**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по практической работе №4**

**по дисциплине «Интеллектуальные системы»**

**Тема: «Координация действий игроков»**

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

Санкт-Петербург

2022

# Цель работы

Целью работы является решение задачи координации действий игроков одной команды для обеспечения голевого паса.

Для достижения поставленной цели необходимо решить следующие задачи:

- создать программу игрока, обеспечивающего передачу паса;

- создать программу игрока, обеспечивающего забивание гола, после переданного паса, с учетом возможных погрешностей в передаче паса;

- обеспечить многократное повторение голевого паса в течение игры.

# Ход работы

В процессе выполнения работы было разработано дерево решений, позволяющее игрокам осуществлять взаимодействие и выполнять атаку на ворота противника с передачей голевого паса.

Для удобства чтения дерево разбито на отдельные модули – без этого размер подписей был бы слишком маленьким. Зеленым цветом отмечаются листья дерева – результаты выполнения, а также начало дерева (которое, формально, также является листом). Розовым цветом отображаются подблоки – отдельные модули дерева, структуру которых можно найти на других рисунках.

Дерево начинается с состояния init (рис. 1). В нем проверяется, известна ли позиция мяча, и когда в последний раз она обновлялась. Если обновление выполнялось достаточно давно, выполняется поиск мяча через ветвь find\_ball, иначе запускается ветвь schema.

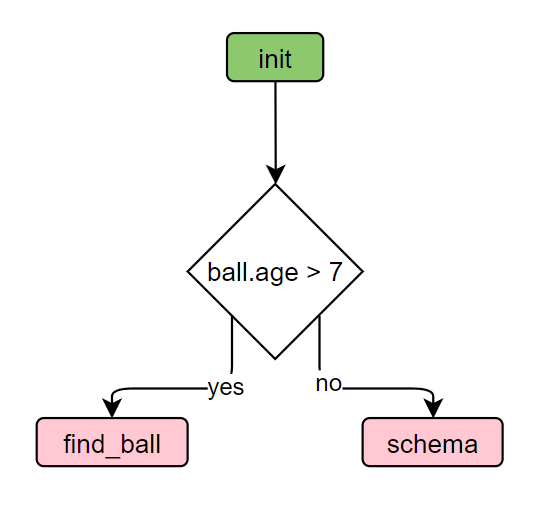


Рис. 1 – ветвь init

Ветвь find\_ball (рис. 2) осуществляет поворот агента на определенный угол с целью обнаружить мяч. Если известно ожидаемое направление, куда нужно повернуться (например, где в последний раз был обнаружен мяч), то поворот выполняется в этом направлении. В противном случае поворот выполняется на фиксированные 90 градусов.

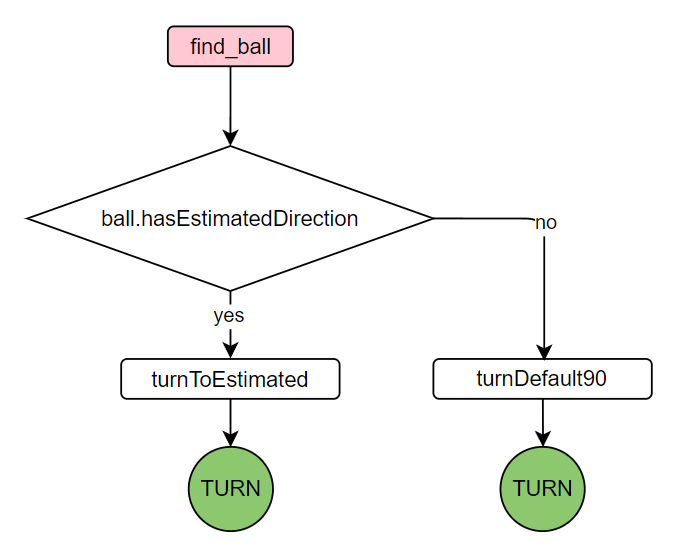
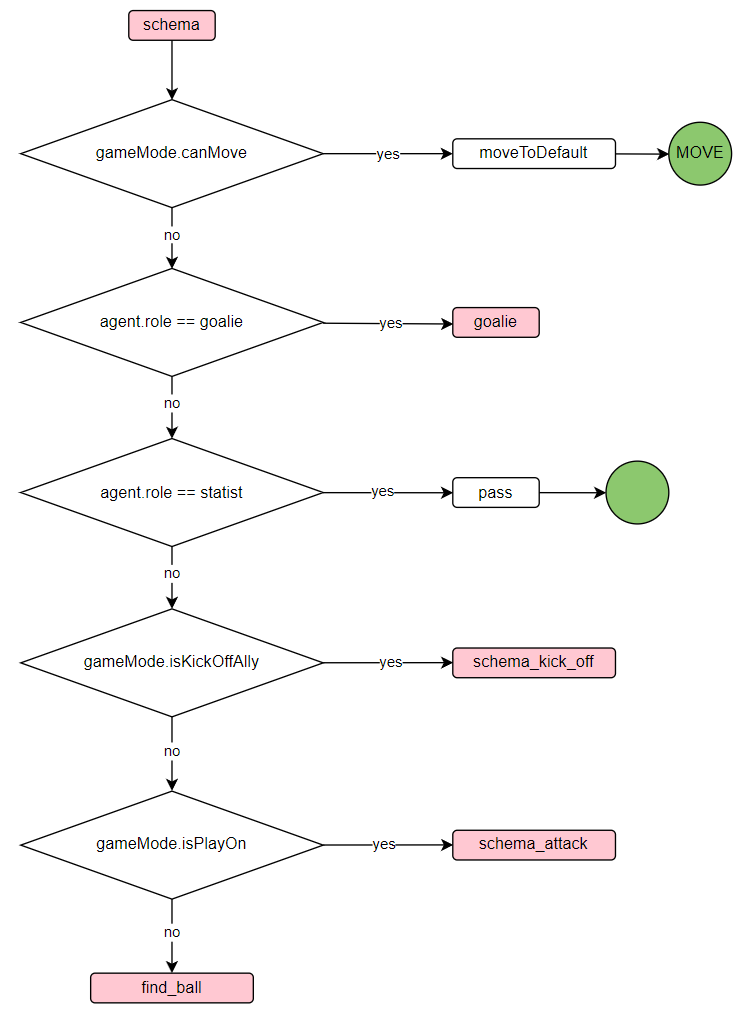


Рис. 2 – ветвь find\_ball

Ветвь schema отвечает за глобальное определение текущего алгоритма действий. Так, если в данный момент возможно перемещение игрока командой move, то выполняется перенос его в стартовую позицию (например, после забитого гола). После этого идет обработка по ролям. Вратарь имеет собственную ветвь действий goalie, т.к. его поведение в корне отличается от поведения остальных игроков. Статист просто стоит и ничего не делает – его задача не допустить оффсайд для нападающей команды. Если в данный момент необходимо разыграть мяч (т.е. один игрок должен подбежать к мячу и обязательно сделать передачу другому игроку), то вызывается ветвь schema\_kick\_off. Если игра находится в состоянии play\_on, то выполняется обычная ветвь задач атакующего игрока schema\_attack, а иначе – просто отслеживается мяч через find\_ball.

Рис. 3 – ветвь schema

Ветвь goalie определяет порядок действий вратаря. Если мяч в данный момент находится на дистанции не более 10 метров от ворот или вратарь может перехватить его на дистанции не более 15 метров, то выполняется перехват мяча ветвью goalie\_intercept. Если вратарь не находится на оптимальной позиции (на прямой между мячом и центром ворот), то выполняется движение к оптимальной точке через ветвь move, иначе – просто наблюдение за мячом через find\_ball.

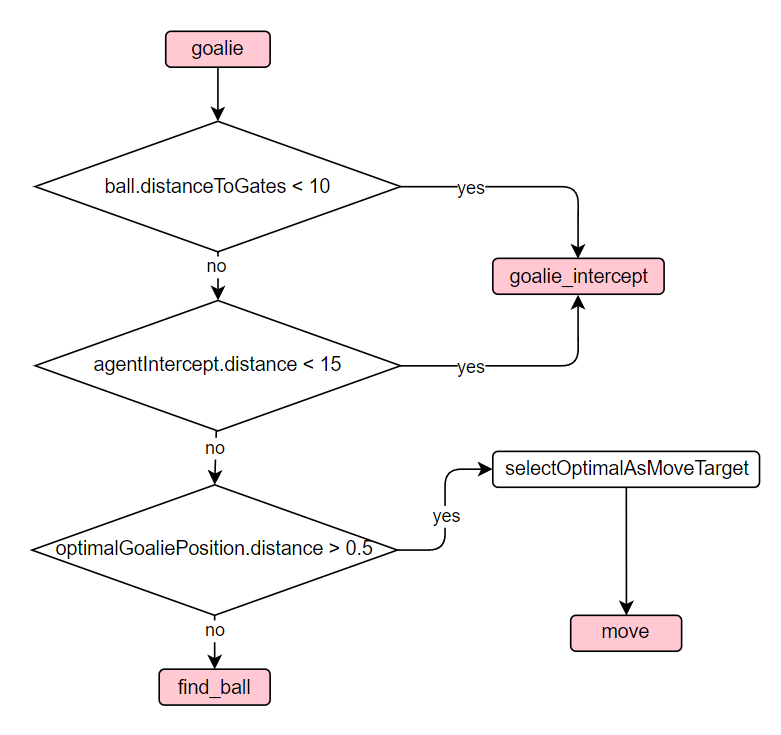


Рис. 4 – ветвь goalie

Ветвь schema\_kick\_off отвечает за порядок разыгровки мяча. Если ролью агента является центр атакующей линии, то агент бежит к мячу через ветвь move и, если мяч близко, выполняет передачу случайному союзнику через schema\_give\_pass. Все остальные игроки в этот момент просто осуществляют наблюдение за мячом через find\_ball.

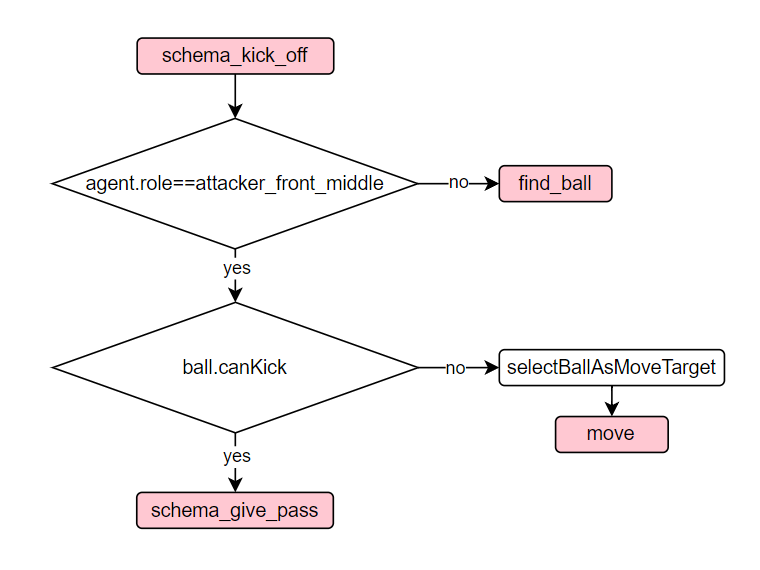


Рис. 5 – ветвь schema\_kick\_off

Ветвь schema\_attack определяет алгоритм действий атакующих игроков в режиме обычной игры. Если игрок может ударить по мячу, то управление передается ветви schema\_kick. Если игрок может добежать до мяча быстрее, чем любой из видимых союзников, то выполняется движение к оптимальной точке перехвата ветвью move. Если же в данный момент игроку нет необходимости взаимодействовать с мячом, то он должен занять свою определенную позицию в построении команды через ветвь move или, если он уже находится на позиции, просто следить за мячом через find\_ball.

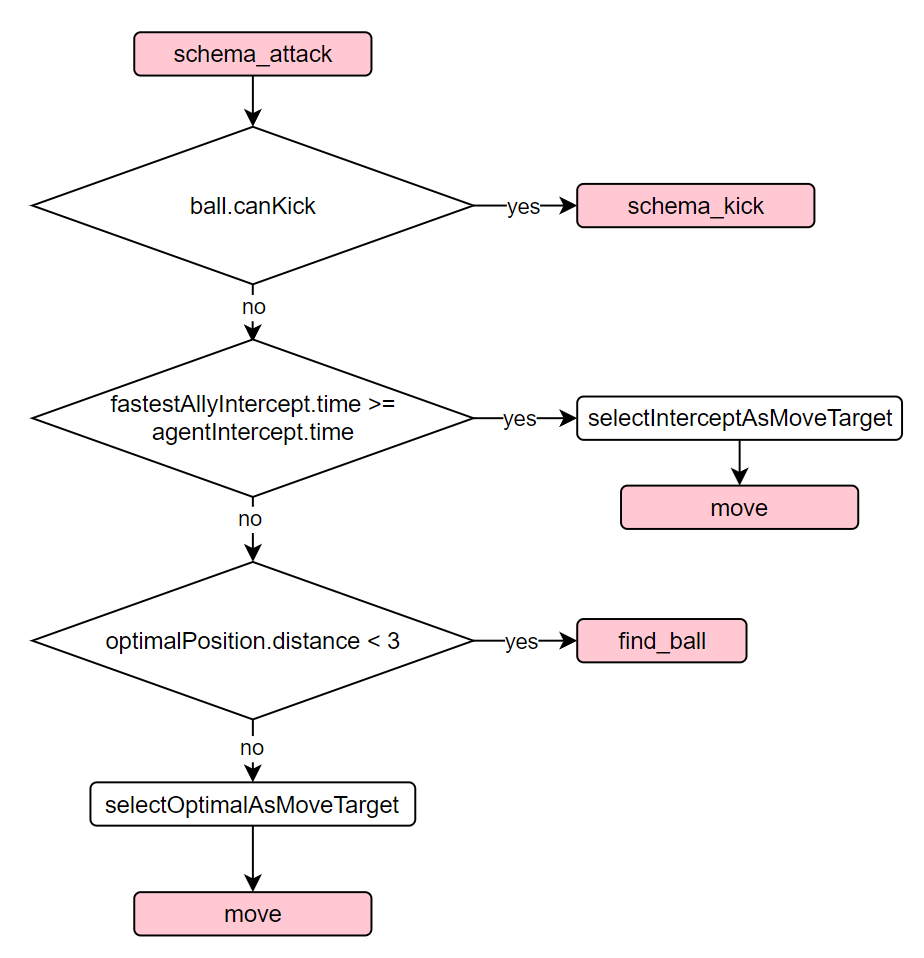


Рис. 6 – ветвь schema\_attack

Ветвь goalie\_intercept должна обеспечить перехват вратарем мяча в оптимальной точке и пнуть его подальше от ворот. Если вратарь может пнуть мяч, то выполняется передача со всей силы к центру поля. Если вратарь может поймать мяч, то выполняется команда catch, в противном случае через ветвь move вратарь движется в сторону оптимальной точки перехвата мяча.

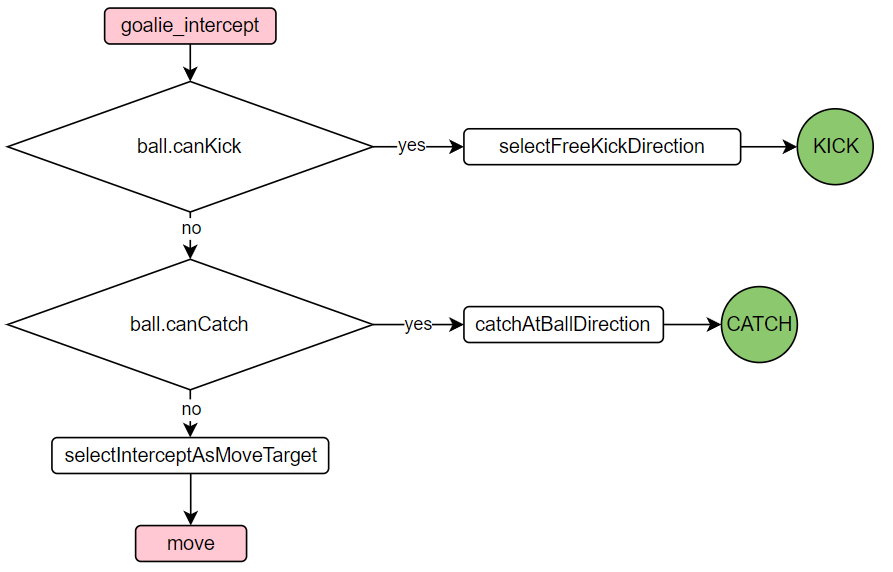


Рис. 7 – ветвь goalie\_intercept

Ветвь move определяет базовый процесс движения к определенной точке. Если направление взгляда игрока отходит от направления объекта не более чем на 6 градусов, то выполняется движение вперед через dash, иначе – выполняется поворот к точке командой turn.

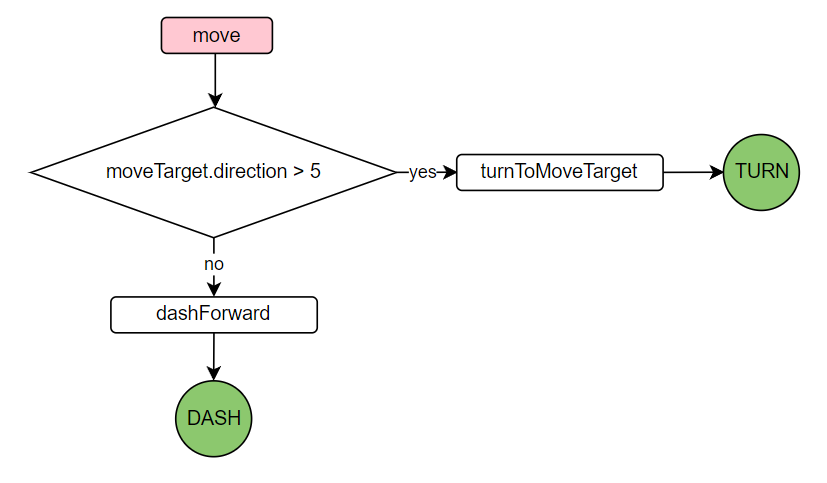


Рис. 8 – ветвь move

Ветвь schema\_give\_pass должна обеспечить передачу мяча, который находится на расстоянии удара, случайному союзнику. Для этого, разумеется, требуется видеть какого-либо союзника. Если в данный момент в поле зрения игрока не наблюдается союзных игроков, то агент выполняет поворот к ожидаемой позиции случайного игрока в схеме (напомним, что игроки имеют определенные позиции в ходе игры и стараются их придерживаться). Если же в поле зрения есть хотя бы один союзный игрок, то из видимых выбирается случайный и ему передается пас через ветвь kick\_ally.

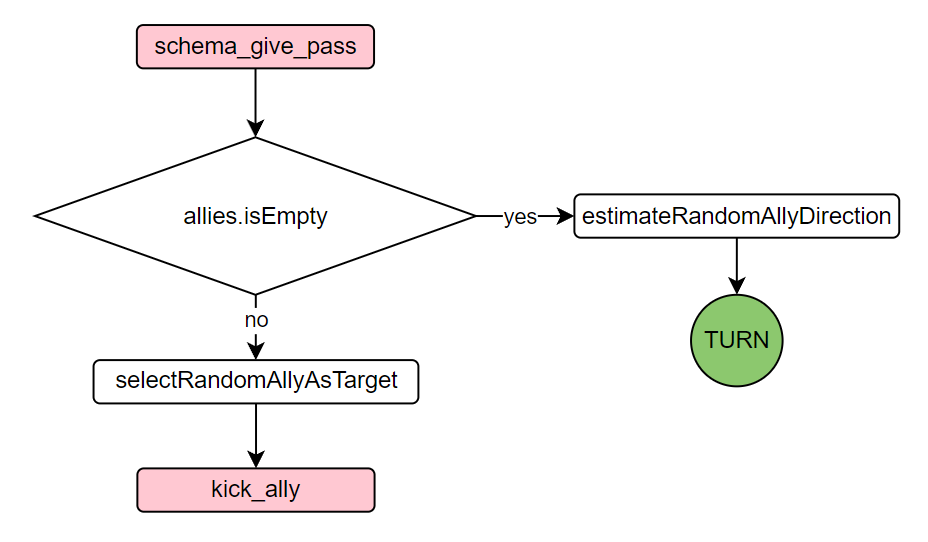


Рис. 9 – ветвь schema\_give\_pass

Задача ветви kick\_ally – дать передачу игроку, находящемуся на определенном расстоянии. Для этого предсказывается требуемая сила удара и выполняется удар по мячу с требуемой силой через kick.

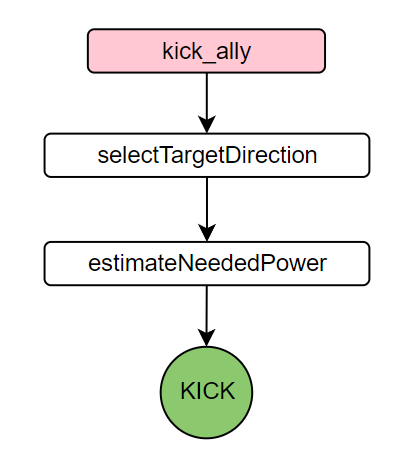


Рис. 10 – ветвь kick\_ally

schema\_kick определяет, что именно должен сделать с мячом атакующий игрок. Если мяч находится далеко от ворот, то выполняется слабый удар в сторону ворот – ведение мяча. Иначе, требуется определить, дать пас союзнику или ударить по воротам. Если последний раз игрок пинал мяч более 30 тактов назад, значит, он не выполнял ведение и мяч ему передали пасом; тогда выполняется удар по воротам. Иначе выполняется передача союзнику через schema\_give\_pass.

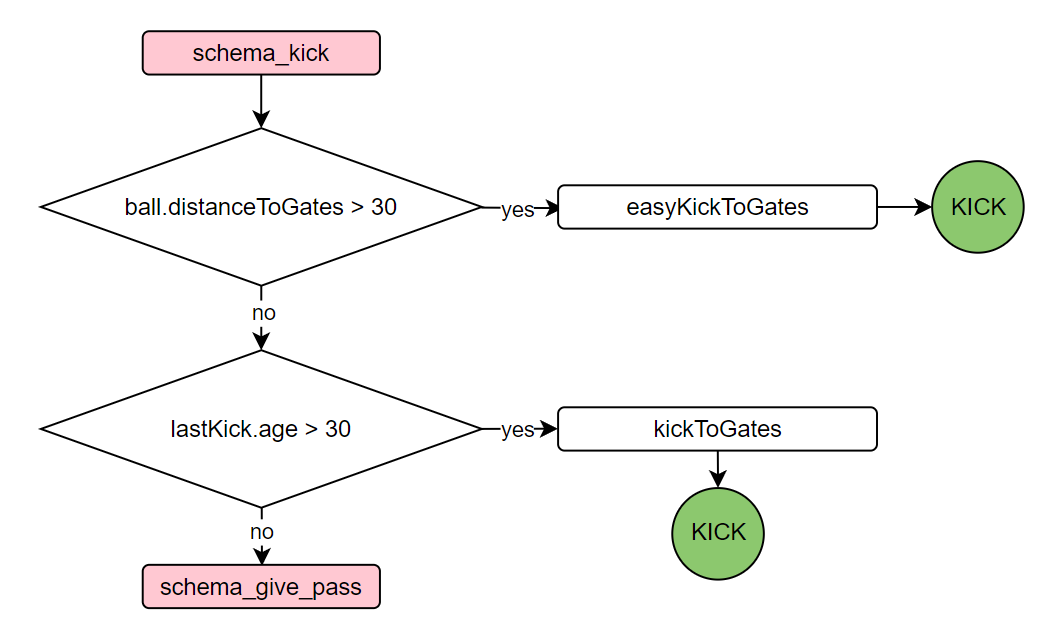


Рис. 11 – ветвь schema\_kick

В результате прохода по дереву образуются команды, которые позволяют игрокам осуществлять взаимодействие путем передачи голевого паса.

# Выводы

В ходе выполнения работы было реализовано дерево решений, предоставляющее набор действий на основе текущего состояния для игроков нападения и вратаря. Игроки выполняют ведение мяча и голевые передачи, вратарь же защищает ворота. В ходе игры игроками нападения осуществляется ряд успешных и неудачных атак, и действия вратаря оказываются достаточно эффективными для противостояния несложной логике действий атакующей команды.