Roboty monitorujace skażenie środowiska - symulator V-REP

Anna Wujek
Łukasz Korpal
Wiktor Ślęczka

10 czerwca 2016

Spis treści

1 Zadanie

Celem projektu było stworzenie symulatora mobilnej sieci ad hoc (MANET) do monitorowania skażenia środowiska naturalnego. Węzłami sieci są roboty mobilne, wyposażone w czujniki oraz komunikujące się między sobą. Zadaniem robotów jest lokalizacja chmury skażenia i otoczenie jej robotami, monitorującymi jej położenie i granice, przy założeniu utrzymania spójności sieci.

2 Projekt systemu

2.1 Założenia

- Roboty potrafią się na bieżąco lokalizować.
- System składa się z pewnej liczby robotów mobilnych.
- Roboty mobilne działają autonomicznie i potrafią same zorganizować sieć.
- Roboty są wyposażone w czujniki pozwalające im wykrywać poziom skażenia oraz przeszkody.
- Roboty są wyposażone w urządzenia pozwalające im się między sobą komunikować się między sobą.
- Chmura skażenia jest odgórnie określona i niezmienna w trakcie trwania symulacji.
- Nieznana jest mapa przestrzeni roboczej, ale znane są jej granice i kształt - prostokąt.

2.2 Scenariusz działania

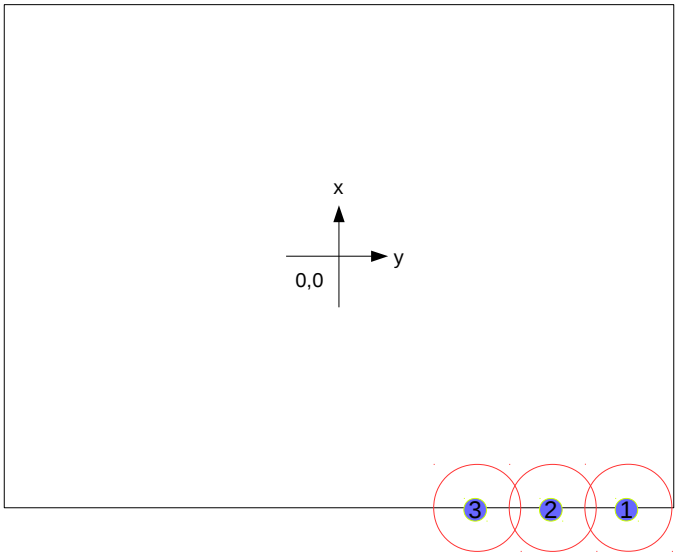
Algorytm realizowany przez każdego robota:

1. Warunek początkowy: roboty znajdują się w dowolnym punkcie przestrzeni na mapie. Z punktu tego musi istnieć droga prowadząca do obszaru poszukiwań.
2. Przemieść się do punktu początkowego (rys. 1), zależnego od liczby robotów, wymiarów przestrzeni poszukiwań i zasięgu czujników (rys. 2), zgodnie ze wzorem:

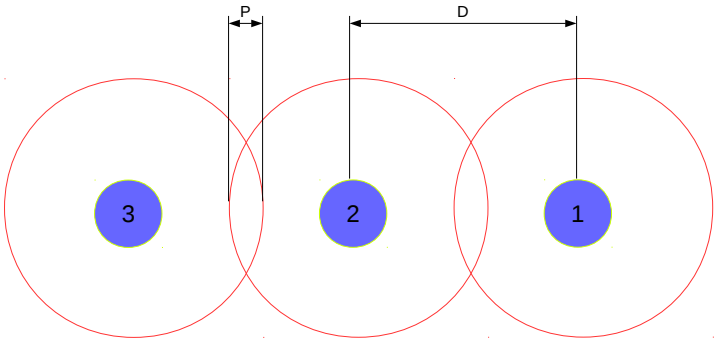
$$x = -length/2 \tag{1}$$
$$y = width/2 + distance * robot\_number + distance/2 \tag{2}$$

W obliczeniach wykorzystywane są następujące parametry:

$$max\_spaces = sensor\_range \cdot precision\_range \tag{3}$$
$$no\_stages = ceil(width/(max\_spaces \cdot no\_robots)) \tag{4}$$
$$spaces\_width = width/no\_stages \tag{5}$$
$$distance = spaces\_width/no\_robots \tag{6}$$



Rysunek 1: Ustawienie początkowe robotów



Rysunek 2: Sposób ustawienia robotów w szeregu

3. Wyznacz kolejny punkt docelowy, zgodnie ze wzorem:

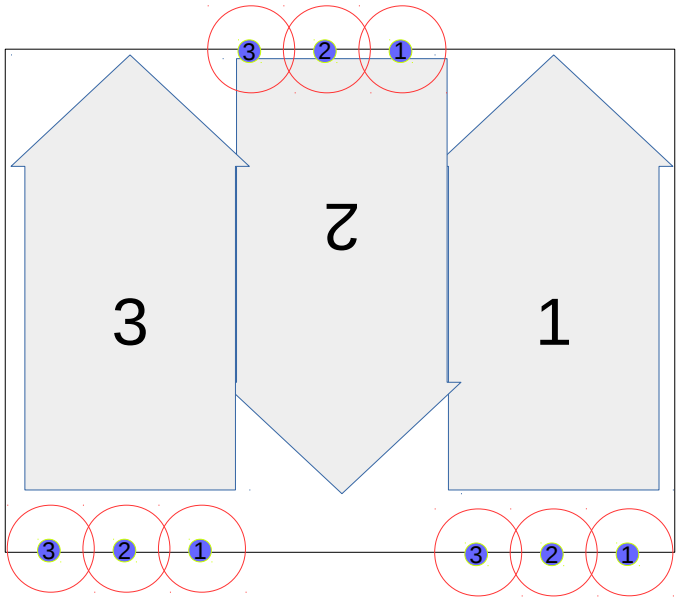
$$x_n = direction \cdot length/2 + direction + substage\_length \cdot substage \quad (7)$$
$$y_n = width/2 + distance \cdot robot\_number + stage \cdot spaces\_width + distance \quad (8)$$

4. Udaj się do punktu docelowego, monitorując skażenie oraz wykrywając przeszkody.

5. Jeśli wykryto przeszkodę, omiń ją algorytmem Bug2. Wyznacz następny punkt docelowy, jest aktualny został minięty.

6. Jeśli wykryto chmurę skażenia, zatrzymaj się.

7. Jeśli cały obszar poszukiwań sprawdzony, to koniec algorytmu; jeśli nie, to przejdź do punktu 3.



Rysunek 3: Ruch szeregu robotów

2.3 Struktura systemu

System prezentowany w projekcie składa się z następujących elementów:

- symulator przestrzeni roboczej
- symulator robotów
- symulator chmury skażenia
- komunikacja

  zapewnia przekazywanie informacji pomiędzy poszczególnymi symulatorami.

2.3.1 Symulator przestrzeni roboczej

Symulator ten jest odpowiedzialny za wizualizację całej symulacji - przestrzeni roboczej, przeszkód, robotów oraz chmury skażenia. Generuje także fizyczne oddziaływania między elementami, np. kolizje oraz odczyty czujników (oprócz czujników skażenia).

2.3.2 Symulator chmury skażenia

Symulator ten jest odpowiedzialny za generowanie odczytów czujników skażenia. Odczyt jest obliczany na podstawie założonego wcześniej kształtu chmury oraz sposobu, w jaki skażenie rośnie wgłąb chmury.

2.3.3 Symulator robota

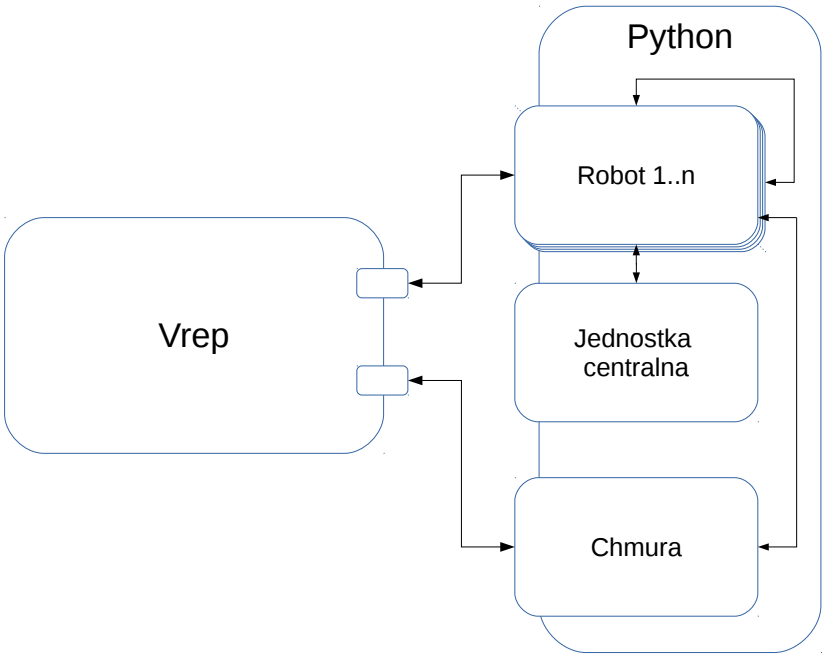
W systemie symulator ten będzie zwielokrotniony, aby uzyskać grupę kilku robotów. Odpowiedzialny jest za wyznaczanie sterowania robotów w zależności od odczytów z czujników i wewnętrznego imperatywu, czyli zadania, jakie roboty realizują.

2.3.4 Komunikacja

Aby cały system działał poprawnie, poszczególne elementy muszą móc się ze sobą komunikować. W systemie rozróżniamy następujące kanały komunikacyjne i przesyłane informacje:

- Symulator przestrzeni roboczej → Symulator robota:
  - odczyty z czujników (oprócz czujnika skażenia)
  - pozycja robota (położenie + orientacja)
- Symulator robota → Symulator przestrzeni roboczej:
  - sterowanie poszczególnymi silnikami
- Symulator chmury skażenia → Symulator robota:
  - odczyt czujnika skażenia
- Symulator robota → Symulator robota:
  - ostatni punkt docelowy, do którego udało się dotrzeć
  - informacja o wykrytej chmurze skażenia

2.4 Wstępny projekt systemu



Rysunek 4: Schemat działania systemu.

Dla robotów, jednostki centralnej oraz chmury stworzone zostaną procesy, które będą się komunikować z symulatorem V-rep:

- procesy robotów będą wysyłać sterowania i odbierać odczyty z czujników przeszkód z symulatora V-rep (V-rep będzie symulować rzeczywiste roboty i rzeczywiste środowisko);
- możliwa będzie komunikacja pomiędzy robotami;
- jednostka centralna będzie odbierać od robotów odczyty skażenia;
- stworzony zostanie osobny proces, który będzie symulował chmurę skażenia; proces ten symulował będzie wartości odczytów czujników skażenia i przekazywał informacje o kształcie i położeniu chmury do V-repa.

3 Realizacja

3.1 Plik konfiguracyjny

Plik konfiguracyjny powinien zawierać informacje o planszy, robotach i chmurze – w trzech rozdzielonych modułach.

- Moduł [board] opisuje wielkość sektora poszukiwań.
  Posiada następujące parametry:
  - width – szerokość planszy

  - length – długość planszy

- Moduł [robots] opisuje roboty biorące udział w symulacji.
  Posiada następujące parametry:
  - count – liczba robotów
  - sensor_range – zasięg czujników toksyczności
  - precision_range – parametr wymaganej precyzji, ogranicza maksymalny zasięg czujników, wpływając na mniejsze odległości między robotami (zwiększa zmienną P, por. rys. 2)

- Moduł [cloud] opisuje położenie i wielkość chmury.
  Posiada następujące parametry::
  - center – położenie środka chmury
  - radius – średnica chmury

3.2 Komunikacja

Komunikacja przebiega na zasadzie ogłaszania aktualnych informacji na swój temat przez roboty (oraz chmurę). Informacje przesyłane są za pomocą struktury JSON do gniazd, przypisywanych każdemu modułowi osobno. Wiadomości opatrzone są typem, pozwalającym sklasyfikować komunikat. Dane tworzone są w postaci usystematyzowanych słowników, których struktura została odgórnie ustalona. Komunikaty są wysyłane i odbierane asynchronicznie.

3.2.1 Roboty

Roboty wysyłają informacje o swoim postępie w wiadomościach typu "progress". Zawierają one następujące elementy:

- żobot- numer identyfikacyjny robota
- śubstage- postęp robota wzdłuż wyznaczonej ścieżki
- śtage- postęp robota w numerze ścieżki

Poniżej przedstawiono przykład tworzenia wiadomości o postępie robota.

```
def broadcast_info(self):
    message = str(self.robot._name)+str(self.substage)
    message = dict()
    message["robot"] = self.robot.name
    message["stage"] = self.stage
    message["substage"] = self.substage
    message["type"] = "progress"
    self.robot.commutron.broadcast(json.dumps(message))
```

Dodatkowo, symulując działanie czujników toksyczności, wysyłają zapytania do procesu chmury o stężenie w swoim położeniu. Są to

3.2.2 Chmura

Skrypt odpowiadający za symulację chmury wysyła informację o jej położeniu do robotów w wiadomościach typu ćloudread". Zawierają one następujące elementy:

- ćoncentration- stężenie toksycznych substancji
- żobot- numer robota, z którym przebiega komunikacja
- "position- pozycja robota

Poniżej przedstawiono przykład tworzenia wiadomości o chmurze.

```
res = dict()
res['concentration'] = self.concentration(data["position"])
res['robot'] = data['robot']
res['position'] = data['position']
res['type'] = "cloudread"
```

3.3 Roboty

Wykorzystywane w realizacji zadania roboty Pionieer p3dx wyposażone w ultradźwiękowe czujniki odległości - będą one zapewnione przez symulator V-Rep. Posiadają one napęd różnicowy, pozwalający na dużą swobodę poruszania się po symulowanej przestrzeni. Są przystosowane do rozbudowy o dodatkowe urządzenia – dzięki czemu mogły by być łatwo dostosowane do realizacji zadania lokalizacji chmury. Dodatkowo, czujniki stężenia toksycznych substancji będą realizowane w skryptach Pythona, ułatwiając jej symulację. Informacje o jej położeniu i kształcie będą przekazywane do Vrepa, gdzie nastąpi ich wizualizacja.

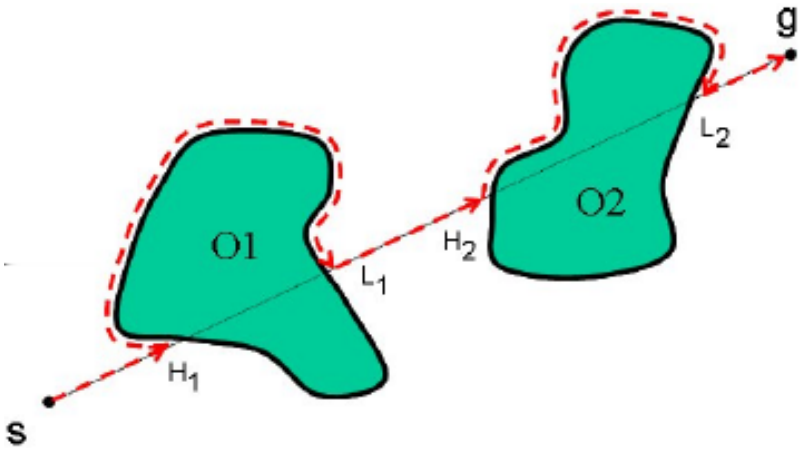
3.4 Algorytmy

3.4.1 Algorytm poszukiwania chmury

Roboty rozpoczynają pracę od ustawienia się w formacji linii – maksymalna odległość między robotami jest określona wzorem: $d_{max} = 0.7 * min(max(z_1, 2 * z_2))$
Gdzie:
d_{max} – maksymalna odległość między robotami
z_1 – zasięg urządzeń komunikacyjnych robotów
z_2 – zasięg czujników odległości
Roboty w ten sposób zaczynają poruszać się do przodu do momentu dotarcia do przeszkody, toksycznej chmury lub granicy obszaru poszukiwań. W przypadku powyższych wydarzeń, uruchamiane są odpowiednie algorytmy. Gdy roboty dotrą do końca obszaru poszukiwań, zatrzymują się, przesuwają w bok i kontynuują poszukiwania w drugą stronę, granicząc z poprzednim obszarem poszukiwań.

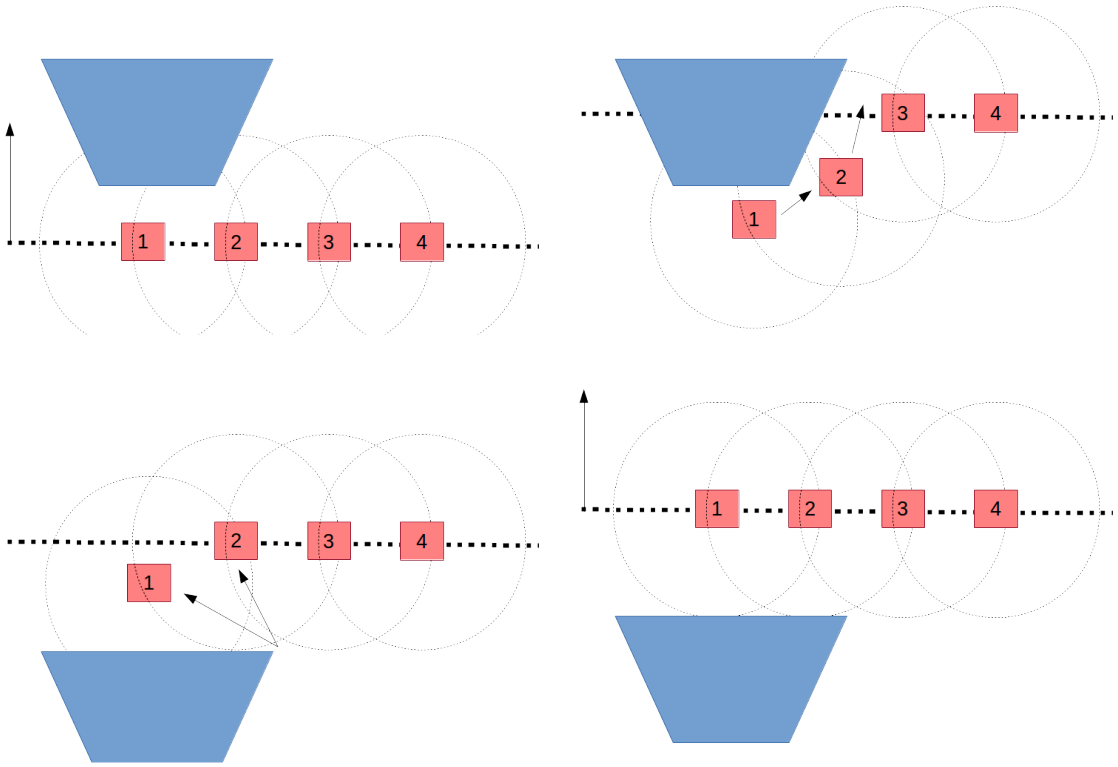
3.4.2 Algorytm bug 2

Algorytm ten będzie stosowany w przypadku napotkania przeszkody przez robota. Polega on na podążaniu wzdłuż ściany przeszkody do momentu osiągnięcia punktu leżącego za napotkaną ścianą, na przedłużeniu początkowej ścieżki ruchu.



Rysunek 5: Działanie algorytmu bug 2

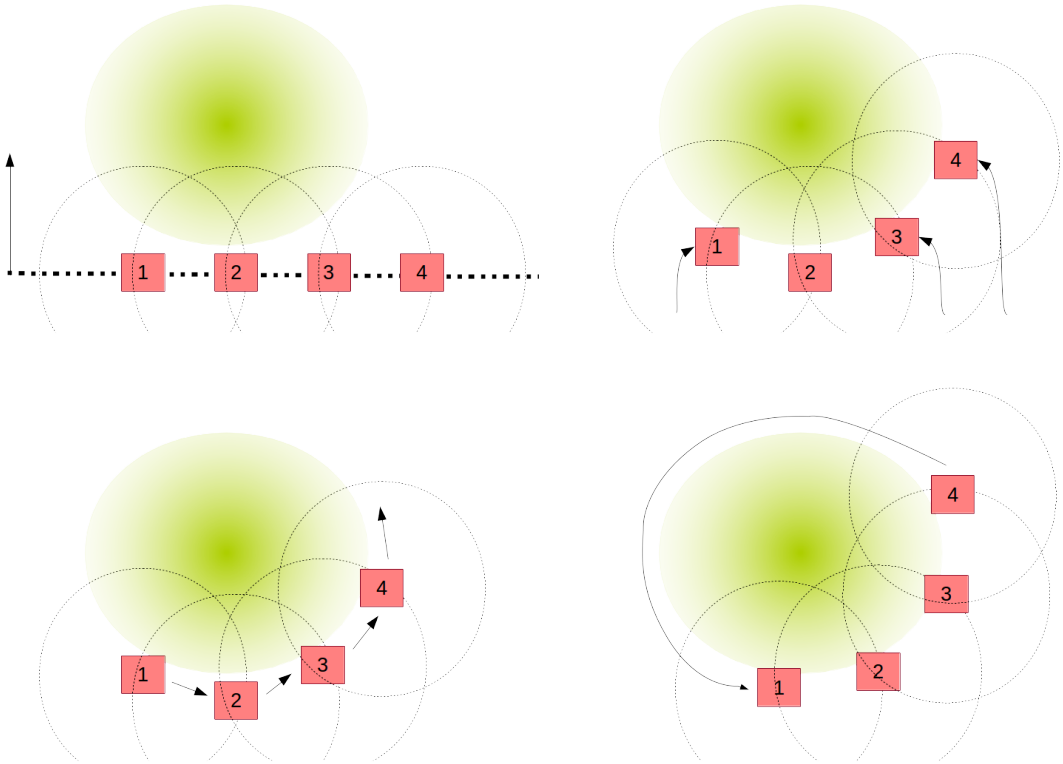
Jego działanie zostało przedstawione na rysunku 5. W zastosowaniu dla grupy robotów zmodyfikujemy ten algorytm w następujący sposób: Gdy robot/roboty wykryją przeszkodę, oceniane jest jej położenie względem środka szeregu. Większość szeregu powinna poruszać się dalej, natomiast pozostałe roboty powinny zacząć omijać przeszkodę w kierunku większości za pomocą algorytmu bug 2, nie przerywając połączenia z siecią. Jego działanie zaprezentowano na rysunku 6.



Rysunek 6: Działanie algorytmu bug 2 dla wielu robotów. Ścianę napotyka jako pierwszy robot 1, robot 2 także ją zauważa. Znajduje się na lewo od środka ciężkości linii, więc roboty z tej strony zaczynają omijać przeszkodę, korzystając z algorytmu bug 2 oraz zachowując spójność sieci.

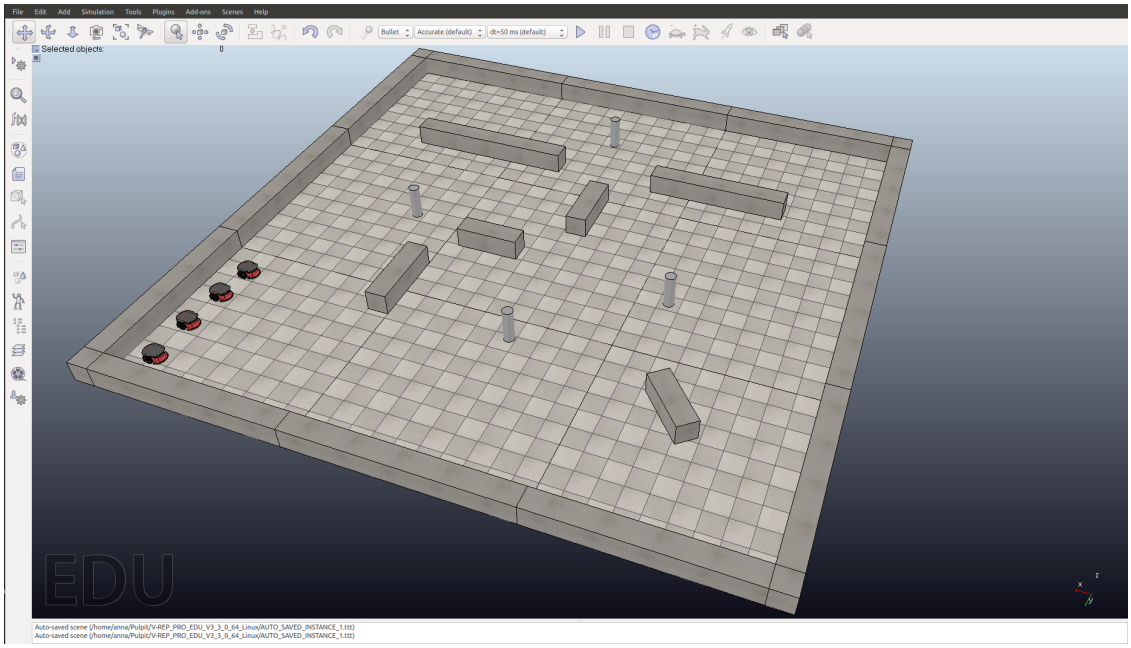
3.4.3 Algorytm otaczania chmury

W momencie napotkania chmury przez któregokolwiek robota, zatrzymuje się on (po osiągnięciu natężenia toksyn określonego programowo). Pozostałe roboty jadą dalej, po łuku – tak, aby nie przerwać połączenia z siecią. Gdy napotkają określone stężenie chmury, zatrzymują się. Gdy wszystkie roboty zatrzymały się, zwracają się w jedną stronę i zaczynają okrążać chmurę, utrzymując się na granicy stężenia oraz zachowując spójność sieci. Jego działanie przedstawiono na rysunku 7.

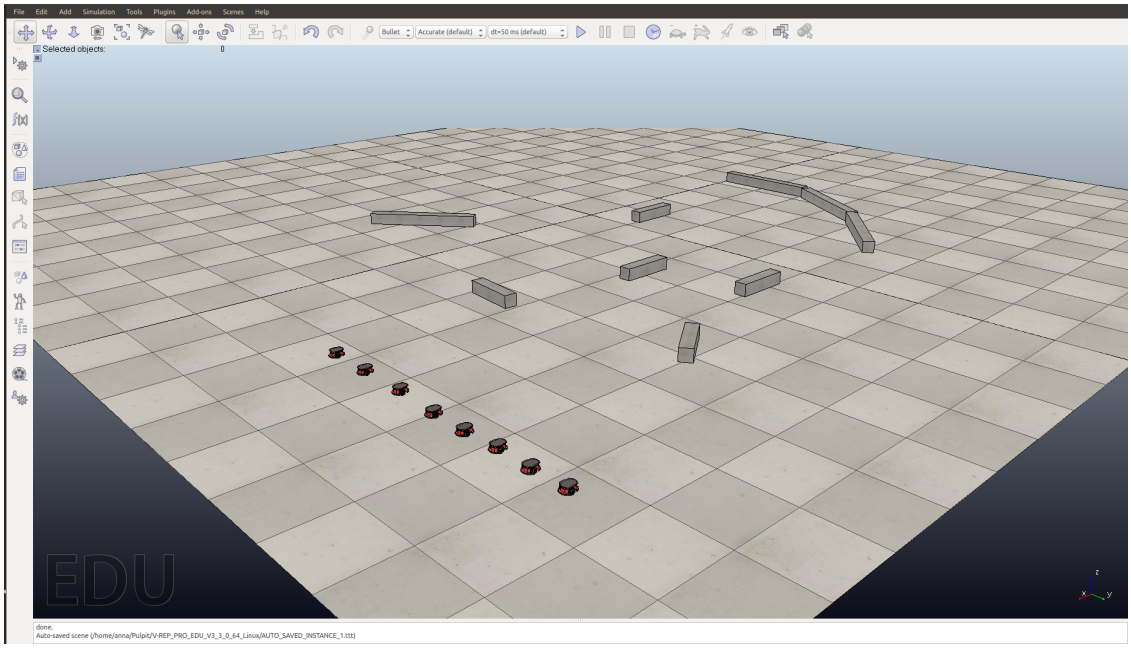


Rysunek 7: Działanie algorytmu otaczania chmury. Robot 2 napotyka chmurę, pozostałe roboty zbliżają się do chmury aż do wartości granicznej, następnie cała sieć zaczyna krążyć wokół chmury, monitorując jej kształt i położenie.

3.5 Przykładowe mapy środowiska



(a) Mały teren



(b) Duży teren

Rysunek 8: Przykładowe mapy środowiska