

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по практической работе №1
по дисциплине «Интеллектуальные системы»
Тема: Определение местонахождения игрока на поле

Студенты гр. 6304

Некрасов Н. А.

Прозорова А. Д.

Преподаватель

Беляев С.А.

Санкт-Петербург

2020

Цель работы

Целью работы является решение задачи позиционирования агента в пространстве с учётом погрешности получения информации из окружающей среды.

Задание

Необходимо разработать программу, имитирующую игрока виртуального футбола. Программа должна решать следующие задачи:

1) Получить из командной строки начальные координаты игрока и число (скорость вращения) и командой `move` переместить игрока в заданные координаты.

2) После получения сообщения о начале игры начать вращать игрока с заданной скоростью (поворот на каждом такте). Следует отметить, что движение игрока начнётся только после команды рефери `play_on`.

3) По информации, полученной от сервера (сообщение `see`) на каждом такте, вычислить координаты игрока и вывести их в консоль (игрок не должен заранее знать свои координаты, они должны быть вычислены по ориентирам).

4) Разместить на поле дополнительного игрока из команды противника.

5) В процессе вычисления координат игрока (п. 3) одновременно вычислять координаты игрока противника (если он находится в поле зрения).

Ход работы

1) Начальное положение игрока

Для получения информации из командной строки использовался модуль `readline`. С помощью него запрашиваются:

- координаты игрока;
- период вращения;

- название команды;

Расположение игрока происходит с помощью отправки сообщения *move* со значением координат: `agent.socketSend("move", `${x} ${y}`).` Демонстрация на рис. 2.

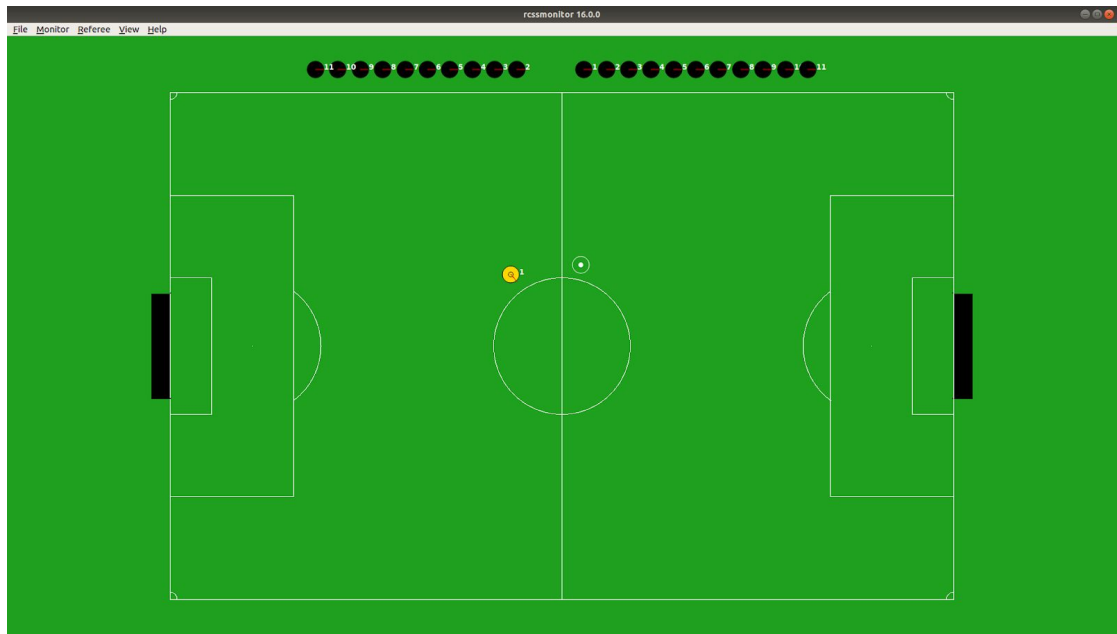


Рисунок 1 – Расположение на поле

2) Вращение после команды

Вращение выполняется с помощью сообщения *turn* со значением $36 / speed$, которое отправляется 10 раз в секунду.

Команда отправляется, если флаг *run* активен. Данный флаг становится активным после получения команды рефери *play_on*.

3) Определение своих координат

Из обработанного сообщения с сервера *see* выбираются все флаги, т.е. объекты, у которых первый элемент списка *cmd.p* начинается с *f* или *g*.

В случае отсутствия больше одного флага выводится сообщение «Нужно больше флагов».

В случае, если игрок видит только 2 флага, выполняется определение своего положения с помощью решения системы уравнений, представленной на рис. 3, с использованием теоремы Пифагора.

$$\begin{cases} d_1^2 = (x - x_1)^2 + (y - y_1)^2 \\ d_2^2 = (x - x_2)^2 + (y - y_2)^2 \end{cases}$$

Рисунок 2 – Система уравнений для двух флагов

Определение своего положения на поле выполняется с помощью решения системы линейных уравнений, представленной на рис. 4, если игрок видит 3 и более флагов, то среди них выбираются первые 3 такие, что любая из координат не повторяется более двух раз.

$$\begin{cases} d_1^2 = (x - x_1)^2 + (y - y_1)^2 \\ d_2^2 = (x - x_2)^2 + (y - y_2)^2 \\ d_3^2 = (x - x_3)^2 + (y - y_3)^2 \end{cases}$$

Рисунок 3 – Система уравнений для трех флагов

```
Нужно больше флагов
-5.99016662274277 37.58614116339385
-6.037058158526136 36.715294117647055
-5.6132343617561675 37.72
-5.516394226440077 36.59448128936643
-5.9305 37.07927884637046
Нужно больше флагов
-5.976000000000002 37.00565198623711
-5.99016662274277 37.58614116339385
-6.0868711174132395 36.800000000000004
-5.6132343617561675 37.72
```

Рисунок 4 – Вывод своих координат

4) Определение координат оппонента

Определение координат другого игрока осуществляется:

- с помощью информации о дистанции до видимых флагов и углов до них по теореме косинусов выполняется определение расстояний от флагов до другого игрока;
- с помощью полученных расстояний определяются координаты другого игрока аналогично определению своих координат, при этом текущий игрок также выступает в роли флага.



Рисунок 5 – Вывод координат обоих игроков

Выводы

В результате выполнения данной практической работы была решена задача вычисления позиции агента в пространстве при помощи информации из окружающей среды.

Для этого был создан агент, вращающийся вокруг заданной точки и по получаемой информации от сообщения see определяющий свои координаты и координаты другого агента, если он находится в поле зрения.