

Dyna Blaster AI

Modelowanie i Analiza Biznesowych Systemów Rozproszonych

Michał Kozłowski Bartosz Wesołowski

5 lutego 2009

1 Inspiracje

Cały kod programu został napisany przez autorów niniejszego sprawozdania. Inspiracje innymi botami, jeśli istnieją, pozostają tylko na poziomie koncepcyjnym. Niestety konkurencja nie chciała zdradzić zbyt wielu szczegółów.

2 Jak to działa

2.1 Testy

Na potrzeby ciągłego sprawdzania, czy napisany kod spełnia określone oczekiwania, napisane zostały testy, które pozwalają zobaczyć, jak zachowa się bot w określonych sytuacjach. Najważniejsze testy znajdują się w klasach `PathfinderTest`, `BonusTest` oraz `VariousTests`.

2.2 Wyszukiwanie drogi

Podstawą działania bota jest inteligentne wyszukiwanie drogi. Stworzony algorytm jest zmodyfikowaną wersją A^* . Pozwala on na znajdowanie ścieżek z uwzględnieniem:

- dokładnych czasów wybuchów bomb (co do klatki)
- dokładnych zasięgów bomb (dość oczywiste)
- murków, króte zostaną wysadzone przez bombę do czasu dojścia tam bota
- bomb, które zdążą wybuchnąć do czasu dojścia tam bota

Algorytm pozwala również na poszukiwanie najbliższego schronienia, czyli pola, któremu, po naszym dojściu na nie, nie zagraża żadna bomba.

2.3 Strategia

Poruszanie się i stawianie bomb są od siebie zupełnie niezależne.

Bomba stawiana jest wtedy, gdy bot znajdzie się w bezpośrednim sąsiedztwie przeciwnika lub wtedy, gdy pomiędzy nim a przeciwnikiem jest tunel i eksplozja jest w stanie sięgnąć jego końca. Dodatkowym warunkiem postawienia bomby jest możliwość znalezienia schronienia po jej postawieniu.

Bot zawsze porusza się w kierunku najbliższego przeciwnika lub bonusu (w zależności od tego, który znajduje się bliżej). Odległość nie jest w żaden sposób estymowana, lecz jest wyznaczana za pomocą algorytmu wyszukiwania drogi, który w sposób dokładny jest w stanie określić liczbę ramek, jaką zajmie mu dojście do określonego pola.

2.4 Dodatkowe atrakcje

Przy wyszukiwaniu drogi przeciwnicy są traktowani jak bomby, a więc przechodzenie przez nich jest zabronione. Również dzięki temu bot "przewiduje" niejako skutki postawienia bomb przez przeciwników i jest w stanie wcześniej na nie zareagować.

Przy sprawdzaniu, czy można w danym miejscu postawić bombę, wyszukiwana jest droga do najbliższego bezpiecznego miejsca, traktując zarówno przeciwników, jak i kratkę gracza, jako bomby. Pozwala to na przewidzenie, czy po postawieniu bomby bot będzie w stanie jeszcze uciec.

W sytuacjach awaryjnych, gdy nie istnieje inna droga ucieczki, schronienie jest wyszukiwane bez uwzględniania żadnych dodatkowych sztucznych bomb. Traktujemy wtedy przeciwników jak zwykłe pola, po których można przechodzić.

3 Wymogi formalne

Implementacja klasy IPlayerFactory: `com.jdyna.players.kozwes.BomberFactory`.
Implementacja klasy ICPlayerFactory: `com.jdyna.corba.CBomberFactory`.

4 Uwagi

Podczas realizacji zadania zaimplementowano niezliczoną ilość pomysłów, które nie znalazły zastosowania, a ślad ich istnienia nie został dołączony do sprawozdania:

- algorytm Dijkstry, który przez krótki czas stanowił podstawę wyszukiwania ścieżki

- algorytm Floyda-Warshalla, którego efektywne wykorzystanie wymagałoby trudnego do oszacowania nakładu pracy

5 Konkurs

Bot zajął pierwsze miejsce w konkursie, rozgryając całą konkurencję i zabijając bez litości każdego, kto stanął mu na drodze do sławy.