

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ

по дисциплине

«ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ИГРОВЫХ ПРИЛОЖЕНИЯХ»

Направление подготовки 09.03.02 Информационные системы и технологии

Направленность (профиль) Прикладное программирование и интернет-технологии

СОДЕРЖАНИЕ

Лабораторная работа №1. Введение в игровой ИИ и базовое движение

Лабораторная работа №2. Рулевое управление (Steering Behaviors)

Лабораторная работа №3. Групповое поведение и флотинг

Лабораторная работа №4. Поиск пути: Алгоритм Дейкстры

Лабораторная работа №5. Алгоритм A* (A-star) и эвристики

Лабораторная работа №6. Конечные автоматы (FSM)

Лабораторная работа №7. Деревья поведения (Behavior Trees)

Лабораторная работа №8. Минимакс и состязательные игры

Лабораторная работа №9. Введение в обучение с подкреплением (RL)

ЛАБОРАТОРНАЯ РАБОТА №1. ВВЕДЕНИЕ В ИГРОВую ИИ И БАЗОВОЕ ДВИЖЕНИЕ

Цели и задачи

Цель лабораторной работы: Изучение структуры игрового цикла, работы с векторами и реализация базового перемещения агента.

Основные задачи:

- изучить структуру игрового цикла: обработка событий, обновление состояния, рендеринг;
- освоить работу с векторами для описания движения точки в 2D-пространстве;
- научиться ограничивать перемещение агента рамками игрового окна (Screen Wrapping).

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla

firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork1.md>

Порядок выполнения работы:

1. Создайте основной файл проекта, инициализируйте Pygame и настройте окно 800x600.
2. Используйте класс `pygame.math.Vector2` для хранения позиции и скорости.
3. Обновление логики выносите в отдельный метод `update()`, отрисовку — в `draw()`.
4. Фиксируйте FPS с помощью `pygame.time.Clock` для стабильности симуляции.

Задание 1. Равномерное движение.

Создание агента, который движется в заданном направлении с постоянной скоростью.

- Реализовать класс `Agent` с атрибутами `position` и `velocity`.
- Отрисовать агента в виде простого треугольника, направленного по вектору скорости.
- Добавить проверку выхода за границы экрана для циклического появления с противоположной стороны.

Задание 2. Управление и инерция.

Добавление пользовательского ввода для изменения вектора движения.

- Привязать изменение вектора скорости к нажатию клавиш (WASD/Arrows).
- Реализовать эффект плавного ускорения и торможения (friction/трение).
- Вывести текущие координаты и модуль скорости текстом на экран.

Задание 3. Следование за целью.

Простейший ии, реагирующий на положение курсора.

- Вычислить вектор направления от агента к координатам `pygame.mouse.get_pos()`.
- Заставить агента постоянно поворачиваться в сторону курсора.
- Реализовать 'прыжок' агента в точку клика мыши.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. За что отвечает метод `pygame.display.flip()` и почему он важен?

2. В чем преимущество использования `Vector2` перед обычными кортежами (x, y) ?
3. Как рассчитать угол поворота спрайта, имея вектор его скорости?

ЛАБОРАТОРНАЯ РАБОТА №2. РУЛЕВОЕ УПРАВЛЕНИЕ (STEERING BEHAVIORS)

Цели и задачи

Цель лабораторной работы: Освоение классических алгоритмов рулевого управления Крейга Рейнольдса для создания естественного поведения.

Основные задачи:

- изучить формулу $\text{Steering} = \text{Desired_Velocity} - \text{Current_Velocity}$;
- реализовать ограничение векторов через метод `scale_to_length` или `clamp`;
- научиться визуализировать векторы сил с помощью `pygame.draw.line`.

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork2.md>

Порядок выполнения работы:

1. Реализуйте расчет силы поворота (Steering Force) как разность между желаемой и текущей скоростями.
2. Ограничьте максимальную силу и максимальную скорость агента.
3. Визуализируйте векторы сил цветными линиями для отладки поведения.

Задание 1. Реализация алгоритма Seek.

Создание поведения активного преследования движущейся цели.

- Написать функцию seek(), возвращающую силу в сторону цели.
- Реализовать плавное изменение траектории агента при резком смещении цели.
- Добавить визуализацию 'желаемой скорости' зеленым вектором.

Задание 2. Реализация алгоритма Arrive.

Создание интеллектуального торможения при достижении точки назначения.

- Ввести параметр 'радиус замедления' (slowing radius).
- Рассчитать линейное снижение скорости внутри этого радиуса.
- Добиться полной остановки агента ровно в центре целевой точки без осцилляции.

Задание 3. Сложные маневры.

Комбинирование сил для создания поведения 'уклонения'.

- Реализовать алгоритм Flee (бегство), работающий обратно Seek.
- Создать сценарий: агент убегает от курсора, если тот находится ближе 200 пикселей.
- Добавить плавное переключение между преследованием и бегством.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Почему нельзя просто присваивать координатам агента координаты цели?
2. Что произойдет с агентом Seek, если его максимальная сила поворота будет очень мала?
3. Каков математический смысл 'желаемой скорости' в алгоритме Arrive?

ЛАБОРАТОРНАЯ РАБОТА №3. ГРУППОВОЕ ПОВЕДЕНИЕ И ФЛОКИНГ

Цели и задачи

Цель лабораторной работы: Изучение принципов эмерджентного поведения через реализацию правил флокинга.

Основные задачи:

- освоить расчет расстояний между множеством объектов в реальном времени;
- изучить влияние весовых коэффициентов на общее поведение группы;
- реализовать базовую оптимизацию: проверка соседей только в ограниченном радиусе видимости.

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork3.md>

Порядок выполнения работы:

1. Создайте массив из 30-50 агентов.
2. Для каждого агента в цикле находите соседей в определенном радиусе.
3. Вычисляйте три вектора сил (Separation, Alignment, Cohesion) и суммируйте их с определенными весами.

Задание 1. Три правила стаи.

Программирование индивидуальных векторов для каждого правила флоккинга.

- Реализовать Separation: вектор отталкивания от слишком близких соседей.
- Реализовать Alignment: вычисление среднего вектора скорости группы и подстройка под него.
- Реализовать Cohesion: стремление к среднему арифметическому позиций соседей.

Задание 2. Настройка баланса.

Экспериментальный подбор параметров для имитации различных видов существ.

- Реализовать ползунки или клавиши для изменения весов сил в реальном времени.
- Настроить стаю так, чтобы она вела себя как косяк рыб (высокий

Alignment).

– Настроить стаю так, чтобы она вела себя как рой насекомых (высокий Cohesion, низкий Alignment).

Задание 3. Взаимодействие с хищником.

Добавление внешних факторов, влияющих на целостность группы.

- Создать крупного агента-'хищника', управляемого мышью.
- Добавить всем бойдам силу Flee по отношению к хищнику.
- Пронаблюдать эффект распада и последующего воссоединения стаи.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Какая из трех сил отвечает за сохранение структуры группы при маневрах?
2. Почему расчет флокинга имеет вычислительную сложность $O(N^2)$?
3. Как влияет радиус видимости (neighborhood radius) на размер формируемых групп?

ЛАБОРАТОРНАЯ РАБОТА №4. ПОИСК ПУТИ: АЛГОРИТМ ДЕЙКСТРЫ

Цели и задачи

Цель лабораторной работы: Реализация базового алгоритма поиска пути на дискретной сетке в среде Pygame.

Основные задачи:

- изучить способы представления игрового поля в виде графа смежности;
- реализовать генерацию препятствий кликом мыши;
- научиться восстанавливать путь от цели к старту через словарь 'родителей'.

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork4.md>

Порядок выполнения работы:

1. Создайте класс Grid для хранения двумерного массива ячеек (0 — свободно, 1 — стена).
2. Реализуйте визуализацию сетки.
3. Используйте очередь (collections.deque) для хранения фронта обхода.
4. Помечайте посещенные ячейки цветом.

Задание 1. Построение сетки.

Визуальное отображение структуры данных графа.

- Отрисовать сетку ячеек (например, 20x20 пикселей каждая).
- Реализовать переключение состояния ячейки (проходима/стена) по клику.
- Добавить возможность установки точек 'Старт' и 'Финиш'.

Задание 2. Алгоритм Дейкстры.

Поиск кратчайшего пути с учетом веса ребер.

- Реализовать обход графа, начиная от стартовой ячейки.
- Добавить визуализацию процесса заполнения (анимация 'волны').
- Отрисовать найденный путь жирной линией.

Задание 3. Разные типы поверхности.

Усложнение графа за счет введения стоимости перемещения.

- Добавить ячейки 'болото' (высокая стоимость) и 'дорога' (низкая стоимость).
- Модифицировать алгоритм для учета веса ячеек.
- Проверить, как путь огибает 'болото' даже при наличии свободного пространства.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Чем отличается алгоритм Дейкстры от поиска в ширину (BFS)?
2. Как хранится информация о том, из какой ячейки мы пришли в текущую?
3. Какова сложность алгоритма Дейкстры на сетке размером $N \times M$?

ЛАБОРАТОРНАЯ РАБОТА №5. АЛГОРИТМ А* (A-STAR) И ЭВРИСТИКИ

Цели и задачи

Цель лабораторной работы: Реализация эвристического поиска пути для повышения производительности ИИ.

Основные задачи:

- изучить влияние эвристики на направление поиска;
- освоить работу с очередью с приоритетом в Python;
- научиться оптимизировать расчеты для работы в реальном времени.

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением

учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork5.md>

Порядок выполнения работы:

1. Модифицируйте алгоритм Дейкстры, добавив приоритетную очередь (heapq).
2. Внедрите функцию эвристики (Манхэттенское или Евклидово расстояние).
3. Сравнивайте количество посещенных узлов в Дейкстре и A*.

Задание 1. Интеграция эвристики.

Превращение алгоритма дейкстры в a путем добавления оценки расстояния.*

- Написать функцию `heuristic(a, b)` для расчета Манхэттенского расстояния.
- Модифицировать расчет приоритета ячейки: $f = g + h$.
- Сравнить визуально, как сужается область поиска по сравнению с прошлой работой.

Задание 2. Динамический поиск.

Обновление пути агента при изменении условий среды.

- Реализовать агента, который движется по найденному пути.
- Сделать так, чтобы при добавлении стены во время движения, агент перестраивал путь.
- Оптимизировать частоту пересчета пути (не каждый кадр, а по событию).

Задание 3. Диагональное движение.

Расширение графа связей до 8 направлений.

- Разрешить перемещение по диагонали.
- Изменить эвристику на Евклидово расстояние или расстояние Чебышёва.
- Учесть, что стоимость диагонального шага должна быть равна корню из 2 (~ 1.41).

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Почему A^* обычно быстрее алгоритма Дейкстры?
2. Что произойдет, если эвристика будет возвращать случайное число?
3. В каком случае A^* найдет не самый короткий путь?

ЛАБОРАТОРНАЯ РАБОТА №6. КОНЕЧНЫЕ АВТОМАТЫ (FSM)

Цели и задачи

Цель лабораторной работы: Создание структурированного поведения агента через управление состояниями.

Основные задачи:

- изучить концепцию состояний и переходов (Transitions);
- научиться разделять логику поведения на независимые модули;
- реализовать базовый ИИ охранника.

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork6.md>

Порядок выполнения работы:

1. Реализуйте базовый класс State с методами enter, execute, exit.
2. Создайте контроллер (StateMachine), который хранит текущее состояние.
3. Визуализируйте текущее состояние текстом над головой агента.

Задание 1. Патруль и преследование.

Создание двух базовых состояний и условий перехода между ними.

- Состояние Patrol: агент ходит между заданными точками по кругу.
- Состояние Chase: если игрок (курсор) ближе 150px, агент переходит к преследованию.
- Реализовать возврат к патрулированию, если игрок скрылся.

Задание 2. Состояние отдыха и ресурсы.

Добавление внутренних параметров агента (энергия, голод).

- Ввести параметр 'заряд' (energy), который тратится при движении.
- Состояние Recharge: когда энергия на нуле, агент идет к 'базе' и стоит там 3 секунды.
- Реализовать визуальную шкалу энергии над агентом.

Задание 3. Стековый автомат.

Реализация памяти состояний для более сложной логики.

- Использовать список (стек) для хранения состояний.
- Реализовать состояние 'Испуг': при клике на агента он убегает 2

секунды, а затем возвращается к тому, что делал раньше.

– Проверить корректность возврата из вложенных состояний.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Зачем нужны методы `enter` и `exit` в классах состояний?
2. Как избежать бесконечного переключения между двумя состояниями в один кадр?
3. В чем отличие FSM от простого набора условий `if-else`?

ЛАБОРАТОРНАЯ РАБОТА №7. ДЕРЕВЬЯ ПОВЕДЕНИЯ (BEHAVIOR TREES)

Цели и задачи

Цель лабораторной работы: Построение гибкой и расширяемой архитектуры ИИ для сложных задач.

Основные задачи:

- изучить иерархическую структуру принятия решений;
- освоить принцип работы селекторов и последовательностей;
- научиться создавать повторно используемые узлы действий.

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением

учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork7.md>

Порядок выполнения работы:

1. Реализуйте базовые классы узлов: Composite (Selector, Sequence), Leaf (Action, Condition).
2. Каждый узел должен возвращать статус: SUCCESS, FAILURE или RUNNING.
3. Соберите дерево для агента-собирателя.

Задание 1. Базовые узлы.

Создание минимального набора классов для работы дерева.

- Реализовать узел Sequence (истина, если все истины).
- Реализовать узел Selector (истина, если хотя бы одна истина).
- Написать действие 'MoveToTarget' и условие 'IsTargetVisible'.

Задание 2. ИИ лесоруба.

Реализация комплексного поведения прс по добыче ресурсов.

- Создать дерево: ЕСЛИ (инвентарь полон) -> ИДТИ (склад) -> ВЫГРУЗИТЬ; ИНАЧЕ -> ИДТИ (дерево) -> РУБИТЬ.
- Визуализировать текущую ветку дерева, которая выполняется в данный момент.
- Добавить в дерево прерывание, если на пути появляется препятствие.

Задание 3. Декораторы и циклы.

Добавление логических модификаторов в дерево.

- Реализовать декоратор Inverter (инверсия результата).
- Реализовать декоратор Repeater (повторение действия N раз).
- Создать поведение 'патруль' через узел-последовательность и список точек.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Какое состояние узла (SUCCESS/FAILURE/RUNNING) позволяет реализовывать длительные действия?
2. Чем Behavior Trees удобнее для геймдизайнера, чем FSM?
3. Как работает узел Selector, если его первый потомок всегда возвращает SUCCESS?

ЛАБОРАТОРНАЯ РАБОТА №8. МИНИМАКС И СОСТЯЗАТЕЛЬНЫЕ ИГРЫ

Цели и задачи

Цель лабораторной работы: Создание непобедимого ИИ для настольных игр с использованием поиска по дереву состояний.

Основные задачи:

- изучить теорию игр с нулевой суммой;
- освоить рекурсивный обход дерева состояний;
- реализовать функцию оценки терминальных состояний (Win/Loss/Draw).

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork8.md>

Порядок выполнения работы:

1. Реализуйте игру 'Крестики-нолики' (3x3) на Pygame.
2. Напишите рекурсивную функцию `minimax`, которая оценивает все возможные ходы.
3. Добавьте визуализацию 'мыслей' ИИ (оценку каждой клетки).

Задание 1. Механика игры.

Создание графической оболочки для пошаговой игры.

- Отрисовать игровое поле и реализовать логику хода человека по клику.
- Написать функцию проверки условий победы.
- Добавить кнопку сброса игры.

Задание 2. Мозг ИИ (Minimax).

Реализация логики выбора оптимального хода.

- Написать рекурсивную функцию поиска лучшего значения для текущего игрока.
- Интегрировать ИИ в игровой цикл: после хода человека ИИ делает свой ход.
- Убедиться, что против ИИ невозможно выиграть (максимум — ничья).

Задание 3. Альфа-бета отсечение.

Оптимизация перебора для ускорения работы алгоритма.

- Добавить параметры α и β в функцию `minimax`.
- Реализовать пропуск ветвей дерева, которые не влияют на результат.
- Замерить количество вызванных рекурсий с отсечением и без него (вывод в консоль).

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. Что такое функция оценки (`evaluation function`) и зачем она нужна в сложных играх?
2. Почему Минимакс редко используется в играх реального времени (например, RTS)?
3. Как глубина рекурсии влияет на сложность и 'ум' алгоритма?

ЛАБОРАТОРНАЯ РАБОТА №9. ВВЕДЕНИЕ В ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ (RL)

Цели и задачи

Цель лабораторной работы: Знакомство с концепцией обучения с подкреплением на примере задачи навигации.

Основные задачи:

- изучить компоненты RL: Agent, Environment, State, Action, Reward;
- освоить баланс между исследованием (Exploration) и использованием знаний (Exploitation);
- научиться сохранять и загружать накопленный опыт (Q-таблицу).

Формируемые компетенции: ПК-2, ПК-4

Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с лекционным материалом.

Оборудование и материалы

Для выполнения лабораторной работы рекомендуется использовать персональный компьютер со следующими программными средствами разработки (выбрать один или несколько программных продуктов для практической реализации задач лабораторной работы): google chrome / mozilla firefox, visual studio code, clion / microsoft visual studio 2022.

Методика и порядок выполнения работы

Перед выполнением заданий рекомендуется ознакомиться с решением учебной задачи:

<https://github.com/vberezina/GameAI/blob/main/LabWork9.md>

Порядок выполнения работы:

1. Создайте лабиринт 5x5.
2. Инициализируйте Q-таблицу нулями.
3. Реализуйте цикл обучения: агент делает ход -> получает награду -> обновляет таблицу по формуле Беллмана.
4. Визуализируйте процесс обучения через изменение цвета ячеек (яркость = ценность пути).

Задание 1. Среда и Агент.

Подготовка полигона для обучения.

- Отрисовать лабиринт, где есть цель (+100 очков) и ловушки (-100 очков).
- Реализовать перемещение агента по клеткам (вверх, вниз, влево, вправо).
- Добавить счетчик накопленной награды за эпизод.

Задание 2. Процесс обучения.

Реализация алгоритма q-обучения.

- Написать логику выбора действия (Epsilon-greedy strategy).
- Реализовать формулу обновления Q-значений после каждого шага.
- Запустить цикл из 1000 быстрых эпизодов без отрисовки, а затем показать результат.

Задание 3. Анализ стратегии.

Визуализация 'памяти' агента.

- Отрисовать в каждой клетке стрелочки, указывающие на действие с максимальным Q-значением.
- Проверить, как меняется путь, если переставить цель в другое место.
- Реализовать возможность 'ручного' добавления препятствий и посмотреть, как быстро агент переучится.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

1. Номер и название лабораторной работы; задачи лабораторной работы.
2. Реализация каждого задания с приведением исходного кода программы, диаграмм и графиков для визуализации данных.
3. Ответы на контрольные вопросы.

Отчет о выполнении лабораторной работы подписывается студентом и сдается преподавателю.

Контрольные вопросы

1. За что отвечает параметр Epsilon в стратегии выбора действий?
2. Почему награда за обычный шаг обычно делается слегка отрицательной (например, -1)?
3. Что произойдет, если коэффициент дисконтирования (γ) будет равен 0?

СПИСОК ЛИТЕРАТУРЫ

Основная литература:

1. Миллингтон И., Фандж Д. Искусственный интеллект в играх: теория и реализация. — М.: Вильямс, 2011
2. Рассел С., Норвиг П. Искусственный интеллект: современный подход. — М.: Вильямс, 2019
3. Бакленд М. Программирование ИИ на примерах. — М.: Вильямс, 2005
4. Nature of Code. Daniel Shiffman. [электронный ресурс] — <https://natureofcode.com> (главы по Steering Behaviors и Autonomous Agents)

Дополнительная литература:

5. Red Blob Games. Amit Patel. Пособие по алгоритмам поиска пути и сеткам. [электронный ресурс] — <https://redblobgames.com>
6. Game AI Pro series. Steve Rabin. Коллекция современных алгоритмов игрового ИИ. [электронный ресурс] — <http://gameai.pro>
7. Pygame Documentation. Справочное руководство по библиотеке и работе с векторами. [электронный ресурс] — <https://pygame.org>
8. Суттон Р., Барто Э. Обучение с подкреплением: Введение. — М.: ДМК Пресс, 2020

Интернет-ресурсы:

9. Блог red blob games (амит патель):

<https://redblobgames.com>

Интерактивное руководство по алгоритмам поиска пути, работе с гексагональными и квадратными сетками, а также процедурной генерации карт

10. The nature of code (дэниел шиффман):

<https://natureofcode.com>

Подробное описание физики и математики природных систем, включая рулевое управление агентами, флоккинг и генетические алгоритмы

11. Серия книг game ai pro:

<https://gameai.pro>

Открытый доступ к статьям от ведущих разработчиков индустрии по современным техникам принятия решений, тактическому ИИ и оптимизации

12. Документация библиотеки pygame:

<https://pygame.org>

Официальное руководство по использованию модулей отрисовки, обработки событий и математического модуля Vector2 для реализации логики ИИ