

Jatank

Krzysztof Romanowski

Inżynieria wiedzy i uczenie maszynowe

wrzesień 2014

Spis treści

Wprowadzenie.....	3
Środowisko.....	3
Przebieg prac.....	3
Konceptja rozwiązania.....	4
Architektura systemu.....	5
Wyniki.....	6

Wprowadzenie

Projekt JaTank został zrealizowany w ramach przedmiotu Inżynieria Wiedzy i Uczenie Maszynowe. Jego celem było zastosowanie reguł do implementacji zachowania robota w środowisku Robocode.

Środowisko

Robocode – to [opensourcowa gra programistyczna](http://pl.wikipedia.org/wiki/Robocode) stworzona przez Mathewa Nelsona. Obecnie rozwijana jest głównie przez Flemminga N. Larsena oraz Pavla Šavarę. Została zaprojektowana jako aplikacja pomagająca w nauce [programowania w Javie](http://pl.wikipedia.org/wiki/Java). Zadaniem gracza jest napisanie przy użyciu tego właśnie [języka](http://pl.wikipedia.org/wiki/Język_programowania) wirtualnego [roboty](http://pl.wikipedia.org/wiki/Robot), który następnie bierze udział w walce ze stworzonymi przez innych pojazdami[1]. Początkujący użytkownicy mogą skorzystać z dostępnych, między innymi na stronie projektu, gotowych robotów, a bardziej zaawansowani programiści mają wręcz nieograniczone możliwości jeśli chodzi o wzbogacanie ich o nowe funkcje[2].

Za: <http://pl.wikipedia.org/wiki/Robocode>

JaTank jest planowany na podzbór możliwości środowiska: walki jeden na jeden.

Przebieg prac

JaTank jest zamiennikiem oryginalnego pomysłu ScaTank. ScaTank miał opierać się na podobnych koncepcjach oraz być napisany w języku Scala. Dokumentacja Robocode wskazywała że jest to wykonalne. Jednak okazało się to prawdziwe tylko w stosunku do składni: biblioteka standardowa nie była ładowana przez Robocode.

Po kilku godzinach prób zmodyfikowania środowiska Robocode została podjęta decyzja prześcia na Java'e.

Konceptcja rozwiązania

Podstawą działania systemu jest garść reguł. Reguły są przystosowane do 2 aspektów działania robota:

- strzelanie
- poruszanie się

Zastawany alorytm jest następujący:

- Wybierz losową kolejność reguł
- Wybierz 1 regułę która pasuje do sytuacji
- Wykonaj regułę

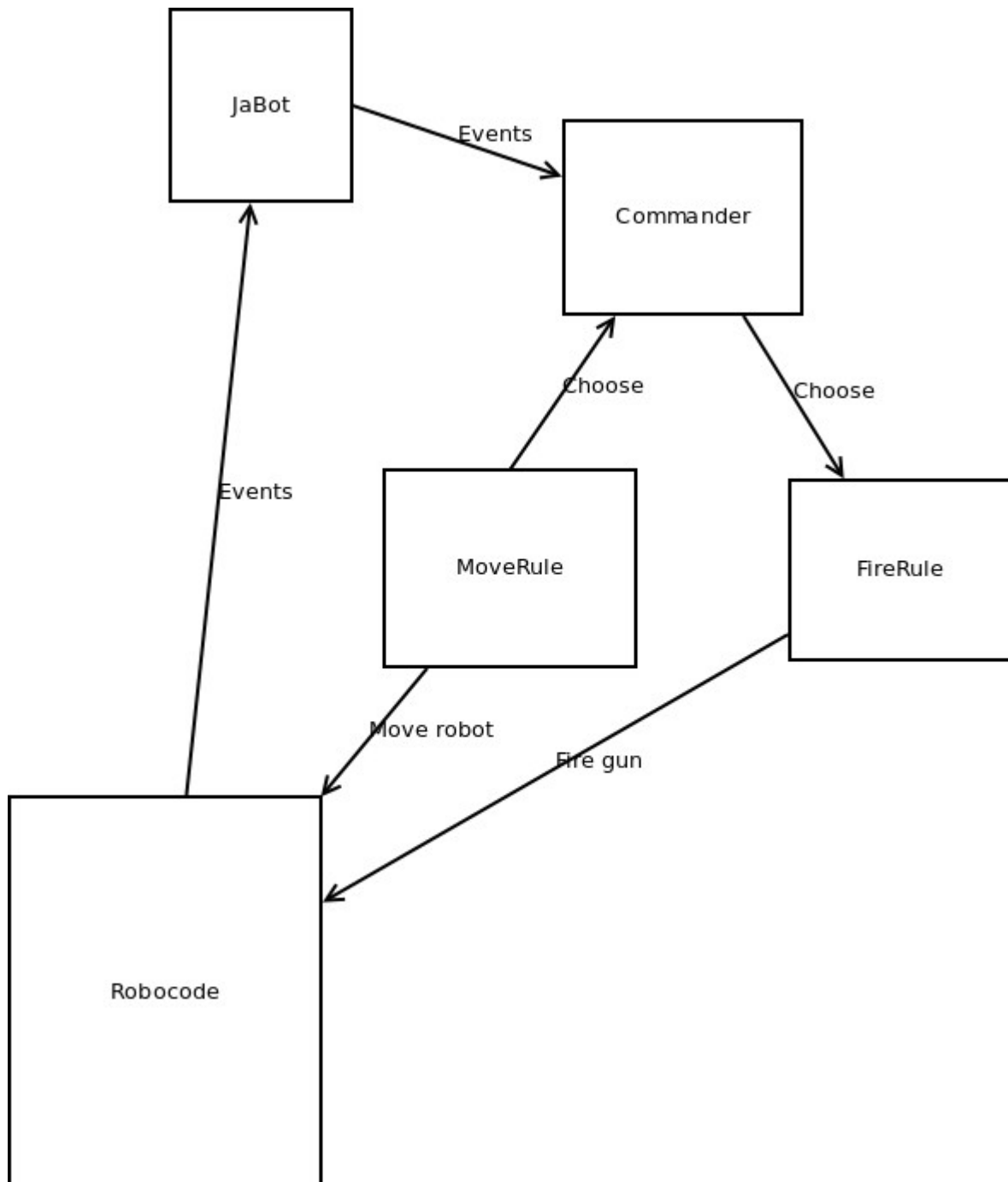
W systemie wyróżniamy dwie sytuację dla których możemy aplikować reguły:

- dostrzeżono przeciwnego robota
- nasz robot został trafiony

Odpowiadają im redzaje regół:

- strzelanie (gdy dostrzeżono wroga)
- poruszenie (gdy zostaliśmy trafieni)

Architektura systemu



Wyniki

Nasz robot poprawnie aplikował reguły co przyczyniało się do lepszych wyników. Osiągane wyniki pozwalały na równorzedną walkę z domyslnymi robotami, jednak starcie z interaktywnym robotem (symulacja pewnych technik zachowania) uwidaczniały słabości naszego robota.

Niedoskonałości nie wynikały z słabości silnika tylko z niedostatecznej ilości regół i słabego pokrycia zbioru możliwych sytuacji.

Wprowadzona losowość sprawiała że system powinien być trudnym przeciwnikiem dla sytemów uczących się zachowania robota.