

Dyna Blaster

Raport

1. Cel projektu

Celem projektu było zaimplementowanie sztucznej inteligencji dla gracza gry „Dyna Blaster”.

2. Faktorie

- Faktoria implementująca IPlayerFactory
`com.kaluza.mikolajczyk.dyna.local.factory.PsychoPlayerFactory`
- Faktoria dla obiektów ICPlayerFactory
`com.kaluza.mikolajczyk.dyna.corba.factory.PsychoCPlayerFactory`

3. Opis algorytmu sztucznej inteligencji

Algorytm sztucznej inteligencji można podzielić na 3 odrębne elementy:

a) moduł zajmujący się decyzjami dotyczącymi stawiania bomb (AIBombsPutter)

Decyduje o postawieniu bomby na podstawie kilku prostych warunków:

- przeciwnik znajduje się w linii poziomej lub pionowej do gracza
- między graczem a przeciwnikiem nie ma ścian (poprzedni punkt spełniony)
- aktualne pole jest bezpieczne lub jest w zasięgu bomby, która wybuchnie później niż po 24 klatkach (frame)
- przeciwnik znajduje się w odległości połowy zasięgu bomby lub w odległości mniejszej niż 3 klatki (cell)

Jeżeli wszystkie wyżej wymienione warunki zostaną spełnione to algorytm decyduje się na postawienie bomby.

b) moduł sterujący graczem podczas ataku (AIAttackDirectionMaker)

W tej części jest implementowany algorytm wyszukiwania ścieżki do przeciwnika metodą przeszukiwania w głąb. Aby jednak wyszukiwanie to nie było za długie to metoda podająca bezpośrednich sąsiadów dla danego pola odpowiednio układa je względem pozycji przeciwnika tak, by zwiększyć szansę szybkiego odnalezienia ścieżki. Jeżeli wyszukiwanie trwa za długo (badane na podstawie liczby iteracji) to wyszukiwanie jest przerywane. Dodatkowo jeżeli odnaleziona zostanie najkrótsza ścieżka (odległość manhatańska), to działanie algorytmu również zostaje wstrzymane.

W przypadku, gdy algorytm nie odnajdzie ścieżki do przeciwnika uruchamiany jest moduł wycofania. Natomiast, gdy ścieżka zostanie odnaleziona, to jest zapamiętywana, by później sprawdzić, czy przeciwnik nadal znajduje się na niej (np. nie rusza się, bądź idzie po tej ścieżce) dzięki czemu kolejne wyszukiwanie nie będzie potrzebne.

c) moduł sterujący graczem podczas wycofania (AIRetreatDirectionMaker)

Ta część algorytmu jest prostsza. Jej celem jest obranie najbardziej korzystnej pozycji.

Ocenia lokację gracza i wszystkich jego bezpośrednich sąsiadów (czyli poziom zagłębienia – 1). Oceny dokonuje na podstawie następujących atrybutów z odpowiednimi wagami wyznaczonymi empirycznie:

- liczba sąsiadów
- liczba bomb, w zasięgu których jest klatka
- czas do wybuchu bomb, w których pole jest w zasięgu – ostatecznie wycofano się z brania pod uwagę tego parametru
- odległości od przeciwników

Na podstawie tak ocenionych pól wybiera to, które ma najwyższą ocenę.

Dodatkowo ważną klasą dla wszystkich trzech jest klasa GameStateCollector zajmująca się zbieraniem wiedzy na temat aktualnego stanu gry.

4. Podsumowanie

Zespół uważa projekt za bardzo interesujący. Spędzono przy nim wiele wieczorów, ale nie tylko dlatego, że było trzeba. Miło było powspominać starego „Dyna Blastera”, odświeżyć kontakt z grą, a następnie zmierzyć się z własną wyobraźnią.

Jak napisano wcześniej, celem projektu było stworzenie modułu sztucznej inteligencji. Całą przyjemność polegała na tym, że jak już się zaczęło pracę, to efekty były natychmiastowe. Zaowocowało to ciągłymi grami ze stworzonym przez zespół „komputerem”. Dzięki takim walkom pojawiały się ciągle pomysły na ulepszanie zachowania sztucznego gracza. Wynikiem tego była wielokrotna przebudowa kodu praktycznie od początku. Koncepcja potrafiła się zmieniać tak często, jak liczba bramek w meczach piłki ręcznej reprezentacji Polski. Kolejna fala radości i pomysłów przysłała, gdy zaczęto walki z botami innych zespołów. Coś jednak należało wybrać.

Zespół zauważył wady swojej implementacji gracza. Część zajmująca się wyszukiwaniem drogi do przeciwnika nie jest doskonała, czasem nie znajduje ścieżki choć powinna. Innym elementem, który mógłby zostać poprawiony to część zajmująca się wycofywaniem zawodnika. Aktualnie została zaimplementowana ocena aktualnej pozycji i maksymalnie czterech bezpośrednich sąsiadów. Mogłoby to zostać rozszerzone do przeszukiwania o większej głębokości.

Pisanie tej części AI pochłonęło zespół tak bardzo, że Corbę odstawiono na później. Niestety na to później zabrakło już czasu i część ta została zrobiona w minimalnym stopniu – przy użyciu owijki dostarczonej przez prowadzącego.