

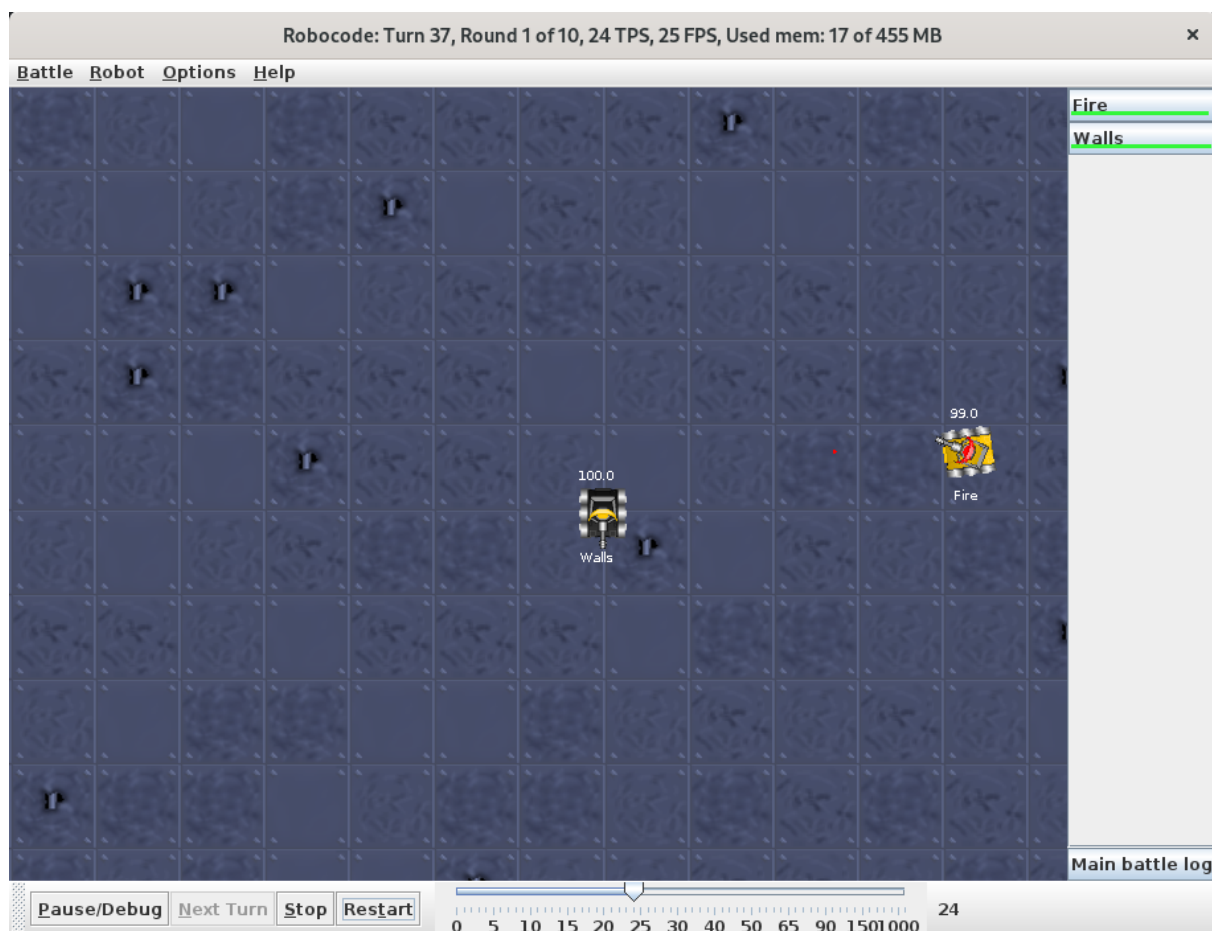
Zastosowanie Q-learningu do zaprogramowania robota w środowisku Robocode

Bartłomiej Plewnia
Jan Rodzoń

Wprowadzenie

Środowisko

Robocode to gra programistyczna pierwotnie mająca na celu pomoc w nauce programowania w Javie. Platforma ta pozwala na programowanie wirtualnych robotów, które walczą między sobą. Walka odbywa się na planszy 2D:



Rys 1. Przykładowy screen z walki czołgów na platformie Robocode

Gra polega na poruszaniu się czołgiem po planszy i niszczeniu przeciwników.

Najważniejsze zasady walki czołgów:

- czołg posiada w pełni obracającą się wieżę niezależnie od kadłuba
- na wieży zamontowane jest działo, z którego robot może strzelać
- każde trafienie w przeciwnika zabiera mu punkty życia, a strzelcowi dodaje punkty zwycięstwa
- pojazdy mogą się wzajemnie taranować, co również jest metodą na uszczuplenie puli życia przeciwnika
- robot posiada niezależnie obracający się radar, którym wykrywa inne pojazdy
- czołg posiada regenerujący się pasek energii, który zużywa na prowadzenie ostrzału
- strzelec decyduje o sile strzału - mocniejszy strzał chociaż zadaje więcej obrażeń to kosztuje więcej energii oraz leci wolniej

Turnieje

Platforma umożliwia rozgrywanie turniejów robotów. Turniej posiada konkretną ilość rund. Każda runda to starcie w trybie "battle royale" - ostatni na polu walki wygrywa.

Aby sklasyfikować uczestników turnieju wprowadzony jest system punktacji, dzięki czemu w końcowej klasyfikacji każdy robot pozycjonowany jest poprzez sumę punktów z wszystkich rund.

Najważniejsze zasady punktacji:

- punkty za przeżycie dłużej niż inni
- nagroda za zwycięstwo - pozostanie ostatni na polu bitwy
- punkty za zadane obrażenia
- bonus za zniszczenie czołgu przeciwnego - zadanie ostatniego ciosu powodującego zniszczenie

Cel projektu

Celem projektu jest zastosowanie Q-learningu do sterowania czołgiem w walkach robotów w środowisku Robocode. Przygotowany program będzie uczył się sterować robotem i starał się osiągnąć jak najlepszy wynik.

Koncepcja algorytmu

Zastosowany algorytm Q-learning uczy się jak najlepszego poruszania po planszy.
Algorytmy zarządzania strzelaniem oraz radarem są zaimplementowane statycznie.

Jako zbiór akcji określone zostały:

- pojeźdź w przód
- pojeźdź w tył
- pojeźdź w przód skręcając w prawo
- pojeźdź w przód skręcając w lewo
- pojeźdź w tył skręcając w prawo
- pojeźdź w tył skręcając w lewo

Stan określony jest przez:

- zdyskretyzowaną odległość od wroga w skali od 1 do 10
- kierunek, w którym jest przeciwnik - lewo, prawo, przód, tył
- kierunek pojazdu - lewo, prawo, przód, tył
- stan energii - niski, poniżej połowy, powyżej połowy, wysoki
- uderzenie w mur - tak lub nie
- bycie trafionym przez pocisk - tak lub nie

Algorytm penalizuje następujące wydarzenia:

- trafienie poprzez pocisk wroga
- taranowanie
- zderzenie ze ścianą
- nie trafienie pociskiem
- śmierć

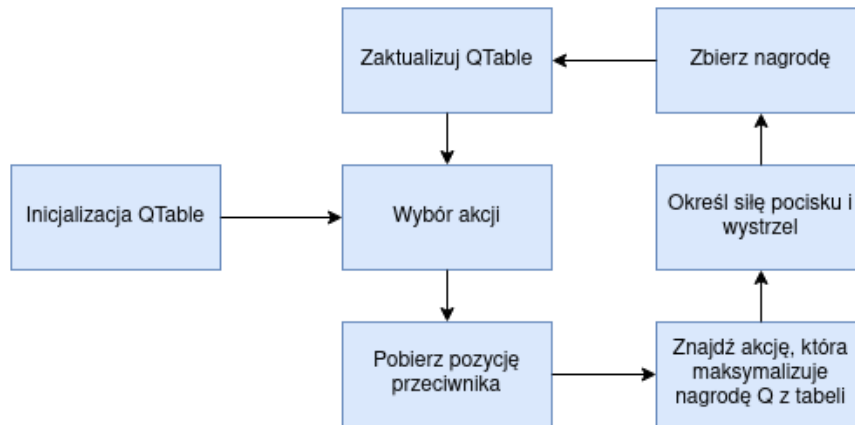
Algorytm nagradza następujące wydarzenia:

- trafienie przeciwnika pociskiem
- zwycięstwo

Algorytm posiada skalibrowane parametry:

- learning rate - 0.8
- discount factor - 0.95
- epsilon - 0.05

Diagram działania algorytmu



Rys 2. Diagram działania algorytmu.