**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по практической работе №2**

**по дисциплине «Интеллектуальные системы»**

Тема: Движение игрока по маршруту

|  |  |  |
| --- | --- | --- |
| Студент гр. 9303 |  | Халилов Ш.А. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2023

**Цель работы.**

Решение задачи целенаправленного движения агента с учётом видимых ориентиров.

**Задание.**

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

1. В программе имитации задана последовательность действий игрока (например, [{act: "flag", fl: "frb"}, {act: "flag", fl: "gl"}, {act: "flag", fl: "fc"}, {act: "kick", fl: "b", goal: "gr"}]).
2. Программа должна выполнять заданную последовательность действий:

* При выполнении действия «движение к флагу» флаг считается достигнутым, если расстояние до флага меньше 3;
* Действие «удар по мячу» должно выполняться до забивания гола;
* После забивания гола последовательность действий (п. 1) должна повториться сначала.

**Ход выполнения.**

Для задавания последовательности действий создан класс ActionController.

ActionController - это класс, который позволяет управлять последовательностью действий агента. Класс содержит приватные свойства \_\_current и \_\_head, которые хранят текущее действие и первое действие соответственно.

В конструкторе класса не происходит никакой инициализации.

Метод ***chain*** добавляет новое действие в конец текущей последовательности. Если последовательность еще не существует, новое действие становится первым и текущим. Метод возвращает экземпляр класса, чтобы можно было вызвать методы цепочкой.

Метод ***current*** позволяет получить текущее действие.

Метод ***reset*** устанавливает текущее действие в начало последовательности. Метод возвращает экземпляр класса.

Метод ***next*** переключает текущее действие на следующее в последовательности. Метод возвращает экземпляр класса. Если последовательность закончилась, метод ничего не делает.

В данном примере мы создаем экземпляр класса ActionController для управления последовательностью действий агента. Мы определяем эту последовательность, вызывая метод chain на экземпляре ActionController. Каждый вызов метода chain добавляет новое действие в конец последовательности, как показано на рисунке 1.

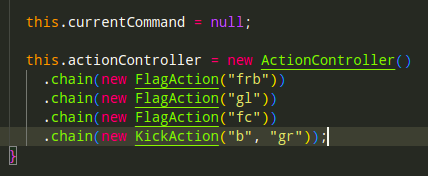


Рисунок 1 – экземпляр класса ActionController

В данном случае, создается последовательность действий, начиная с действия FlagAction с параметром "frb", затем добавляется действие FlagAction с параметром "gl", затем FlagAction с параметром "fc", и наконец KickAction с параметрами "b" и "gr". Таким образом, агент будет выполнять эти действия в указанном порядке.

Для выполнения последовательности действий были использованы следующие два алгоритма. Для поиска флага был использован следующий алгоритм:

1. Искомый флаг не виден, значит игроку следует повернуться в поисках искомого флага.
2. Искомый флаг виден и находится далеко от игрока, тогда необходимо повернуться и двигаться в направлении флага.
3. Искомый флаг виден и находится близко от игрока (расстояние меньше 3), тогда необходимо переходить к следующему действию в массиве действий.

Для забивания гола использован следующий алгоритм:

1. Мяч не виден, значит игроку следует повернуться в поисках мяча.
2. Мяч виден и находится далеко – движение к мячу по аналогии с движением к флагу, но целевое расстояние до мяча меньше – 0.5.
3. Мяч находится в зоне возможного удара, но ворота не видны, тогда необходимо с небольшой силой ударить мяч, чтобы он откатился вбок (на 30 градусов вправо).
4. Мяч находится в зоне возможного удара и ворота видны, тогда следует сильно ударить мяч в направлении ворот.

**Выводы.**

Была достигнута цель управления движением агента через использование визуальных ориентиров. Для этого был создан специальный объект, который сохраняет последовательность действий агента, а также были разработаны алгоритмы, которые позволяют агенту выполнять определенные действия, в зависимости от того, какие ориентиры он видит.