**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по практической работе №6**

**по дисциплине «Интеллектуальные системы»**

**Тема: «Командная игра»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8382 |  | Мирончик П.Д. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2022

# Цель работы

Целью работы является создание команда игроков и координация ее действий с учетом автономности каждого игрока.

Для достижения поставленной цели необходимо решить следующие задачи:

- создать программу игрока, обеспечивающую интерпретацию иерархического контроллера;

- создать контроллеры для всех игроков команды и вратаря;

- проверить командную игру одной команды против другой.

# Ход работы

Для реализации алгоритма поведения игроков будем использовать временной автомат из предыдущей работы с необходимыми модификациями для улучшения качества взаимодействия игроков.

Разобьем решение на подгруппы (контроллеры) по типам решаемых задач.

#### 1. Парсинг информации и представление ее первичных параметров.

С сервера приходят данные в формате, с которым было бы сложно работать напрямую. Для упрощения были реализованы классы, позволяющие прочитать и сформировать объекты с первичной информацией на основе «сырых» данных.

Hear – обрабатывает сообщения типа hear и содержит ряд вспомогательных методов, таких как isPlayOn, isKickOffAlly и canMove, на основании которых в дальнейшем может быть выбрана линия поведения для текущей ситуации.

Params – обрабатывает сообщения типа server\_param и собирает необходимые, заранее заданные, параметры – player\_size, ball\_decay и т.д.

Position – обрабатывает сообщения типа see. Задачи Position несколько сложнее задач Hear и Params: помимо простого создания объектов выполняется также расчет позиции игрока относительно флагов, а также расчет позиций объектов, для которых это представляется возможным.

Sense – собирает информацию из сообщений sense\_body. Помимо «сырых» также предоставляет возможность получить относительный вектор скорости игрока через метод relativeSpeed.

#### 2. Анализ первичной информации е ее сложение с данными из воспоминаний.

Контроллер второго уровня создан для решения задач тактического уровня – расчет всевозможных вспомогательных значений (скорость мяча, точка оптимального перехвата и т.д.).

BallAnalyzer – решает задачи, связанные непосредственно с мячом. Определяет, выполнялся ли удар по мячу в предыдущем такте, текущие координаты и скорость мяча, когда мяч был виден последний раз, можно ли ударить по мячу и какой силы и направления должен быть удар, чтобы направить мяч в желаемую точку.

PathAnalyzer – определяет оптимальные точки перехвата мяча в зависимости от скорости игрока.

PlayersAnalyzer – хранит список игроков, их позиции, принадлежность к команде, а также запоминает игроков на 20 тактов в глубину. В этот же класс включены методы, результат которых зависит от расположения других игроков: определение следующей оптимальной точки для ведения мяча и выбор оптимального союзника, которому можно отдать пас.

#### 3. Формирование действий игрока

В контроллере верхнего, третьего, уровня происходит принятие решений на основе данных, предоставляемых предыдущим контроллером. Определяется, стоит ли перехватывать мяч, стоит ли дать пас или же вести мяч самому, какая позиция сейчас лучшая с точки зрения расположения в схеме и нужно ли к ней приблизиться и т.д.

Этот контроллер представляет собой тот самый доработанный временной автомат из предыдущих работ.

Можно также подробнее рассказать про особенности поведения игроков. Группа нападения состоит из 6 человек: первая и вторая линии по 3 человека. Первая линия – основа нападения. Эти игроки почти не ограничены в действиях: они могут дать пас любому игроку своей команды, вести мяч хоть до самых ворот соперника, и именно они выполняют последний удар по воротам. Вторая линия – вспомогательные игроки нападения. Они следуют примерно в 25 метрах позади первой линии и имеют куда более строгие ограничения: ведение мяча допустимо, то только тогда, когда они отдалились от своей оптимальной позиции не более чем на 20 метров. Пас эти игроки могут давать лишь союзникам, которые находятся ближе к воротам соперника.

Линия защиты из четырех игроков по поведению имеет те же ограничения, что и вторая линия нападения, однако они не связаны с первой линией нападения и имеют свои фиксированные позиции.

Вратарь – особый игрок. Он всегда старается встать так, чтобы оказаться на линии между мячом и центров своих ворот. Он всегда побежит на перехват, если мяч находится слишком близко к воротам и, конечно, поймает мяч руками при первой возможности (если в порыве он не выбежал за пределы штрафной зоны).

# Выводы

В ходе выполнения практической работы была реализована программа управления агентами в составе полноценной команды из 11 человек. Проведены пробные матчи между командами, игроки в которых используют одинаковые алгоритмы, в результате которых каждая из команд забивала хотя бы один гол за игру.