

УДК 004.896

Иерусалимов Илья Денисович – ассистент кафедры Инструментального и прикладного программного обеспечения МИРЭА – Российского технологического университета, г. Москва, Россия.

Шошников Иван Кириллович – ассистент кафедры Инструментального и прикладного программного обеспечения МИРЭА – Российского технологического университета, г. Москва, Россия.

**ПРОЕКТИРОВАНИЕ ГИБКОГО СРЕДСТВА ИМИТАЦИИ
ПОВЕДЕНЧЕСКИХ МОДЕЛЕЙ ЖИВОТНЫХ НА БАЗЕ НЕЙРОННОЙ
СЕТИ С ИСПОЛЬЗОВАНИЕМ РЕНДЕРИНГА РЕАЛЬНОГО ВРЕМЕНИ.**

Аннотация. В данной работе рассматривается процесс проектирования системы автоматизированной разработки игрового искусственного интеллекта на базе нейронной сети с использованием рендеринга реального времени. Разработана двухкомпонентная архитектура, включающая пользовательскую подсистему на базе Unity и серверную часть с нейронными сетями, взаимодействующими через протокол HTTP/2 с использованием фреймворка gRPC. Предложенный подход позволяет моделировать сложные поведенческие паттерны животных с применением метода обучения с подкреплением, что расширяет возможности традиционных методов разработки игрового ИИ. Практическая значимость работы заключается в создании инструмента для автоматизации процесса разработки поведенческих моделей с визуализацией результатов в режиме реального времени.

Annotation. This paper examines the process of designing an automated system for game artificial intelligence development based on neural networks with real-time rendering. A two-component architecture has been developed, including a Unity-based

user subsystem and a server component with neural networks interacting via the HTTP/2 protocol using the gRPC framework. The proposed approach enables modeling complex behavioral patterns of animals using reinforcement learning methods, which expands the capabilities of traditional game AI development methods. The practical significance of the work lies in creating a tool for automating the development process of behavioral models with real-time result visualization.

Ключевые слова: искусственный интеллект, unity, автоматизированное средство, gRPC, разработка программных средств.

Key words: artificial Intelligence, unity, automated tool, gRPC, software development

1 Исследовательский раздел

1.1 Обобщенная характеристика предметной области

1.1.1 Машина конечных состояний

Машина конечных состояний (конечный автомат) - это абстрактная модель дискретного устройства, имеющего один вход и один выход, содержащего конечное число состояний и находящегося в одном из них в любой момент времени. Используется для представления и управления потоком выполнения каких-либо команд.

Конечные автоматы различаются на детерминированные и недетерминированные. Детерминированные конечные автоматы — автоматы, в которых следующее состояние однозначно определяется текущим состоянием и выход зависит только от текущего состояния и текущего входа. Недетерминированные конечные автоматы – это автоматы, следующее состояние у которых в общем случае неопределённо и, соответственно, не определён выходной сигнал. Если переход в последующие состояния происходит с некоторыми вероятностями, то такой автомат называют вероятностным конечным автоматом.

Конечный автомат можно представить в виде графа, вершины которого являются состояниями, а ребра — переходы между ними. Каждое ребро имеет метку, информирующую о том, когда должен произойти переход. Ниже представлен граф конечного автомата, описывающего работу настольной лампы (Рисунок 1.1). Черным кругом обозначен вход в данный конечный автомат. Его наличие опционально, однако оно показывает, каким было состояние автомата в начальный момент времени.

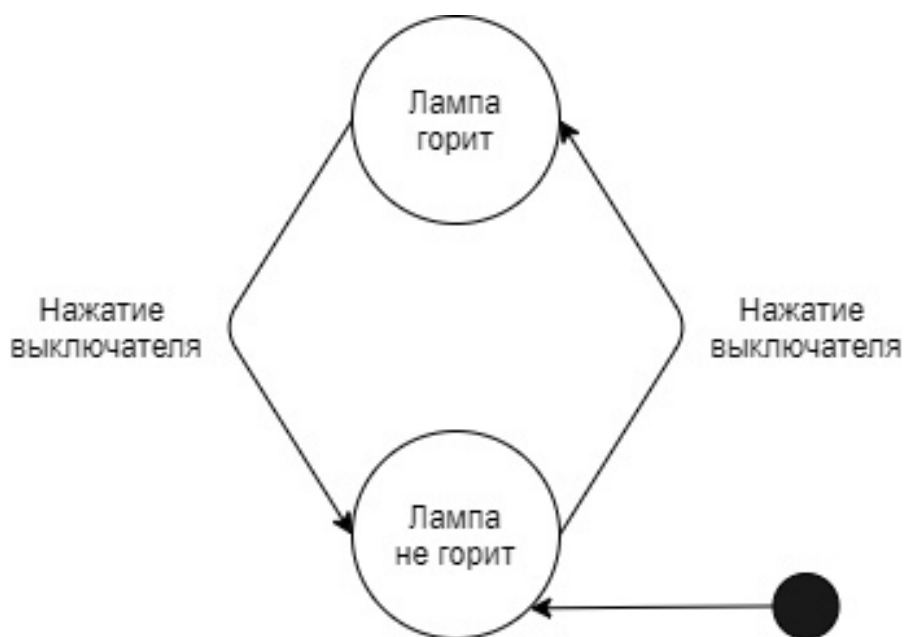


Рисунок 1.1 - Граф состояний настольной лампы

Конечный автомат идеально подходит для реализации искусственного интеллекта в играх, получая аккуратное решение без написания громоздкого и сложного кода.

1.1.2 Нейронные сети

Нейронная сеть – это математическая модель, получившая свое название из-за своего строения, аналогичного строению биологической нейронной сети. Это строение представляет собой множество соединенных между собой нейронов, образующих сложную сеть.

Математический нейрон по аналогии с биологическим имеет входы, через который он принимает сигналы. Простыми словами, он считает взвешенную сумму на своих входах, добавляет смещение (bias) и решает, следует это

значение исключать или использовать дальше. Поступившие на входы сигналы умножаются на свои веса – коэффициенты, отвечающие за влияние сигнала на нейрон. Сигнал каждого входа x_n умножается на соответствующий этому входу вес w_n . Затем все произведения передаются в сумматор. Сумматор – устройство, преобразующее группу сигналов в результирующий сигнал, эквивалентный по сумме входным. Иными словами, сумматор проводит операцию сложения сигналов со всех входов, помноженных на соответствующие веса. Затем результирующий сигнал передается на функцию активации. Функция активации – определяющая выходной сигнал нейрона. Существует много различных функций активации, которые подходят для разных ситуаций. Однако, все они сводятся к тому, что данная функция передает на выход нейрона значение, по которому определяется «активировался» нейрон или нет. Ниже приведена схема математического нейрона (Рисунок 1.2).

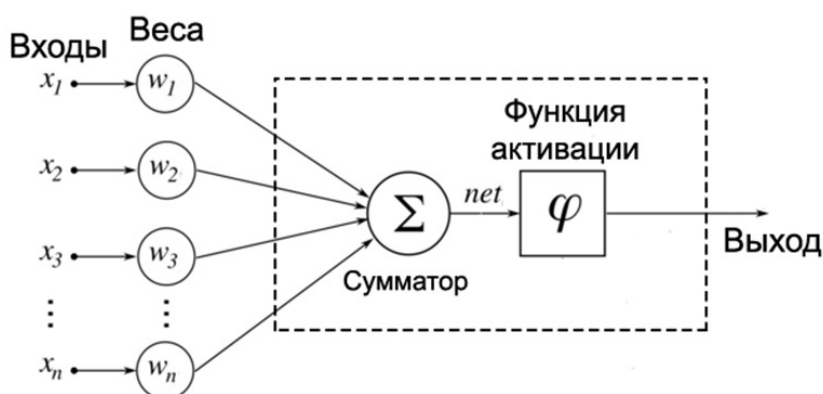


Рисунок 1.2 - Математическая модель нейрона

Обобщая, можно сказать, что математический нейрон похож на конечный автомат, но зависит от большего количества переменных, и способен выдавать на выход не дискретное значение.

Классифицируются нейронные сети по нескольким признакам, а именно:

- По настройке весов у нейронов (фиксированная, динамическая);
- По характеру обучения (с учителем, без учителя);
- По типу входной информации (аналоговые, двоичные);
- По модели нейронной сети (сети прямого распространения, рекуррентные, радиально-базисные функции, самоорганизующиеся карты).

Существует несколько методик обучения нейронных сетей. Среди них выделяется три наиболее распространенные:

- метод обратного распространения;
- метод упругого распространения;
- генетический алгоритм.

Метод обратного распространения использует алгоритм градиентного спуска. То есть при помощи движения вдоль градиента рассчитывается локальный минимум и максимум функции. В процессе поступления информации нейронная сеть последовательно передаёт её от одного нейрона к другому посредством синапсов, до того момента, пока информация не окажется на выходном слое и не будет выдана как результат. Такой способ называется передачей вперёд. После того как результат получен вычисляется ошибка и на её основании выполняется обратная передача, суть которой — последовательно изменить вес синапсов начиная с выходного и продвигаясь к входному слою. При этом значение веса меняется в сторону лучшего результата. Условием применения данного метода обучения является дифференцируемость функции активации. В сухом остатке обратное распространение – разница результатов, умноженная на производную функции активации от значения на входе. Ниже приведена иллюстрация данного метода (Рисунок 1.3).

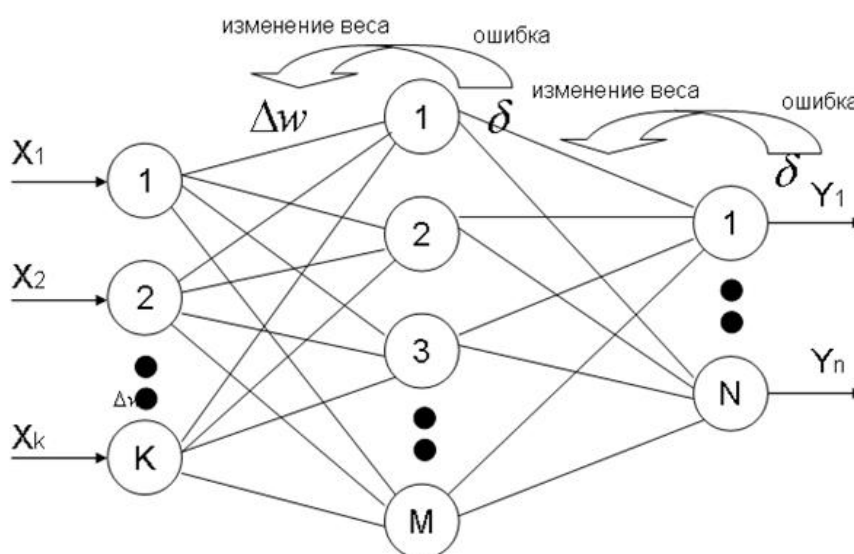


Рисунок 1.3 – Метод обратного распространения ошибки

Метод упругого распространения использует в качестве основы обучение по эпохам и применяет только знаки производных частного случая для корректировки весовых коэффициентов. Используется определенное правило, по которому производится расчёт величины коррекции весового коэффициента.

Последовательность действий в этом случае следующая: инициализируется величина коррекции, вычисляются частные производные, подсчитывается новое значение коррекции значений весов, корректируются веса, если условие остановки алгоритма не выполняется процесс переходит к вычислению производных и заново повторяет цикл. Данный метод значительно ускоряет процесс сходимости сети относительно предыдущего метода обучения нейронных сетей.

Генетический алгоритм представляет собой упрощённую интерпретацию природного алгоритма, основанного на скрещивании результатов. То есть, по сути, происходит скрещивание результатов, выбор наилучших и формирование на их основе нового поколения. В случае если результат не устраивает алгоритм повторяется пока поколение не становится идеальным. Алгоритм может завершиться без достижения нужного результата если количество попыток будет исчерпано или же будет исчерпано время на мутацию. Этот алгоритм применим к процессу оптимизации веса нейронной сети, при заданной по умолчанию топологии. В этом случае вес подвергается преобразованию в двоичный код, а каждый результат соответствует полному его набору. Выходная ошибка позволяет оценить качество прошедшего обучения.

Существуют и другие методы обучения нейронных сетей. Обучение с учителем — это тип тренировок присущий таким проблемам как регрессия и классификация. Нейронной сети предоставляются входные данные и желаемый результат, а она, как ученик, посмотрев на входные данные поймет, что нужно стремиться к тому результату, который ему предоставили.

Обучение без учителя — тип обучения, встречающийся не так часто. В этом методе нет учителя, поэтому сеть не получает желаемый результат или же их количество очень мало. В основном такой вид тренировок присущ нейронным

сетям, у которых задача состоит в группировке данных по определенным параметрам.

Еще один крайне полезный метод обучения – это обучение с подкреплением (reinforcement learning). Данный способ применим тогда, когда возможно, основываясь на результатах, полученных от нейронной сети, дать ей оценку. Иными словами, нейронной сети предоставляется право найти любой способ достижения цели до тех пор, пока результат не будет достаточно хорош. Таким способом, сеть начнет «понимать», чего от нее хотят добиться, и пытается найти наилучший способ достижения этой цели без постоянного предоставления данных «учителем». Ниже приведено схематичное изображение обучения с подкреплением (Рисунок 1.4).

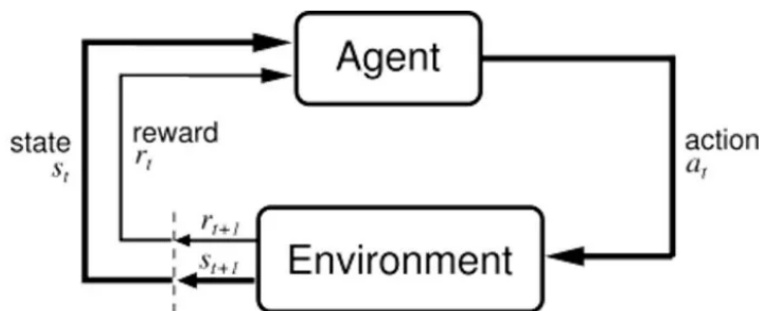


Рисунок 1.4 – Схема обучения с подкреплением

Основываясь на всем вышесказанном, можно сделать вывод, что на данный момент существует большое количество архитектур и методов обучения нейронных сетей, что позволяет подобрать нейронную сеть для решения практических любых задач. Данная сфера продолжает активно развиваться и расширять возможности машинного обучения.

1.2 Существующие технологии создания игрового искусственного интеллекта

Искусственный интеллект является крайне глубоким понятием. В общем случае, искусственный интеллект – это способность интеллектуальных систем принимать решения, выполнять творческие функции и обучаться. Однако для игровой индустрии подобная глубина является излишней.

В разработках игровой индустрии чаще всего используются гораздо более простые и узкоспециализированные аналоги искусственного интеллекта. Основная цель ИИ в играх состоит в имитации разумного поведения и в предоставлении игроку убедительной, правдоподобной опыта взаимодействия с ним.

Поскольку основной задачей, встающей перед игровым искусственным интеллектом, является принятие тех или иных игровых решений, чаще всего можно встретить две модели взаимодействия ИИ с игровыми механиками. Одна из них - это «вещание»: система ИИ является изолированной частью проекта и вычисляет оптимальные решения для игровых параметров. Решения, которые принимает ИИ в этом случае передаются участвующим игровым объектам. Эта модель часто используется в стратегиях и тактических симуляторах. Вторая модель больше подходит для игр с простыми объектами, то есть когда ИИ обрабатывает информацию не постоянно. Каждый раз, когда объект «думает», он обращается к ИИ, и обновляет свое состояние.

Одной из самых простых форм ИИ является система на основе правил. Однако, подобные системы очень далеки от настоящего искусственного интеллекта. Поведение игровых объектов в такой форме определяет набор алгоритмов, заданных заранее. При достаточно количестве заготовленных сценариев результат может выглядеть неявной поведенческой системой. Но по-настоящему «интеллектуальной» такая система являться не будет.

Одной из самых простых реализаций искусственного интеллекта в играх является конечный автомат. С его помощью можно описать объект, имеющий конечное число состояний в течение своего жизненного цикла. Состояния могут отражать положение объекта, поведенческие модели. Создать конечный автомат можно различными способами, используя инструменты объектно-ориентированных языков программирования. Ниже приведена схема переходов состояний простейшего игрового искусственного интеллекта (Рисунок 1.6).

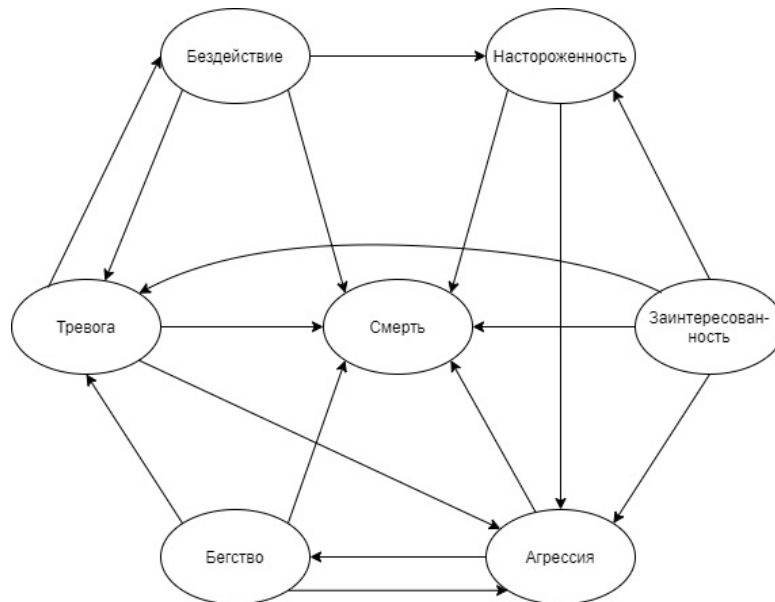


Рисунок 1.6 – Схема переходов состояний игрового ИИ

1.3 Характеристика поведенческих моделей

Вопрос классификации форм поведения животных достаточно многогранен и глубок. Попытки классифицировать поведенческие модели предпринимались с начала XIX века. Современная наука выделяет несколько способов классифицировать поведение животных. Одна из наиболее распространенных теорий основана на выделении следующих признаков, по которым возможно разделить на группы те или иные действия(рефлексы):

- Поведение, определяемое обменом веществ – например, поиск пищи, сон;
- Поведение, связанное с размножением – например, территориальное поведение, спаривание, забота о потомстве;
- Социальное поведение – взаимодействия внутри стаи/прайда;
- Оборонительное поведение – например, защитные реакции в случае опасности;
- Постройка гнезд/убежищ/нор;
- Комфортное поведение – например, уход за телом.

Обмен веществ является одним из основных процессов, отличающих живых существ. В основе лежит взаимодействие центральных механизмов возбуждения

и торможения. Элементы, из которых состоят эти процессы, отвечают, как за реакцию на различные пищевые раздражители, так и на характер движений при еде. Определенную роль в формировании пищевого поведения играет индивидуальный опыт животного, в частности опыт, определяющий ритмы поведения. Запасание пищи также относится к пищевому типу поведения. Процесс созданию запасов крайне специфичен для каждого вида. Поведенческие модели, проявляющиеся во время сна также уникальны для различных видов.

Половое поведение описывает все многообразные поведенческие акты, связанные с процессом размножения. Оно относится к наиболее важным формам поведения, так как связано с продолжением рода. В половом поведении, особенно у низших животных, большую роль играют ключевые раздражители (релизеры). Существует великое множество релизеров, которые в зависимости от ситуации могут вызвать или сближение половых партнеров, или драку. Действие релизера напрямую зависит от равновесия совокупности составляющих его раздражителей. Большая часть релизеров, выставляемых напоказ во время брачных игр, напоминает незавершенные движения, которые в обычной жизни используются для совершенно иных целей. У большинства животных в половом поведении выделяются отдельные поведенческие блоки, которые выполняются в строго определенной последовательности.

Общественное поведение отличается большим разнообразием. Существует множество классификаций различных типов объединения животных, а также особенностей поведения животных внутри разных групп. У птиц и млекопитающих встречаются различные переходные формы организации от одиночной семейной группы до истинного сообщества. Внутри этих групп взаимоотношения построены, в основном, на разнообразных формах полового, родительского и территориального поведения, но некоторые формы характерны только для животных, живущих в сообществах. К общественному поведению также относится групповая забота о потомстве. К социальному поведению относят и совместное выполнение работы, которое управляется системой

сенсорной регуляции и координации. Такая совместная деятельность заключается в строительстве, невозможном для отдельной особи.

Оборонительное поведение - реакция на изменение во внешней среде. Оборонительные рефлексы могут возникать в ответ на любые факторы внешней или внутренней среды: звуковые, вкусовые, болевые, термические и другие раздражители. Оборонительная реакция может носить и местный характер, и принимать характер общей поведенческой реакции животного. Поведенческая реакция может выражаться и в активной обороне или нападении, и в пассивном замирании на месте. К особой форме оборонительного поведения относятся реакции избегания, за счет которых животные сводят к минимуму попадание в опасные ситуации. Избегание относится к высокоспецифичным рефлексам.

Комфортное поведение - это многообразные поведенческие акты, направленные на уход за телом, а также сопутствующие движения, не имеющие определенного пространственного направления и места локализации. Комфортное поведение, именно та его часть, которая связана с уходом животного за своим телом, может быть рассмотрена как один из вариантов манипулирования, причем, в данном случае, в качестве объекта манипуляций выступает само тело животного. Подобное поведение видотипично, а последовательность действий по очищению тела, зависимость определенного способа от ситуации врожденны и проявляются у всех особей.

Рассмотрев вышеупомянутую классификацию поведенческих моделей животных, можно сделать вывод о том, что подобные модели многообразны и уникальны для каждого вида живых существ, однако все данные процессы обусловлены простой логикой, которую можно передать техническим языком.

1.4 Техническое задание на разработку гибкого средства

Общие технические требования на разработку гибкого средства:

- Гибкое средство должно состоять из двух или более подсистем для возможности использования для вычислений специализированных устройств;
- Данные между подсистемами должны передаваться с

использованием протокола HTTP/2 для достаточной скорости передачи;

– Средство должно иметь интерфейс, отображающий действия моделируемого животного с рендером в реальном времени.

2 Аналитический раздел

2.1 Описание структуры гибкого средства

Гибкое средство для создания поведенческих моделей животных состоит из двух подсистем: пользовательской и серверной. Пользовательская часть представляет собой проект на движке Unity, и располагается на устройстве пользователя. Серверная часть содержит нейронные сети и может располагаться либо на удаленном выделенном сервере, либо на устройстве пользователя, при наличии необходимой вычислительной мощности. Взаимодействие между подсистемами осуществляется при помощи фреймворка gRPC. Ниже приведен пример работы системы с использованием gRPC (Рисунок 2.1).

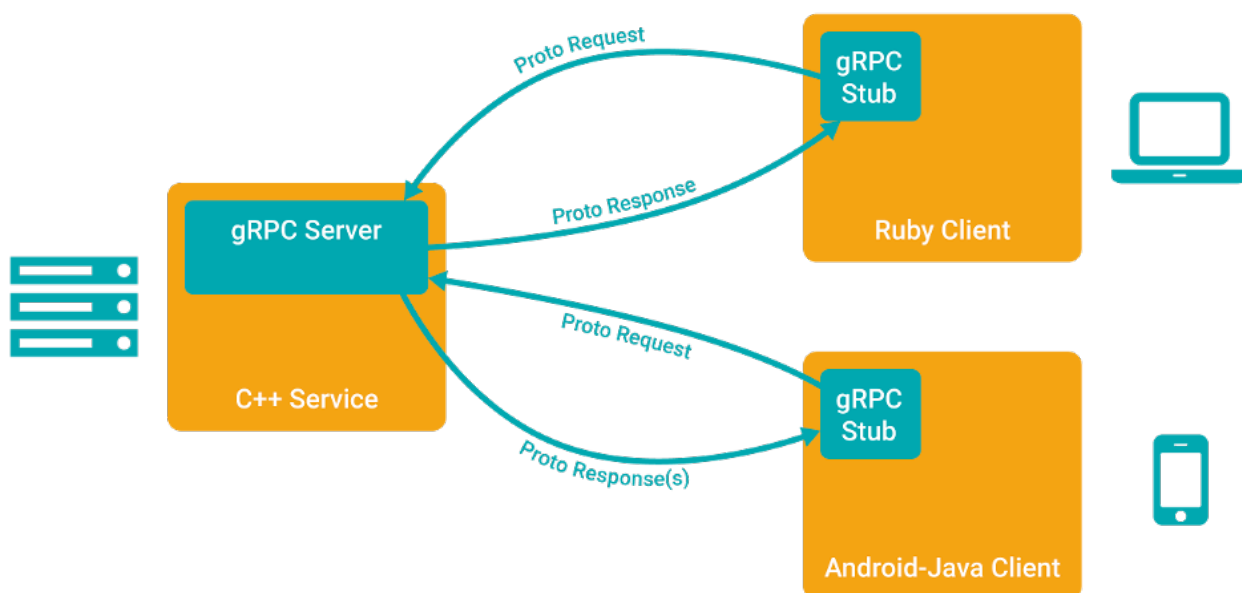


Рисунок 2.1 – Пример работы gRPC

Внутри пользовательской части реализован клиент gRPC, прикрепленный к объекту-животному. С заданной частотой клиент посылает на сервер запрос, содержащий в себе перечень заданных параметров. В данном этапе жизненного цикла проекта этот перечень можно задать жестко, однако при последующей доработке системы возможно создание интерфейса для управления этими параметрами.

Сервер gRPC принимает данные от пользовательской части, а затем использует их в качестве входных параметров для нейронной сети. На основе этих полученных данных нейронная сеть обучается и выдает решение, относительно дальнейшего действия. Это решение в качестве параметра отправляется на пользовательский клиент, где и отображается в виде действий животного.

Общая структура средства выглядит следующим образом (Рисунок 2.2):



Рисунок 2.2 – Схема структуры гибкого средства

2.2 Описание математического обеспечения гибкого средства

Математическое обеспечение гибкого средства должно включать в себя следующее:

Для реализации математической модели нейронных сетей необходимо использовать язык программирования Python версии не менее 3.0 с применением таких библиотек, как TensorFlow и Keras. Используемые нейронные сети должны классифицироваться, как полносвязные сети с динамическими весами, принимающими аналоговые входные данные.

Математические алгоритмы вычисления должны быть оптимальны для реализации в программном виде.

Дополнительных требований к математическому обеспечению пользовательской части не предъявляется.

2.3 Описание информационного обеспечения гибкого средства

Требования к составу, структуре и способы организации данных в системе:

Данные пользовательской подсистемы должны храниться на внутреннем носителе информации устройства, на котором она расположена;

Данные серверной подсистемы, содержащие в себе нейронные сети, должны храниться на внутреннем носителе выделенного сервера или другого устройства, на котором подсистема расположена.

Требования к информационному обмену между частями системы:

Обмен данными между пользовательской и серверной подсистемами должно происходить по интерфейсу Wi-Fi или другому прямому соединению между подсистемами с использованием системы gRPC.

Представление данных для сервера с нейронной сетью должны представляться в формате JSON и быть четко структурировано. Структура передаваемых данных должна быть задана заранее.

Требования к защите данных от разрушений при авариях и сбоях в электропитании системы:

Актуальными должны быть только те данные, которые были успешно переданы между подсистемами. Данные, находящиеся в процессе передачи, должны быть недоступны для чтения до окончания их передачи.

Требования к контролю, хранению, обновлению и восстановлению данных:

Подсистемы средства должны жестко проконтролировать все события, связанные с изменением данных внутри них, и иметь возможность в случае сбоя восстановить свое состояние, используя ранее запротоколированные изменения данных.

Данные, переданные внутри системы успешно должны храниться и быть доступны до момента, пока они не будут заменены новыми, успешно переданными данными. В рамках серверной части системы были реализованы механизмы для передачи данных, а также для обучения нейронной сети. Обмен данными реализован при помощи фреймворка gRPC через определённый заранее порт. На данном этапе развития системы адреса подсистем и порт взаимодействия заранее прописаны в коде средства. При дальнейшем развитии системы возможно создание интерфейса для редактирования этих параметров.

Нейронные сети имеют следующую структуру: в качестве модели обучения выбрана Reinforcement learning – обучение с подкреплением без учителя. Данная модель обучения оптимальна, поскольку задача, реализуемая разрабатываемым средством – это моделирование взаимодействия объекта со средой. Алгоритм обмена нейронной сети со средой, представленной сценой в Unity, является основным в разрабатываемой системе. Для данного обучения необходима полносвязная нейронная сеть, состоящая из трех слоев.

Тестирование инициализации новой сети и ее последующее обучение не требовало дополнительных манипуляций над системой и завершилось успешно.

Заключение

В данной работе мною были произведены следующие действия исследовательского характера:

- Изучены принципы работы полносвязных нейронных сетей и их обучения;
- Изучены интерфейсы и протоколы, используемые для передачи данных между платформами;
- Исследованы методы создания игрового искусственного интеллекта;
- Разработан метод создания поведенческих моделей живых существ при помощи нейронных сетей;
- Выбрана архитектура и средства, для реализации данного метода;
- Составлен план разработки программного обеспечения для разрабатываемого гибкого средства;
- Разработан алгоритм работы ПО для пользовательского устройства и сервера.
- Разработан алгоритм обмена данными между пользовательским ПО и сервером.

В данной работе я столкнулся с некоторыми сложностями, такими как выбор архитектуры нейронных сетей и способа обучения. Также были проблемы с длительностью обучения и подбором устройства, удовлетворяющего техническим требованиям для него.

После выполнения данной работы я получил бесценный опыт в проектировании и обучении нейронных сетей. Однако данный этап жизненного цикла рассматриваемой разработки не является заключающим. Возможно доработка данного гибкого средства до плагина, используемого движком Unity, что позволит легко подключать его к проекту любого рода

Список информационных источников

1. Статья о развитии реалистичности игр. [Электронный ресурс]
URL: <https://gamemag.ru/news/152960/take-twos-ceo-thinks-games-will-look-exactly-like-live-action-in-10-years> (последнее обращение 15.03.2021)
2. Инфографика прибыли рынка мультимедиа индустрии. [Электронный ресурс] URL: <https://dtf.ru/gameindustry/294571-igrovaya-industriya-za-2020-god-v-cifrah-infografika-ot-gamesindustry-biz> (последнее обращение 15.04.2021)
3. Статья об игровой индустрии. [Электронный ресурс] URL: <https://dtf.ru/gameindustry/294571-igrovaya-industriya-za-2020-god-v-cifrah-infografika-ot-gamesindustry-biz> (последнее обращение 05.04.2021)
4. Сводки о доходах игровой индустрии. [Электронный ресурс] URL: <https://plus.rbc.ru/news/5bb572797a8aa90ae8e15656> (последнее обращение 10.04.2021)
5. Статья об обучении нейронных сетей. [Электронный ресурс] URL: <https://livesurf.ru/zhurnal/6068-metody-obucheniya-nejronnyh-setej.html> (последнее обращение 18.04.2021)
6. Статья о классификации поведенческих моделей животных. [Электронный ресурс] URL: https://studref.com/396277/psihologiya/povedenie_zhivotnyh (последнее обращение 15.05.2021)
7. Информативная статья о методах обучения и их различиях. [Электронный ресурс] URL: <https://habr.com/ru/post/313216/> (последнее обращение 20.04.2021)
8. Статья о нейронных сетях. [Электронный ресурс] URL: <https://habr.com/ru/post/312450/> (последнее обращение 20.02.2021)
9. Полезная информации по язык программирования Python. [Электронный ресурс] URL: <https://pythonworld.ru/samouchitel-python>

(последнее обращение 19.02.2021)

10. Подробное описание программы Blender3D. [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/Blender> (последнее обращение 24.04.2021)

11. Электронный ресурс с большим количеством тематических уроков и подсказок по использованию программы Blender3D на русском языке. [Электронный ресурс] URL: <https://blender3d.com.ua/> (последнее обращение 24.04.2021)

12. Статья «Полный цикл создания модели персонажа для игры» с большим количеством полезной информации. [Электронный ресурс] URL: <https://habr.com/ru/post/341050/> (последнее обращение 26.02.2021)

13. Статья о преимущества движка Unreal Engine 4 [Электронный ресурс] URL: <https://cubiq.ru/dvizhok-unreal-engine/> (последнее обращение 25.05.2021)

14. Описание движка Unity [Электронный ресурс] URL: [https://ru.wikipedia.org/wiki/Unity_\(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA\)](https://ru.wikipedia.org/wiki/Unity_(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA)) (последнее обращение 15.05.2021)

15. Документация gRPC [Электронный ресурс] URL: <https://docs.microsoft.com/ru-ru/dotnet/architecture/cloud-native/grpc> (последнее обращение 10.05.2021)

16. Методические рекомендации по выполнению организационно-экономической части выпускных квалификационных работ [Электронный ресурс]: метод. указания / Т. Ю. Гавриленко, О. В. Григоренко, Е. К. Ткаченко. — М.: РТУ МИРЭА, 2019. — Электрон. опт. диск (ISO) (последнее обращение 10.05.2021)

17. Григоренко О.В., Садовнича И.О., Мыльникова А. Экономика предприятия и управление организацией М.: РУСАЙНС, 2017-235с. (последнее обращение 09.05.2021)