

Kamil Breczko

# Bot do gry Ghost in the Cell

9 grudnia 2017

Spis treści

1. Opis gry . . . . .	3
2. Opis algorytmu . . . . .	3

## 1. Opis gry

Gra "*Ghost in the Cell*" rozgrywa się na planszy z dwoma graczami, na której umieszczono fabryki (od 7 do 15 fabryk). Początkowo każdy z graczy posiada po jednej fabryce, reszta fabryk jest neutralna. Fabryka może produkować cyborgi, które mają za zadanie bronić daną fabrykę i atakować inne. Celem gry jest wyprodukowanie maksymalnej ilości cyborgów, aby zniszczyć przeciwników. W każdej turze gracz może podjąć nieograniczoną liczbę ruchów:

- przeniesienie cyborgów do zaprzyjaźnionej fabryki;
- wysłanie cyborgów do wrogiej fabryki, w celu zniszczenia;
- wysłanie bomb, w celu uszkodzenia cyborgów stacjonujących w fabryce oraz zablokowanie produkcji cyborgów na 5 tur; (ilość bomb: 2)
- zwiększenie produkcji fabryki kosztem 10 cyborgów;

## 2. Opis algorytmu

Rozwiązanie do gry "*Ghost in the Cell*" wykorzystuje drzewa decyzyjne. Algorytm podejmuje ciąg decyzji analizując każdą fabrykę z osobna, czyli każda decyzja z danej fabryki nie zależy od stanu innych fabryk gracza. Aby zwyciężyć, algorytm przyjmuje strategię ofensywną.

Zastosowane techniki:

- Co 50 tur wybierana jest "największa" fabryka, pod względem ilości przebywających tam cyborgów. Po wybraniu wrogiej fabryki, wyszukiwana jest fabryka gracza, która znajduje się najbliżej. Następnie wysyłana jest jedna jednostka bomby, w celu zniszczenia znajdujących się tam cyborgów oraz spowolnienie ich produkcji;
- W każdej fabryce gracza jest uruchamiana funkcja oceniająca fabryki przeciwnika, w celu określenia najlepszego celu. Dana funkcja wykorzystuje informacje: odległość, ilość cyborgów w fabryce gracza, ilość cyborgów w fabryce przeciwnika, ilość cyborgów wysłanych do fabryki przeciwnika;

```
rateTarget() {
    rateDistance = 2 / (distance(myFactory, enemyFactory));
    sumCyborgs = -cyborgs;
    rateProduction = 10 * production;
    for (Troop troop : findTroopsByTargetFactory(id, troops)) {
        sumCyborgs += (troop.getOwner() == 1) ? troop.getCyborgs() : -troop.getCyborgs();
    }
    result = (sumCyborgs + rateProduction) * rateDistance;
    if (result > 0) {
        return result * 2;
    }
    return result * 1.5;
}
```

Wykonując ruch w danej turze, uwzględniane są wszystkie fabryki. Z każdej fabryki gracza są wysyłane cyborgi do fabryk przeciwnika, aż ilość zrówna się z zerem. Początkowe cele wybierane są zgodnie z oceną funkcji, a ilość cyborgów zgodnie ze wzorem:  $(\text{ilość\_cyborgów})/2$ .