

CORBA - Dyna Blaster

Artur Kłopotek

6 lutego 2009

1. Zaimplementowane interfejsy

Zaimplementowano zarówno wersję lokalną jak i zdalną (CORBA) interfejsów:

- ICPlayerFactory – `com.arturklopotek.dyna.ai.corba.CAIPlayerFactory`
- IPlayerFactory – `com.arturklopotek.dyna.ai.AIPlayerFactory`

Faktyczna implementacja interfejsów znajduje się w ich wersjach lokalnych – wersje zdalne są jedynie „owijkami” wokół interfejsów lokalnych i wykorzystują wzorzec delegacji do przekierowania wywoływanych na ich rzecz metod.

2. Zarys działania algorytmu

Algorytm sztucznej inteligencji gracza jest stosunkowo prosty w działaniu oraz implementacji – bazuje on na dwóch spostrzeżeniach¹:

- gracz w dowolnej chwili może poruszać się co najwyżej w jednym z czterech kierunków, przy czym zazwyczaj liczba dozwolonych kierunków jest jeszcze mniejsza
- stan gry ulega szybkim i częstym zmianom

Biorąc pod uwagę te spostrzeżenia, zdecydowano się na metodę, która bazuje na punktacji poszczególnych pól. Liczba pól, w których kierunku możemy w danej chwili podążać jest niewielka (0-4), więc możemy w pojedynczej iteracji algorytmu każdemu polu przypisać liczbę punktów, która powinna być tym mniejsza (bliższa zeru), im korzystniejsze dla gracza jest pójście w ich kierunku. Następnie stan kierunku kontrolera przyjmie wartość, odpowiednią dla pola o najmniejszej liczbie punktów. O tym czy gracz powinien postawić w danym momencie bombę, decydują również proste zasady, które są niezależne od tego, jaka jest aktualna decyzja o kierunku kontrolera.

3. Zbieranie informacji o rozgrywce

Aby algorytm działał poprawnie, zbierane są na bieżąco informacje o trwającej rozgrywce. Należą do nich:

- dane o „bonusach”, które zebrali poszczególni gracze – zbierane są wyłącznie informacje o dodatkach, które powodują zwiększenie zasięgu stawianych bomb – informacje o tych dających dodatkowe bomby są nieistotne;
- dane o postawionych bombach – zbierane są informacje o zasięgu postawionej bomby oraz o chwili jej wybuchu – wyliczane są efektywne momenty wybuchu (tzn. uwzględniany jest fakt, że bomba może wybuchnąć w wyniku eksplozji innej bomby).

4. Punktowanie sąsiednich pól

W każdej iteracji algorytmu, sąsiednim polom, do których możliwe jest przejście z pola aktualnego, przypisywana jest liczba punktów. Jest ona wyliczana na podstawie najmniejszej odległości od danego pola do:

- najbliższego bezpiecznego pola²
- najbliższego bonusu, zwiększającego zasięg bomby
- najbliższego, żyjącego i śmiertelnego przeciwnika

¹W pierwszym podejściu nie wzięto tego pod uwagę i sztuczną inteligencję oparto o algorytm znajdowania drogi A*. Szybko okazało się jednak, że w znacznym stopniu komplikuje to algorytm, a znalezioną drogę i tak trzeba stale aktualizować z powodu szybko zmieniającej się sytuacji na planszy. Zbytne wybieganie w przyszłość rozgrywki mija się więc z celem.

²Pole bezpieczne – pole, które nie jest w danym momencie w zasięgu rażenia żadnej bomby (uwzględniając także przeszkody stojące pomiędzy danym polem a bombą)

Powyższym odległościom zostały przypisane malejące wagi (w kolejności takiej jak powyżej). Suma iloczynów odległości i ich wag jest liczbą punktów, jakie przypisuje się danemu polu. Wynikiem tego, jest następujące zachowanie:

- gracz nigdy nie pójdzie do pola niebezpiecznego, jeśli może pójść do pola bezpiecznego;
- jeśli na planszy istnieje bonus zwiększający zasięg bomby, gracz pójdzie w jego kierunku;
- jeśli powyższe warunki nie rozstrzygają najlepszego kierunku, gracz pójdzie w kierunku przeciwnika;

Można modyfikować wartości wag, np. po to by gracz nie szedł w kierunku bonusu, który jest bardzo daleko, podczas gdy dość blisko znajduje się przeciwnik.

5. Stawianie bomb

Warunkiem postawienia bomby przez gracza w danym momencie jest spełnienie wszystkich poniższych warunków:

- odległość do bezpiecznego pola po postawieniu bomby wynosić będzie maksimum 2;
- po postawieniu bomby, przeciwnik znajdzie się w jej zasięgu;
- gracz nie znajduje się aktualnie w zasięgu innej bomby;

6. Podsumowanie

Mało skomplikowane zasady zastosowane w algorytmie oraz brak wybiegania w przyszłość rozgrywki, pozwoliły stworzyć prosty, ale dość efektywny i elastyczny algorytm. Daje się on dość prosto rozbudowywać – wystarczy zastosować bardziej wyrafinowane metody punktowania pól. Wartym zauważenia jest fakt, że pomimo iż algorytm nie „planuje” rozgrywki z wyprzedzeniem, a jedynie bazuje na obecnej sytuacji na planszy, sprawdza się w grze bardzo dobrze.