**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ**

**ИМЕНИ Г. В. ПЛЕХАНОВА»**

Высшая школа кибертехнологий, математики и статистики

Кафедра цифровой экономики института развития информационного общества

«Допустить к защите»

Заведующий кафедрой

Цифровой экономики института развития информационного общества

Уринцов Аркадий Ильич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_ г.

**Выпускная квалификационная работа бакалавра**

Направление «Математическое обеспечение и администрирование информационных систем»

Профиль «Системное и интернет-программирование»

**Разработка алгоритма для управления группой объектов для преследования цели в играх**

Выполнил студент Федоров Кирилл Константинович

Группа 15.11Д-МО12/19б

Научный руководитель

выпускной квалификационной работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО, звание, степень, должность)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Автор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Москва – 2023

**Содержание**

Оглавление

[**Введение** 3](#_Toc130825568)

[**Глава I** 4](#_Toc130825569)

[**1.1 Введение в предметную область** 4](#_Toc130825570)

[Алгоритмы преследования 4](#_Toc130825571)

[Методы решения проблемы обхода полигонов 7](#_Toc130825572)

[**Список литературы** 12](#_Toc130825573)

# **Введение**

# **Глава I**

## **1.1 Введение в предметную область**

Алгоритмы преследования используются в разных областях жизни. Из чего следует, что существуют разные типы алгоритмов преследования с разными начальными данными и характеристиками объектов-преследователей. В данной главе будут продемонстрированы данные алгоритмы преследования и выбор оптимального для реализации заданной цели – разработка алгоритма, управляющего группой объектов для преследования цели.

### Алгоритмы преследования

#### Метод погони.

Этот метод ещё называют чистым преследованием, преследованием по кривой погони. Методом погони называется метод преследования, при котором вектор ПЕ\*Vпе в любой момент времени направлен на цель, то есть его курсовой угол α равен нулю. Проанализировав данный метод, можно сделать вывод, что при методе погони, преследователь приближается к цели сзади и кривая погони характеризуется большой кривизной, при любых начальных условиях. Таким образом, даже в условиях отсутствия каких-либо маневров со стороны цели, это приводит к низкой точности преследования, что является недостатком метода. Достоинством этого метода преследования является его помехоустойчивость – для реализации метода в каждый момент времени надо знать только, слева или справа от вектора преследования находится цель, для корректировки курсового угла соответствующим образом.

#### Метод постоянного угла упреждения.

Это метод, при котором курсовой угол в любой момент времени равен некоторой фиксированной величине α0. Величина α0 должна подчиняться условию (Vпе /Vпо)\*|sinα0| < 1, в противном случае преследователь начнет описывать вокруг цели бесконечную спираль, так её и не достигнув. Этот метод является модификацией метода погони, но у него есть достоинство в том, что при использовании угла упреждения кривая погони гораздо менее искривлена, чем для метода погони. Кроме этого, метод обладает похожей помехоустойчивостью, что и метод погони. Однако для реализации этого метода необходима информация о пеленге цели, а также о направлении движения цели для выбора правильного угла упреждения.

#### Метод параллельного сближения.

Параллельным сближением называется вид преследования, когда линия визирования всегда смещается параллельно самой себе. Если цель движется прямолинейно и равномерно, то траектория преследователя есть прямая. При маневрах цели, когда цель получает ускорение, для сохранения условия параллельности линии визирования, ускорение преследователя будет совпадать с нормальным ускорением цели. Это является достоинством данного метода. К недостаткам метода можно отнести большое количество требуемой информации: курсовой угол цели, скорость.

#### Метод пропорционального наведения.

У него нет тех недостатков, которые есть у метода параллельного сближения. Это такой метод, при котором угловая скорость преследователя пропорциональна угловой скорости линии визирования. Назначение этого метода – поражение цели, с учетом тенденции поворота линии визирования. Как следует из принципов действия этого метода, для его реализации необходима лишь информация о пеленге цели.

#### Оптимальные алгоритмы преследования.

Оптимальными называются алгоритмы, который дают возможность получить оптимальное решение задачи управления с четко определенным функционалом (целевой функцией). Решение задач оптимального управления усложняется, при усложнении движения целевого объекта. Когда цель движется равномерно и прямолинейно, возможно аналитическое решение. Что невозможно в случае, когда цель маневрирует, используя сложные пространственные траектории. Подобные задачи называются стохастическими задачами преследования. Использование теории оптимального управления в решении задач наведения является важной составляющей, часто использующейся на практике, например, в частном случае преследования цели, движущейся под острым углом к встречному курсу. В таком случае классические алгоритмы погони и постоянного угла упреждения не будут работать. Теория оптимального управления позволяет получить алгоритмы преследования, которые могут решать и задачу преследования быстродвижущихся целей. Но недостатком данных алгоритмов является большая сложность в реализации при сложных траекториях искомого объекта.

#### Дифференциальные игры преследования.

Одной из классических дифференциальных игр является задача преследования – уклонения, когда преследователь пытается минимизировать (а цель максимизировать) время, за которое преследователь настигнет цель.

Опираясь на [источник](#СЛ_1), оптимальной стратегией является метод погони. При разных игровых моделях, а также различных управлениях, оптимальными стратегиями могут стать стратегии, реализующиеся аналогами классических алгоритмов со всеми их достоинствами и недостатками.

Учитывая то, что перед данной дипломной работой не стоит задача реализации сложных алгоритмов преследования, которые применяются в военно-промышленной области, например, учитывающие физические нюансы, классический алгоритм преследования, или алгоритм погони, наиболее подходит для поставленной цели. Однако, использование обычного метода погони без каких-либо изменений не вполне рационально, потому что предполагается, что на местности, на которой будет работать данный алгоритм, будут располагаться препятствия в виде полигонов разной величины. Опираясь на [источник](#СЛ_2), использование обычного метода погони, приведет к тому, что объекты-преследователи не будут способны обходить препятствия и просто будут застревать в них, пытаясь пройти через них. К тому же, задача дипломной работы заключается в реализации алгоритма, управляющего группой объектов. Из [статьи](#СЛ_3) следует, что в играх в основном используется комбинированная модель поведения объектов – они управляются единым интеллектом, однако конкретные функции, например, преследование, реализуется на конкретном объекте-преследователе. По сути, единый интеллект распоряжается лишь состояниями подконтрольными объектами.

Для решения задачи обхода полигонов существует множество методов, имеющие принципиально разные концепции и разные входные данные.

### Методы решения проблемы обхода полигонов

#### Метод с предварительной оценкой местности.

[Статья](#СЛ_4) предлагает метод для решения проблемы обхода полигонов, основывающийся на предварительной оценке карты местности и последующей корректировки знаний. Карта, в свою очередь, представляется в виде двумерного массива, состоящая из единиц и нулей. Каждая ячейка этого массива представляет собой отдельную клетку, где 0 – клетка проходима, а 1 – не проходима. Так же, данный метод имеет два случая – NPC (Non-Player Character) «знают» местоположение игрока и обратный случай.

В первом, когда NPC «знаю» местоположение игрока, ситуация в понятиях игрового искусственного интеллекта называется cheat-методом. Игрок ставится в неравное положение по отношению к неигровым персонажам, и тем самым повышается сложность игры. Проведенный предварительный анализ проблемы показал, что проблема преследования обычно решается двумя основными способами:

— использование алгоритмов поиска пути;

— комбинация алгоритмов поиска пути и алгоритмов преследования.

С точки зрения практической реализации, первый способ является самым простым. При изменении местоположения игрока пути до него просчитываются заново. Подобный подход имеет серьезные недостатки, связанные с большими временными затратами, а также с большим потреблением памяти.

Если в качестве алгоритма поиска пути выбран алгоритм A\*, то временные затраты, в худшем случае, растут экспоненциально, аналогично экспоненциально растут затраты памяти.

Второй способ более сложный. Путь до игрока вычисляется с помощью алгоритмов поиска пути. Если расстояние между игроком и неигровым персонажем становится меньше R, то используется более простой и менее ресурсоемкий алгоритм преследования. Эффективность такого метода определяется, прежде всего, величиной радиуса видимости R и размером карты.

NPC «не знаю» о местоположении игрока. Методы из данной группы добавляют в игру больший реализм. Данные методы требуют, чтобы неигровые персонажи были в состоянии предсказывать местоположение игрока. В простейшем случае они могут просто двигаться в случайных направлениях. Как только игрок попадает в область видимости неигрового персонажа, дальнейшее преследование осуществляется при помощи соответствующих алгоритмов. Таким образом, первоочередной задачей становится задача оценки вероятности появления игрока в той или иной ячейке карты.

К разработанному алгоритму на этапе проектирования были выдвинуты следующие требования:

— результатом его работы должна быть матрица весов, каждый элемент которой–вес конкретной ячейки игровой карты;

— алгоритм должен обеспечивать учет предыдущих игровых исходов, т.е. быть самообучающимся;

— первоначальный расчет должен проводиться с учетом положения входа и выхода и структуры карты. Таким образом, в функционировании алгоритма можно выделить три этапа:

— этап инициализации;

На данном этапе проводится инициализация матрицы весов

— этап обучения;

Вычисления на данном этапе происходят в конце каждой игровой сессии.

— этап корректировок.

В ходе работы алгоритма возникают ситуации, когда необходимо нормировать значения в матрице весов.

Описанный метод был реализован в компьютерной игре Paclight. На произвольных картах размерности 40 на 40 метод показал свою пригодность. Проведенные наблюдения показали, что на картах такого размера время, затраченное на расчет путей, было меньше примерно в1,2разапосравнению с применение только А\* (в случаях, когда NPC знают местоположение игрока). Ожидается, что с увеличением размерности карт выгода от применения метода предварительной оценки карт возрастет.

#### [Хранение информации о местности.](#СЛ_2)

Наиболее простой способ обойти проблему столкновения объектов-преследователей с полигонами состоит в том, чтобы сохранить некоторую "информацию об окружении" (ИО) внутри каждого объекта. Это означает, что если объект утыкается в твердый объект, то он мог бы "спросить" этот объект, куда двигаться, чтобы обойти его.

В зависимости от стороны, с которой произошло столкновение, каждое препятствие возвращает значения пары смещения dx/dy относительно объекта. Правила, для нахождения этих значений:  
 — Допустим, объект свободно перемещается по экрану, к примеру, из левой верхней к правой нижней его части. И он соприкоснулся с левой стороной препятствия, тогда пускай он движется вниз.  
— Если в препятствие ударились по направлению какой-нибудь оси, возвратим противоположное направление по этой же оси. Если препятствие имеет ИО (-1, 1) и получает удар от чего-то движущегося по оси X, следует возвратить 1.

Очевидный недостаток этого приема в том, чтобы сохранить другие два значения для каждого объекта. Возможным решением данной проблемы станет сохранение данной информации, используя всего лишь 4 бита.

#### [Движение объекта-преследователя по выпускаемым лучам.](#СЛ_5)

Идея данного алгоритма в выпускании от каждого бота преследователя 2 отрезка сенсора, направленные на игрока. При пересечении полигона лучом, данный луч должен поворачиваться (один поворачивается по часовой стрелке, другой против) до тех пор, пока они не перестанут пересекать полигоны и далее объект-преследователь движется по тому лучу, который быстрее добрался до преследуемой цели и раньше перестал пересекать полигоны.

Однако, у этой идеи есть свои недостатки. Во-первых, она достаточно трудоемкая в силу того, что на каждой итерации надо проверять от каждого объекта-преследователя лучи пересекаются ли они с каждым ребром полигона, которых потенциально может быть много на местности. Во-вторых, данный алгоритм не всегда приводит к ожидаемым результатам. Если длина луча слишком коротка, то бот может зайти в тупик, из которого придется выбираться.

Например:



Изображение №1.

На [изображении выше](#К_1), бот пойдет оранжевым путем, когда красный явно оптимальнее.

Однако, если длина луча слишком длинная, то бот, наоборот, примет узкий проходит за тупит и будет обходить его, как на [изображении ниже](#К_2).



Изображение №2

Серый вектор – луч, пересекший полигоны. Здесь бот пойдет по оранжевому пути, хотя красный оптимальнее.

Частично эти проблемы нивелируются, если игрока, объектов-преследователей и вершины полигонов представить в виде вершин графа и совершать поиск наикратчайшего пути, например при помощи алгоритма Дейкстры.

Однако, проблема оптимизации остается прежней. Чем больше карта, полигонов и объектов-преследователей, тем медленнее работает алгоритм, потому что для каждой итерации придется пересчитывать алгоритм Дейкстры на большое дерево и для каждого объекта преследователя.

#### Метод ищейки

# **Список литературы**

1. <https://cyberleninka.ru/article/n/sravnenie-algoritmov-presledovaniya-obektov/viewer>

1. <https://programcpp.ucoz.ru/publ/4-1-0-6>

1. <https://intuit.ru/studies/courses/1104/251/lecture/6456>

1. <https://cyberleninka.ru/article/n/problema-presledovaniya-v-igrah-snizhenie-resursoemkosti-resheniya-s-pomoschyu-metoda-s-predvaritelnoy-otsenkoy-kart/viewer>

1. <https://qna.habr.com/q/870001>
2. <https://habr.com/ru/post/496878/>