu przyśpieszenia wyszukiwania sąsiadów, plansza z grą jest reprezentowana przez macierz sąsiedztwa.

3.1. Postawnie ściany

Dany przypadek jest wykonywany tylko wtedy gdy nasz gracz jest na pozycji przegranej. Aby odwrócić losy gry, algorytm stara się utrudnić ścieżkę przeciwnikowi, który jest w danej chwili na pozycji wygranej. Aby wyznaczyć ścieżkę, przez którą prawdopodobnie będzie podążał przeciwnik używany jest algorytm A*. Po wyznaczeniu ścieżki, rozważane są pierwsze 4 kroki przeciwnika. Następnie wykonywana jest próba postawiania ścianki, odpowiedniej do kierunku w jakim podąża przeciwnik. Algorytm wykorzystuje różne kombinacje aby postawić ściankę na danym polu, przesuniecie w góre, dół lub bok. W przypadku gdy postawienie ścianki jest niemożliwe, wykonywany jest ruch na planszy.

3.1.1. Ustawienie

Rodzaj ścianek jest wybierany w zależnośći od celu przeciwnika, jeśli porusza się w poziomie to wstawiane są tylko pionowe ścianki, w przeciwnym przypadku poziome ścianki.

4. Wnioski z testowania

Z obserwacji testów, zostało wprowadzone wiele ulepszeń. Jeśli gracz jest na pozycji wygranej to powinien wykonywać ruch przemieszczania do osiągnięcia celu. Jeśli gracz jest na pozycji wygranej i zdecyduje się na umieszczenie ściany to w najgorszym przypadku wyrównuje się zwycięstwo, ponieważ koszt ominięcia ścianki dla przeciwnika może kosztować tylko jedną turę. Dla gracza także nie opłacalne jest stawiania pod rząd więcej niż 4 ścianki (tzn. umieszczanie ścianki na każdym kroku w najkrótszej ścieżce przeciwnika), ponieważ nie tylko utrudnia to przejście przeciwnikowi do celu ale także naszej postaci.

Inną propozycją modyfikacji algorytmu A*, jest dotarcie do połowy celu. Z wyników obserwacji, w niektórych przypadkach program może źle działać. Postać porusza się tylko na dwóch polach.

5. Możliwe ulepszenia

Możliwe usprawnienia w algorytmie nie zważając na ograniczenia czasowe:
— Ulepszenie algorytmu A*, który mógłby analizować całą plansze aby znaleźć najkrótszą ścieżkę. Aktualnie jest przerywany po przeglądnięciu 30 wierzchołków.
— Ulepszenie algorytmu do stawiania ścian. Aktualnie rodzaj ściany jest wybierany analizując cel gracza. Jeśli cel gracza jest na bokach planszy, wtedy ściany są stawiane w pionie, w przeciwnym przypadku poziomie.
— Zabezpieczenie przed zamknięciem przeciwnika ściankami. Przy stategii podanej powyżej, przypadki gdy przeciwnik jest blokowany ścianami zdarzają się rzadko.
— Ulepszenie alogrytmu Minimax, który mógłby dokładniej określać aktualny stan gry. Minimax mógłby używać algorytmu A* zamiast prostej heurystyki.
— Wybór miedzy postawiniem ściany czy wykonaniem ruchu na planszy, mógłby zależeć od najkorzystniejszych ruchów w przyszłości. Analizując grę za pomocą minimax do głębokości 3, gdzie aktualnie jest 1.