

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики
Направление подготовки – 09.03.01 «Информатика и вычислительная техника»
Кафедра вычислительной техники

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка и программная реализация нейросетевого алгоритма для принятия решений в интеллектуальной игре «Покер»

УДК 004.7.032.26:519.7

Студент

Группа	ФИО	Подпись	Дата
8В2А	Зиганшин Алексей Тахирович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ВТ	Хаустов П.А.	-		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. МЕН	Николаенко В.С.	-		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ЭБЖ	Невский Е.С.	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ВТ	Марков Н.Г.	Д.Т.Н., профессор		

**ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ПО ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ
ПРОГРАММЕ ПОДГОТОВКИ БАКАЛАВРОВ 09.03.01 «ИНФОРМАТИКА И
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА», ИК ТПУ, ПРОФИЛЬ «ВЫЧИСЛИТЕЛЬНЫЕ
МАШИНЫ, КОМПЛЕКСЫ, СИСТЕМЫ И СЕТИ»**

Код результата тов	Результат обучения (выпускник должен быть готов)
<i>Профессиональные компетенции</i>	
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<i>Универсальные компетенции</i>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Николаенко Валентин Сергеевич
Социальная ответственность	Невский Егор Сергеевич

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	10.09.2015
---	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ВТ	Хаустов Павел Александрович	-		10.09.2015

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В2А	Зиганшин Алексей Тахирович		10.09.2015

Министерство образования и науки Российской Федерации
 Федеральное государственное автономное образовательное учреждение
 высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики
 Направление подготовки – 09.03.01 «Информатика и вычислительная техника»
 Уровень образования – Бакалавриат
 Кафедра Вычислительной техники
 Период выполнения осенний / весенний семестр 2015/2016 учебного года

Форма представления работы:

Бакалаврская работа

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
 выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	10.06.2016
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.09.2015	Составление и утверждение ТЗ	5
12.10.2015	Изучение существующих концепций покерных интеллектов	5
05.11.2015	Изучение теоретического материала по теории генетических алгоритмов и искусственных нейронных сетей	5
01.12.2015	Проектирование клиент-серверного приложения	5
10.02.2016	Разработка клиент-серверного приложения	25
01.03.2016	Разработка инструментов обучения и тестирования нейросетевых интеллектов	20
30.04.2016	Обучение и тестирование ИИ	25
20.05.2016	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	5
1.06.2016	Социальная ответственность	5

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ВТ	Хаустов Павел Александрович	-		10.09.2015

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Вычислительной техники	Марков Николай Григорьевич	д.т.н.		10.09.2015

РЕФЕРАТ

Выпускная квалификационная работа содержит 73 страницы, 14 рисунков, 4 таблицы, 18 источников.

Ключевые слова: игровой искусственный интеллект, искусственные нейронные сети, генетический алгоритм.

Объектом исследования является задача разработки игрового искусственного интеллекта.

Цель работы – разработка и исследование искусственного игрового интеллекта, с внедрением его в многопользовательское игровое клиент-серверное приложение.

В ходе выполнения работы было реализовано многопользовательское игровое клиент-серверное приложение. Кроме того, были реализованы инструменты, необходимые для обучения и тестирования нейросетевых интеллектов.

В результате исследования были получены различные варианты игровых искусственных интеллектов, эффективность применения которых была оценена на практике.

Область применения: интеллектуальные системы, методы машинного обучения, моделирование поведения сложных систем.

Значимость работы заключается в инновационном подходе к решению одной из наиболее популярных задач в IT сфере – создание искусственного интеллекта, моделирующего поведение человека.

Полученные результаты позволили определить наиболее перспективные направления для дальнейших исследований. В будущем планируется некоторым образом изменить полученную концепцию искусственного интеллекта, чтобы устранить слабые места текущего решения.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ

Искусственные нейронные сети – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма [1].

Формальный нейрон – узел искусственной нейронной сети, являющийся математической моделью естественного нейрона [2].

Функция активации – функция, вычисляющая выходной сигнал искусственного нейрона.

Топология искусственной нейронной сети – организация связей между формальными нейронами в искусственной нейронной сети.

Генетический алгоритм – эвристический алгоритм поиска, основывающийся на моделировании процесса естественного отбора и эволюции в природе [3].

Особь – центральное понятие генетического алгоритма, представляет собой одно из возможных решений поставленной задачи, соответственным образом закодированное.

Ген – неделимая единица информации, носителей которой является особь. Совокупность генов называется **генотипом** особи.

Популяция – множество особей. Смежное понятие с термином «Поколение», но последнее чаще всего упоминается в контексте сменяющихся друг друга популяций в процессе итеративного поиска.

Функция приспособленности – функция, отображающая из множества особей во множество действительных (целых) чисел. Полученное в результате отображения число называется значением функции приспособленности и характеризует удаленность данной особи от идеального решения по некоторой метрике.

Селекция – процесс оценивания популяции и отбора согласно некоторой стратегии тех особей, которые будут допущены до скрещивания.

Оператор скрещивания (кроссинговера) – функция, которая принимает на вход две особи, называемые **родительскими**, а на выход выдает две т.н. **дочерние** особи, каждая из которых наследует часть генотипа родителей [4].

Оператор мутации – процесс внесения случайных изменений в генотип некоторой особи. Мутация необходима для расширения пространства поиска и для того, чтобы в популяции появлялись решения, которых не было изначально [5].

ОБЪЕКТ И МЕТОДЫ ИССЛЕДОВАНИЯ

В данной работе объектом исследования является задача создания искусственного игрового интеллекта. Предметом исследования являются алгоритмы, позволяющие решать поставленную задачу.

Также были использованы следующие методы исследования:

- Анализ и синтез – в работе были проанализирована структура задачи создания нейросетевых интеллектов, особенности эволюционного подхода в ее решении. Был синтезирован генетический алгоритм, осуществляющий обучение нейросетевого интеллекта;
- Идеализация – этот метод является одним из основных в концепции генетических алгоритмов и искусственных нейронных сетей, поскольку используются упрощенные и идеализированные модели реальных объектов и явлений;
- Аксиоматический метод – использование теории нейросетевых вычислений и теории эволюционных вычислений влечет за собой использование некоторого набора аксиом, т.е. утверждений, которые считаются истинными и не требуют доказательств;
- Эксперимент и измерение – в данном исследовании эти методы являются центральными, поскольку особенности нейросетевых интеллектов затрудняют их теоретический анализ, что приводит к необходимости проведения серии тестов для оценки качества полученного решения;
- Сравнение – данная работа предполагает получение некоторого множества решений, для оценки эффективности которых необходимо провести сравнение с ранее полученными вариантами искусственных интеллектов.

Содержание

	С.
Введение	12
1. Теоретические основы	14
1.1 Клиент-серверные технологии	14
1.1.1 Сокеты, протоколы передачи данных.....	14
1.2.1 Сравнение популярных графических движков.....	16
1.3 Проблема создания искусственного игрового интеллекта	19
1.3.1 Основные концепции построения покерных интеллектов ...	19
1.3.2 Генетические алгоритмы.....	22
1.3.2.1 Особенности генетических алгоритмов.....	22
1.3.2.2 Оценивание популяций.....	23
1.3.2.3 Селекция.....	24
1.3.2.4 Скрещивание и формирование нового поколения.....	24
1.3.2.5 Мутация.....	25
2. Проектирование клиентского приложения	27
2.1 Устройство пользовательского приложения.....	27
2.2 Устройство менеджера интеллектов.....	34
2.2.1 Структура менеджера игровых интеллектов.....	35
2.2.2 Модуль обучения игрового интеллекта, основанного на искусственной нейронной сети	37
2.2.3 Модуль централизованного управления экземплярами игровых интеллектов	39
2.2.4 Модуль игровых интеллектов.....	41
3. Описание предлагаемых решений.....	43
3.1 Кодирование особи генетического алгоритма.....	43
3.2 Оценивание особей в популяции.....	43
3.3 Оператор кроссинговера.....	44
3.4 Оператор мутации.....	45
4. Полученные результаты	46

4.1	Описание подхода к обучению каскадов нейронной сети.....	47
4.2	Результаты экспериментов.....	49
5.	Менеджмент, ресурсоэффективность и ресурсосбережение.....	51
5.1	Введение.....	51
5.2	SWOT-анализ.....	52
5.3	Вывод.....	61
6.	Социальная ответственность.....	62
6.1	Влияние на общество.....	62
6.2	Игровая зависимость.....	62
6.3	Искусственный интеллект.....	67
	Заключение.....	70
	Список публикаций.....	72
	Список использованных источников.....	74

ВВЕДЕНИЕ

На сегодняшний день индустрия компьютерных игр является бурно развивающимся сектором экономики. Разработкой игр занимаются как крупные компании мирового уровня, так и небольшие команды разработчиков, известные только узкому кругу пользователей. Некоторые из компьютерных игр являются одними из наиболее требовательных приложений на персональных компьютерах, для создания и воспроизведения которых используются самые передовые технологии. Другие же игры наоборот, не требуют особой производительности от платформы и способны воспроизводиться даже на смартфонах.

Среди всего разнообразия компьютерных игр особое место занимают многопользовательские онлайн игры. Благодаря использованию клиент-серверных технологий давно прошли те дни, когда люди нуждались в личных встречах с друзьями, чтобы играть в игры. Теперь можно играть когда угодно, и где угодно. Некоторые игры настолько просты, но в то же время увлекательны, что в них можно играть в обеденный перерыв, или по пути на работу. Многие из них созданы по мотивам известных настольных или азартных игр. Подобные игры популярны среди людей, которые хотят отвлечься от повседневной суеты или провести своё свободное время в общении с друзьями.

Однако популярность многопользовательских компьютерных игр обуславливается не только живым общением с другими игроками, но ещё и тем, что человек непредсказуем в своём поведении, он не ограничен никакими рамками. В то время как игровые интеллекты, основанные на стандартных алгоритмах, зачастую предсказуемы и достаточно быстро надоедают. Однако существуют определённые подходы в создании игровых интеллектов, которые позволяют имитировать поведение человека. Например, использование искусственных нейронных сетей позволяет создать игровой интеллект, который в своём поведении практически неотличим от человека.

В настоящее время искусственные нейронные сети применяются повсеместно: начиная от систем распознавания речи до распознавания вторичной структуры белка, классификации различных видов рака и генной инженерии. Развитие данной области программирования позволяет как можно ближе приблизиться к созданию полноценного искусственного интеллекта, способного заменить человека во многих областях его деятельности.

Не на последнем месте стоит применение ИНС в покерных интеллектах, над созданием которых бьются исследовательские группы из многих университетов мира. Покер, который по своей сложности в поиске оптимального решения не уступает шахматам, является идеальной задачей для изучения искусственных нейронных сетей и методов их обучения.

Таким образом, в данной работе будут применены активно развивающиеся технологии программирования и исследованы актуальные проблемы на примере решения достаточно нетривиальной задачи. Всё это обеспечивает значительную актуальность данной работы.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

1.1 Клиент-серверные технологии

1.1.1 Сокеты, протоколы передачи данных

Как правило, любое сетевое взаимодействие не может обойтись без сокетов. Сокеты разделяются в основном на два типа: UDP и TCP. Для корректного выбора типа сокета нужно иметь представление о специфике работы разрабатываемого приложения [6]. Для данной научно-исследовательской работы необходим сокет с высокой надежностью отправки пакетов данных, при этом скорость передачи данных не столь важна. Рассмотрим подробнее свойства каждого типа сокетов.

Протокол TCP обладает свойствами, необходимыми для реализации научно-исследовательской работы: надежность, гарантированная доставка пакетов, использование контрольных сумм, повторная передача, настраиваемое время жизни пакета и прочее. Клиент-серверное взаимодействие, спроектированное в рамках данной работы, требует строгую очередность при отправке и принятии команд с данными. В случае некорректной последовательности принятых команд, могут произойти сбои в работе серверного приложения. Поэтому гарантия строгой очередности доставки пакетов является главным преимуществом рассматриваемого протокола. Кроме того, протокол TCP полностью автоматизирован, что облегчает задачу программистам проекта.

Технология «скользящего окна», используемая протоколом TCP, построена по принципу «запрос-ответ». Протокол TCP не гарантирует высокую скорость передачи [7]. Но, так как многопользовательское приложение, реализованное в этом проекте, не требует быстрого отклика (10-50мс), то этот недостаток протокола не является критическим.

Протокол UDP характеризуется высокой скоростью передачи данных, но имеется существенный минус. Данный протокол не гарантирует, что пакеты вообще дойдут до получателя, хотя вероятность потери этих пакетов

пренебрежительно мала. На практике, в среднем доля потерянных пакетов составляет 3-5%. Минимальные накладные ресурсы для отправки пакетов, а также отсутствие надобности подтверждения получения пакета, позволяют протоколу передавать данные с малым временем отклика, что является несравненным преимуществом для игр таких жанров: шутеры, онлайн симуляторы [8].

Подводя итоги, можно выделить основные качества каждого из перечисленных типов протоколов:

TCP:

- использует принцип соединений
- гарантирует очередность и доставку;
- автоматически формирует пакеты с информацией;
- контролирование потока данных;
- простота в использовании.

UDP:

- не использует принцип соединений;
- не гарантирует очередность доставленных пакетов;
- пакеты могут не дойти вообще;
- нужно вручную разбивать данные на пакеты и отправлять их;
- отсутствие контролирования потока данных;
- необходимость отслеживать вручную потерянные пакеты.

Проведённый анализ показал, что наиболее целесообразным решением для реализуемого проекта является использование протокола TCP. Основное преимущество протокола UDP – скорость передачи данных, не является приоритетным качеством для рассматриваемой задачи. В то же время протокол TCP, обеспечивая надёжную передачу данных в строго установленном порядке, удовлетворяет главным требованиям проекта.

1.2 Выбор и обоснование выбора инструментов разработки

1.2.1 Сравнение популярных графических движков

Ни одно игровое приложение не способно работать без физического «движка». На текущий момент на рынке существует множество полноценных игровых «движков». В данной главе будут рассмотрены наиболее популярные из них. При этом основным критерием выбора графического движка для данной работы является кросс-платформенность.

Unity 3D – один из самых востребованных и используемых графических «движков» на сегодняшний день [9]. Его главное преимущество заключается в высоком уровне технической поддержки: любая новая технология, разработанная в игровой индустрии, сразу же реализуется в этом графическом «движке». Огромным достоинством также является свободный доступ для начинающих разработчиков. Так как проект такого масштаба у разработчиков является дебютным, свободный доступ среды разработки является немаловажным параметром. Однако, не все платформы доступны без приобретения лицензии. Например, платформы Android, iOS и BlackBerry доступны только в платной версии. В свободно распространяемый набор входят платформы OS X и Windows, являющиеся основными для разработчиков, поэтому бесплатная версия приложения является приемлемым выбором. Поддерживаемые языки программирования Unity 3D: Javascript, C#, Boo.

Достоинства Unity 3D:

- свободный доступ;
- обширное сообщество;
- кросс-платформенность;
- низкий порог вхождения;
- популярность

Основным недостатком графического «движка» Unity является ограниченный набор инструментов, находящихся в свободном доступе.

Unreal Engine 3 чаще используется для консолей, но все же он так же достоин внимания. Графические возможности этого «движка» крайне высоки, но приложения не требовательны к ресурсам вычислительных систем, поэтому могут использоваться и на мобильных устройствах [10]. В отличие от Unity, есть свой инструмент для дизайна игровых сцен в разрабатываемой среде. Также графический «движок» поддерживает обширное множество платформ, включая Android, iOS, iPhone 8, Windows. Поддерживаемым языком программирования является Unreal Script.

Достоинства Unreal Engine:

- постоянная техподдержка;
- все новые инструменты попадают в ближайшее обновление;
- выгодная лицензионная политика;
- обширный набор инструментов.

Недостатки:

- средний порог вхождения.

CryEngine 3 – мощный графический «движок», разработанный компанией Crytek [11]. Он намного мощнее двух, ранее рассмотренных в этом разделе, но при этом, порог вхождения значительно выше. CryEngine 3 больше подходит для игр с дополненной реальностью, нежели для карточного симулятора. Для изучения он доступен бесплатно, но чтобы разрабатывать на нем коммерческие проекты, необходимо приобрести лицензию на использование.

Достоинства CryEngine 3:

- самый мощный графический «движок»;
- поддержка музыкального сопровождения;
- легко разрабатывать пользовательский интерфейс.

Недостатки:

- маленькое сообщество;
- относительно высокий порог вхождения.

Подводя итоги, можно сказать, что низкий порог вхождения для разработки данного приложения является основным критерием выбора, а графические способности «движка» не так важны. Несмотря на то, что всеми «движками» можно пользоваться в свободном доступе для учебных целей, наиболее предпочтительным выбором является Unity 3D, так как он имеет самый низкий порог вхождения и достаточно большой круг пользования, поэтому найти справочный материал не является проблемой.

1.3 Проблема создания искусственного игрового интеллекта

1.3.1 Основные концепции построения покерных интеллектов

Существует три основные концепции стратегических профилей: экспертные системы, эксплуататоры и равновесные стратегии.

Исторически первыми появились экспертные системы, которые представляют собой набор правил if-then, составляемых программистом системы. Строго говоря, экспертная система представляет собой набор различных частных случаев, которые могут произойти, и чем больше частных случаев рассмотрено, тем более эффективной будет система. Недостаток данного подхода очевиден – невозможно перебрать все возможные частные случаи, так или иначе останутся «дыры», при выявлении которых покерный интеллект становится легко эксплуатируемым. Конечно, при обнаружении уязвимости экспертную систему можно доработать, но в связи с этим будет расти сложность поддержки, а также будет снижаться производительность в целом. Поэтому данный подход на сегодняшний день наименее популярен.

Следующий вид стратегических профилей называется эксплуататором. Данный профиль достаточно сложен как в реализации, так и в применении, но в большинстве случаев он является невероятно эффективным.

Данный вид интеллектов получил такое название благодаря тому, что он использует слабости своих противников для максимизации прибыли. Для этого интеллектом моделируется поведение противника в той или иной ситуации, либо используются заранее посчитанные стратегические профили под определенного оппонента. При этом решение принимается после многократного проигрывания «виртуальных» раздач с различными исходами, что позволяет определить наиболее выгодный ход.

Таким образом, эффективность решений такого интеллекта будет зависеть от адекватности модели поведения оппонента. Для построения модели поведения, как правило, используются искусственные нейронные

сети. Здесь и заключается основной недостаток данных интеллектов: они нуждаются в предварительном обучении, которое обеспечит начальную стратегию поведения. При этом необходимо учитывать, что, даже имея начальное обучение, невозможно за несколько десятков раздач построить адекватную модель поведения оппонента. Скорее, для этого понадобятся десятки тысяч раздач, что существенно сказывается на области применения данных ботов: в играх со случайными игроками они становятся чрезвычайно уязвимыми. Однако данный подход является достаточно распространённым.

При создании третьего вида покерных интеллектов применяется принцип равновесных стратегий. Применение данных стратегий можно рассматривать как игру двух эксплуататоров, постоянно подгоняющих свою смешанную стратегию под оппонента. Очевидно, что рано или поздно стратегии обоих игроков сойдутся к равновесию, когда нельзя будет с высокой долей вероятности предсказать ход оппонента. В таком случае становится безразличным, какой ход предпринять. Такая особая ситуация получила названия равновесия Нэша, поскольку находясь в равновесии ни один из игроков не может сыграть лучше другого, изменив свою стратегию.

Однако, как уже говорилось ранее, исходная игра слишком сложна для обработки и построения равновесных стратегий. Поэтому исследователи сосредоточились на равновесных стратегиях абстракций покера, но тут возникает вопрос, связанный с тем, насколько адекватно равновесие абстрактной игры соответствует равновесию полной игры? В связи с этим равновесие Нэша абстрактной игры принято называть приближенными равновесиями или ϵ -равновесиями, где ϵ символизирует потенциальную эксплуатируемость стратегии.

У ϵ -равновесий главной характеристикой выступает их эффективность при игре с супер-эксплуататором. Или, другими словами, насколько далеко от истинного равновесия исходной игры располагается найденное ϵ -равновесие. Вполне ожидаемо, что использование слишком грубых или неадекватных абстракций дает грубое и неадекватное ϵ -равновесие. Также

вполне ожидаемо, что чем меньше информации о полной игре теряется при переходе к абстракции, тем больше сама абстракция и выше требования к алгоритмам обучения и вычислительным ресурсам.

Главным преимуществом интеллектов, построенных с применением принципа равновесных стратегий, является их низкая эксплуатируемость. В то же время, эти стратегии малоэффективны даже против достаточно слабых игроков в покер.

Разработанный в данной работе игровой интеллект не может быть отнесён ни к одному из вышеперечисленных видов. Однако, по принципу работы, он некоторым образом схож с эксплуататорами: в нём также применяются искусственные нейронные сети для обучения некоторой стратегии. Отличием является то, что в эксплуататорах ИНС применяются для обучения интеллекта стартовой стратегии, а также для построения модели поведения оппонента. В данной работе ИНС применяется для построения модели поведения самого покерного интеллекта. При этом интеллект не копирует поведение оппонента, а придерживается собственной стратегии, которая позволяет ему обыграть противника и получить наибольшую прибыль. Впрочем, сама стратегия полученного интеллекта, также не поддаётся классификации, а скорее похожа на стиль игры человека, который был выработан благодаря полученному опыту.

Таким образом, покерный интеллект, разработанный в ходе исследования, не придерживается определённой выигрышной стратегии, а скорее похож на искусственный интеллект, позволяющий имитировать поведение человека. В данном случае интеллект имитирует человека при игре в покер, но сам подход позволяет обучить его практически любой человеческой деятельностью.

1.3.2 Генетические алгоритмы

1.3.2.1 Особенности генетических алгоритмов

В настоящее время для настройки и обучения искусственных нейронных сетей все чаще применяются генетические алгоритмы [12]. С их помощью обучаются искусственные нейронные сети, адаптированные для решения трудно формализуемых задач.

ГА используют для работы эволюционные принципы наследственности, изменчивости и естественного отбора. Общая схема ГА представлена на рисунке 1.1.

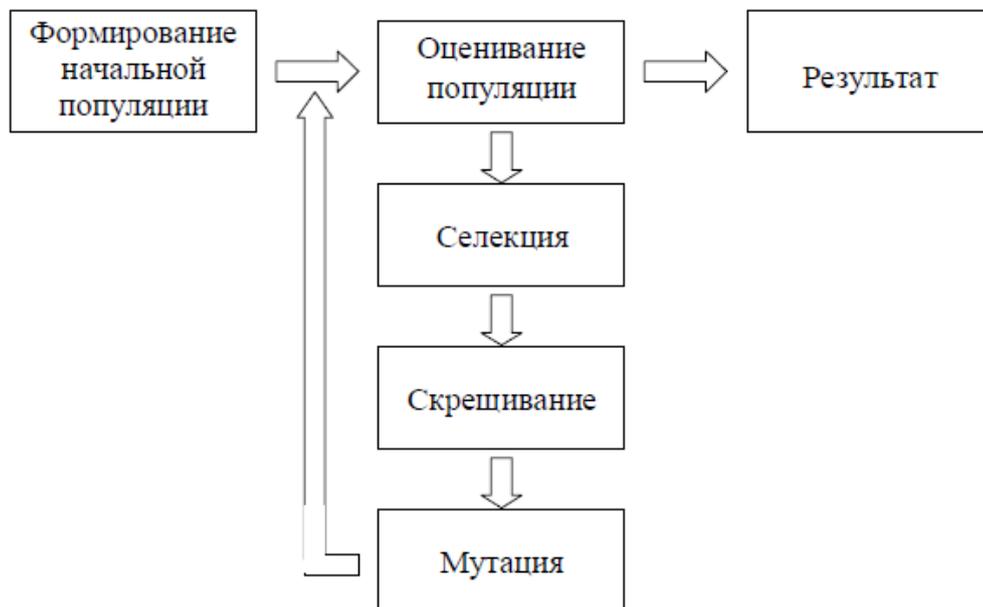


Рисунок 1.1. Общая схема генетического алгоритма.

Генетический алгоритм работает с популяцией особей, в хромосоме (генотип) каждой из которых закодировано возможное решение задачи (фенотип). В начале работы алгоритма популяция формируется случайным образом. Для того чтобы оценить качество закодированных решений используют функцию приспособленности, которая необходима для вычисления приспособленности каждой особи (блок «Оценивание популяции»). По результатам оценивания особей наиболее приспособленные из них выбираются (блок «Селекция») для скрещивания. В результате

скрещивания выбранных особей посредством применения генетического оператора кроссинговера создается потомство, генетическая информация которого формируется в результате обмена хромосомной информацией между родительскими особями (блок «Скрещивание»). Созданные потомки формируют новую популяцию, причем часть потомков мутирует (используется генетический оператор мутации), что выражается в случайном изменении их генотипов (блок «Мутация»). Этап, включающий последовательность «Оценивание популяции» – «Селекция» – «Скрещивание» – «Мутация», называется поколением. Эволюция популяции состоит из последовательности таких поколений. Длительность эволюции может определяться следующими факторами:

- нахождение решения в результате эволюционного поиска;
- ограниченность количества поколений;
- ограниченность количества вычислений функции приспособленности (целевой функции);
- вырождение популяции, когда степень разнородности хромосом в популяции становится меньше допустимого значения.

1.3.2.2 Оценивание популяций

Оценивание популяции необходимо для того, чтобы выявить в ней более приспособленные и менее приспособленные особи. Для подсчета приспособленности каждой особи используется функция приспособленности (целевая функция):

$$f_i = f(G_i),$$

где $G_i = \{ g_{ik}: k = 1, 2, \dots, N \}$ – хромосома i -й особи, g_{ik} – значение k -го гена i -й особи, N – количество генов в хромосоме.

Как правило, использование эволюционного алгоритма подразумевает решение задачи максимизации (минимизации) целевой функции, когда необходимо найти такие значения параметров функции f , при которых значение функции максимально (минимально). В соответствии с этим, если

решается задача минимизации и $f(G_i) < f(G_j)$, то считают, что i -я особь лучше j -й особи. В случае задачи максимизации, наоборот [13].

1.3.2.3 Селекция

Из рассматриваемых типов селекции были выявлены: рулеточная селекция, турнирный отбор и селекция с усечением. Наиболее пригодная для данной исследовательской работы показалась селекция с усечением.

Селекция усечением. При отборе усечением после вычисления значений приспособленности для скрещивания выбираются ln лучших особей, где l – «порог отсечения», $0 < l < 1$, n – размер популяции. Чем меньше значение l , тем сильнее давление селекции, т.е. меньше шансы на выживание у плохо приспособленных особей. Как правило, выбирают l в интервале от 0,3 до 0,7 [14].

1.3.2.4 Скрещивание и формирование нового поколения

Отобранные в результате селекции особи (называемые родительскими) скрещиваются и дают потомство. Хромосомы потомков формируются в процессе обмена генетической информацией (с применением оператора кроссинговера) между родительскими особями. Созданные таким образом потомки составляют популяцию следующего поколения.

Оператор кроссинговера (crossover operator), также называемый кроссовером, является основным генетическим оператором, за счет которого производится обмен генетическим материалом между особями. Моделирует процесс скрещивания особей.

Пусть имеются две родительские особи с хромосомами $X = \{x_i, i=1, L\}$ и $Y = \{y_i, i=1, L\}$. Случайным образом определяется точка внутри хромосомы, в которой обе хромосомы делятся на две части и обмениваются ими. Назовем эту точку точкой разрыва.

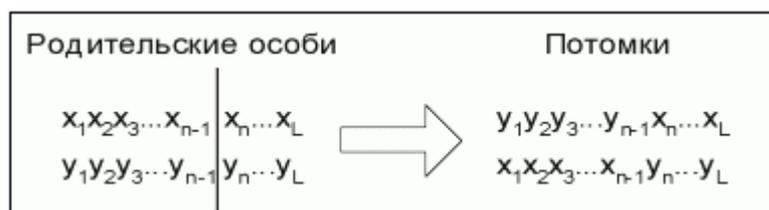


Рисунок 1.2. Пример работы одноточечного кроссинговера

Данный тип кроссинговера называется одноточечным (Рисунок 1.2), т.к. при нем родительские хромосомы разрезаются только в одной случайной точке. Также существует 2-х и n-точечный операторы кроссинговера.

Как уже упоминалось выше, в результате скрещивания создаются потомки, которые формируют популяцию следующего поколения. Отметим, что обновленная таким образом популяция не обязательно должна включать одних только особей-потомков. Пусть доля обновляемых особей равна T , $0 < T < 1$, тогда в новое поколение попадает Tn потомков, n – размер популяции, а $(1 - T)n$ особей в новой популяции являются наиболее приспособленными родительскими особями (так называемые элитные особи). Параметр T называют разрыв поколений (generation gap). Использование элитных особей позволяет увеличить скорость сходимости генетического алгоритма [15].

1.3.2.5 Мутация

Оператор мутации используется для внесения случайных изменений в хромосомы особей. Это позволяет «выбираться» из локальных экстремумов и тем самым эффективнее исследовать пространство поиска. Аналогично оператору кроссинговера, работа оператора мутации зависит от вероятности применения мутации P_M .



Рисунок 1.3. Пример работы мутации

Также как и кроссинговер, мутация проводится не только по одной случайной точке. Можно выбирать некоторое количество точек в хромосоме

для мутации, причем их число также может быть случайным. Вероятность мутации значительно меньше вероятности кроссинговера и редко превышает 2%.

2. ПРОЕКТИРОВАНИЕ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ

Данная работа направлена на изучение поведения искусственной нейронной сети в игровой среде. Чтобы отслеживать его работу и применять его решения в действии, необходимо визуальное представление данных. Так как алгоритмический интеллект взаимодействует с сервером, необходима среда, которая эмулировала бы работу искусственного интеллекта на основе интерфейса, аналогичного тому, что используется реальным пользователем. Другими словами, интеллект, так же как и человек, отделен от серверной части и общается с ним посредством команд. Менеджер игровых интеллектов будет производить обучение нетривиального алгоритма, создавать комнаты для тестирования, управлять взаимодействием игровых интеллектов. Таким образом, клиентская часть разбита на два автономных модуля:

- Представитель данных или пользовательское приложение;
- Менеджер игровых интеллектов.

2.1 Устройство пользовательского приложения

Серверная часть полностью абстрагирует все данные о клиентах и инкапсулирует их. Система команд, разработанная для взаимодействия пользовательского и серверного приложений, обеспечивает стабильную работу приложения, сводя вероятность возникновения ошибок в результате действий пользователя к минимуму. На клиентское приложения возложены только те задачи, которые связаны с визуализацией игрового процесса, и при этом не затрагивают вычислений, связанных с игровых взаимодействием. Тем не менее, передаваемые данные должны быть зашифрованы, чтобы избежать подмены перехваченных пакетов и последующей их отправки на сервер.

Визуализатор имеет пять сцен, каждая из которых выполняет свою роль:

- сцена загрузочного экрана;
- сцена авторизации;

- сцена регистрации;
- сцена общего меню;
- сцена игровой комнаты.

Все объекты, размещенные на любой из сцен, объявляются внутри классов каждой сцены. Объекты имеют координаты относительные левого верхнего угла сцены, для того, чтобы, при изменении размера игрового окна, объекты сцен меняли свое расположение и размер. Вектора объектов хранятся в конфигурационном файле и загружаются при запуске приложения.

Сцена загрузочного экрана (Рисунок 2.1) анимирует логотип графического движка Unity и логотип продукта PokerSharp. Данную сцену можно пропустить, указав флаг Logo в конфигурационном файле в папке с приложением. Появление сцены происходит автоматически, и так же происходит переход на сцену авторизации без возможности обратного перехода.



Рисунок 2.1. Сцена загрузочного экрана

Сцена авторизации (Рисунок 2.2) представляет собой окно с доступом перехода в сцену регистрации или сцену общего меню. К данной сцене прикреплены два класса: класс Authorization, отвечающий за прорисовку и

размещение объектов на сцене, и класс AuthorizationLogic, отвечающий за прослушивание входящих сообщений, переход на другие сцены, фильтр и проверку корректности ввода логина и пароля. У пользователя есть возможность сохранения логина и пароля в конфигурационном, после чего они будут загружаться автоматически. При сохранении эти данные шифруются, чтобы избежать несанкционированного доступа к чужому аккаунту.

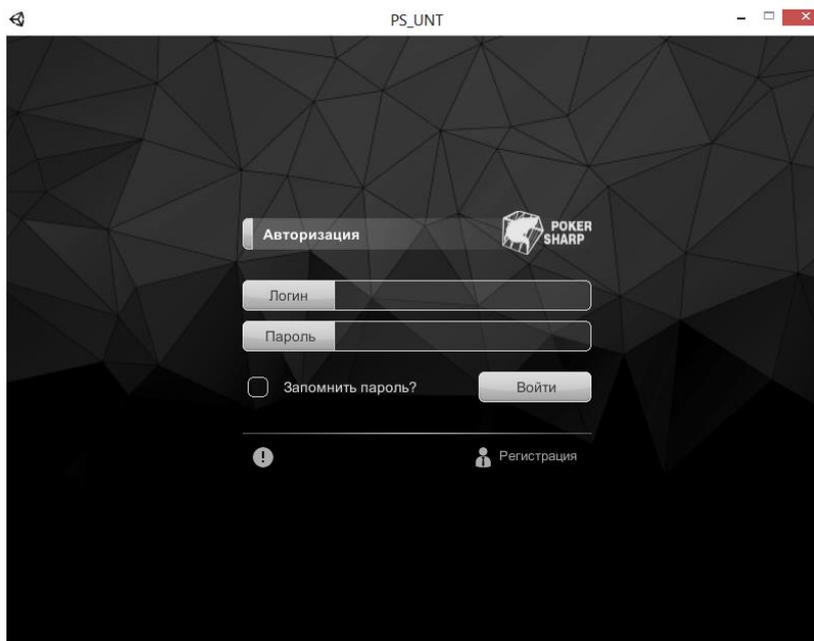


Рисунок 2.2. Сцена авторизации

Сцена регистрации (Рисунок 2.3) является окном для ввода данных пользователя. Все вводимые данные являются обязательными во избежание безвозвратной утери контроля над учетной записью пользователя. Переход на данную сцену осуществляется только со сцены авторизации с обратной возможностью перехода. К сцене прикреплены классы Registration и RegistrationLogic. Класс Registration отвечает за расположение объектов на сцене и их поведение при взаимодействии с пользователем. В классе RegistrationLogic содержатся различные фильтры и проверка корректности ввода данных с текстовых окон. Такие процедуры необходимы для уменьшения вероятности возникновения ошибок в работе клиент-серверного

приложения. С особенностями функционирования сервера можно ознакомиться в работе [16].

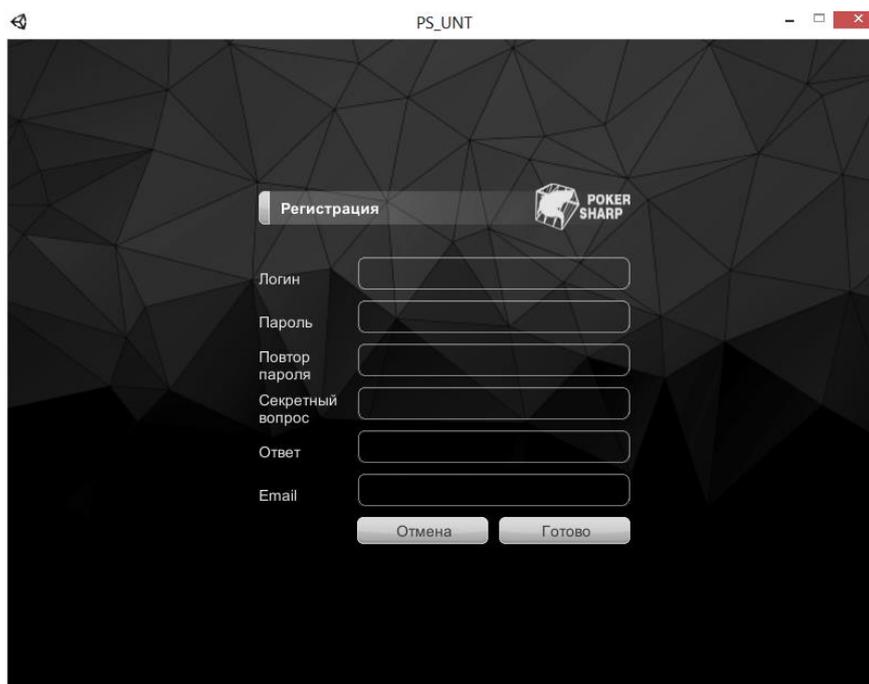


Рисунок 2.3. Сцена регистрации

Сцена общего меню (Рисунок 2.4) представляет собой динамическую сцену, имеющую несколько состояний. Первое состояние отображает список комнат, общий чат. Второе отображает параметры приложения в целом (размер окна, управление загрузочным экраном, размер карт, стиль фона). Третье состояние отображает профиль игрока, его сведения и статистику. К данной сцене прикреплены следующие классы: `Lobby`, `LobbyLogic`. Класс `Lobby` изменяет сцену в зависимости от состояния. Переход к различным сценам осуществляется с помощью переключателей на верхней панели. Класс `LobbyLogic` отвечает за прослушивание входящих сообщений и их обработку.

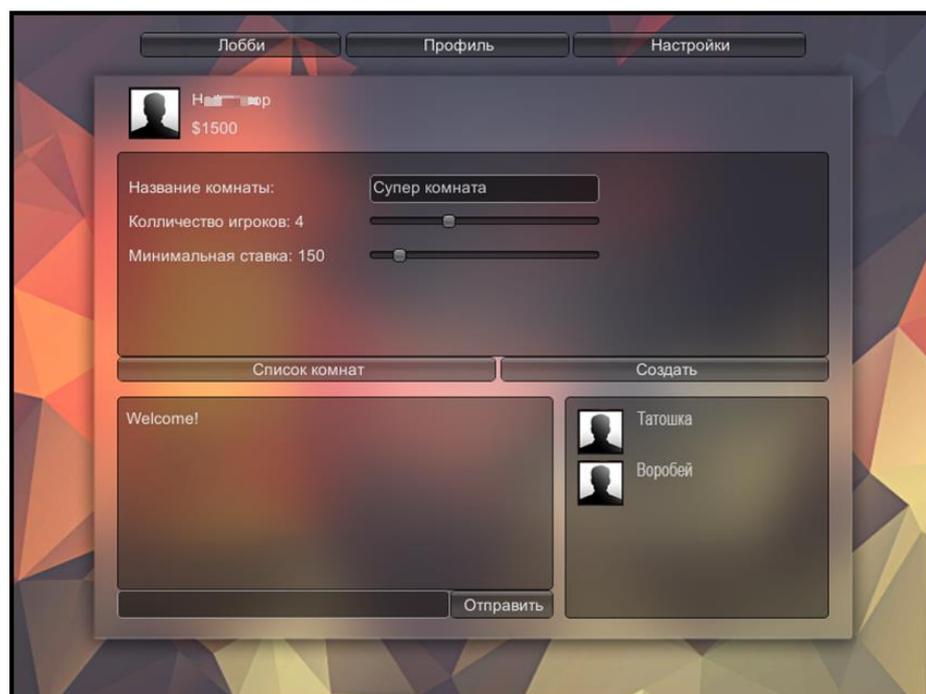


Рисунок 2.4. Сцена общего меню

Сцена игровой комнаты (Рисунок 2.5) анимирована и имеет значительно больше классов относительно других сцен, так как на сцене находятся объекты, аналогов которых нет в стандартных библиотеках, предоставляемых средствами графического «движка». Объекты на сцене загружают различные спрайты. Спрайт представляет собой изображение в формате jpeg. Игровая комната рассчитана по умолчанию на шесть игроков, но этот параметр можно изменить, соответственно комната будет перерисована. К данной сцене прикреплены такие классы как: Gamer, GamerPosition, GameRoom, GameRoomLogic. Классы Gamer и GamerPosition являются абстракциями игроков и игровых мест, на которых располагаются эти игроки. Класс GameRoom отвечает за размещение всех объектов на сцене, их поведение и взаимодействие их с пользователем, а также за анимированное поведение некоторых объектов. Управление всей логикой сцены осуществляется с помощью класса GameRoomLogic. Сцена общего меню является ключевой для эмуляции игры. На этой сцене происходит исчерпывающее визуальное представление игрового процесса. Таким образом, можно наглядно изучить поведение игроков, что крайне важно для решения задачи обучения игрового интеллекта.



Рисунок 2.5. Сцена игровой комнаты

Различные возможности переходов представлены на рисунке 2.6.

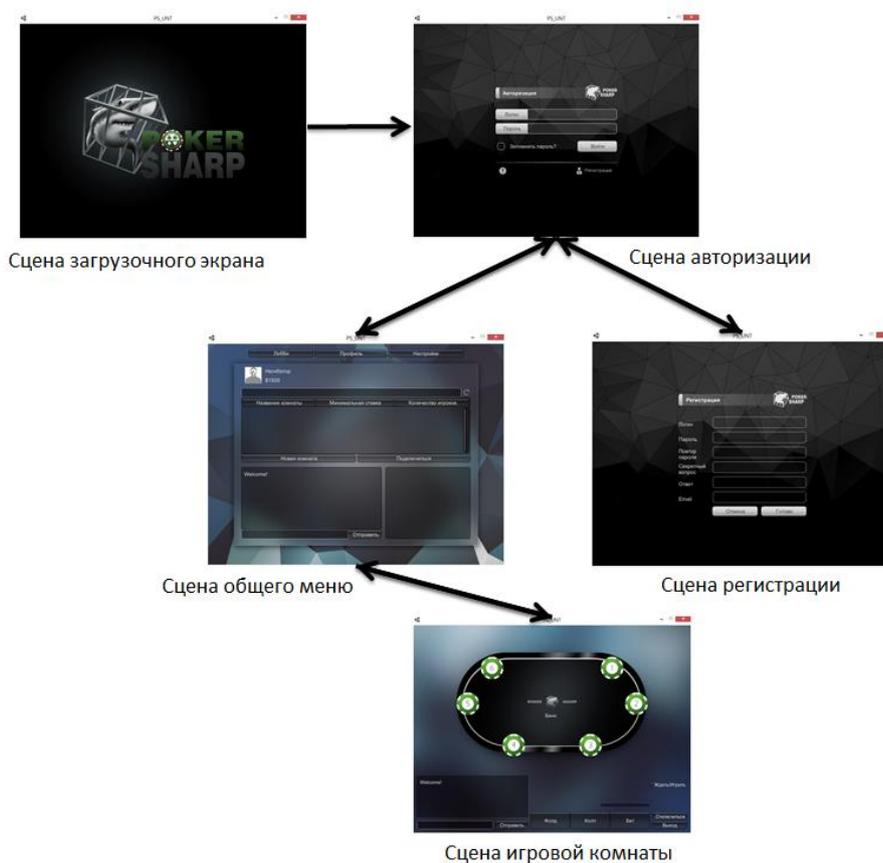


Рисунок 2.6. Схема переходов сцен

Как видно из схемы, переходы существуют практически из любого состояния, кроме состояния загрузочного экрана. При смене события, все

объекты старой сцены очищаются, и загружаются объекты новой сцены. При загрузке новой сцены объекты серверного сокета не передаются, так как они помещены в специальный класс `Global`, инициализируемый на стадии авторизации пользователя и являющийся глобальным для всех сцен.

При реализации пользовательского приложения, помимо разделения на отдельные сцены, каждая сцена была разбита на соответствующие классы, помогающие защищать пользовательские данные. У каждой сцены, в которой присутствует взаимодействие с пользователем, есть класс, содержащий методы прослушивания входящих данных и команд, обработки и передачи этих данных в класс представления. Такой подход необходим для абстрагирования и ускорения работы метода прослушивания (чем меньше в прослушивании набор обрабатываемых команд, тем быстрее работает отклик приложения в целом). Каждая сцена имеет собственный метод прослушивания. Соответственно, нужно отделять классы, отображающие данные, и классы, обрабатывающие их (шифровка/расшифровка сообщений). Взаимодействие классов организовано таким образом, что класс, обрабатывающий данные, управляет классом представления данных. Чем больше объектов для представления, тем больше абстракций используется на сцене. Часть классов, экземпляры которых отображались на сцене, использовалась из стандартного набора библиотек: `Sprite`, `GameObject`, `List`, `TextBox`. Кроме того, были разработаны классы: `Chat`, `Gamer`, `Toggle`, `Scrollbar`. Также были реализованы классы, являющиеся хранилищем данных: `Gamer`, `GamerPlace`. Поэтому методы, которые они используют, всего лишь преобразовывают информацию и делают её проще к восприятию пользователем.

Например, можно рассмотреть класс `GamerPlace`, хранящий информацию о наличии игрока, сидящего за конкретным местом, о номере места, о дилере. Игрок за местом так же имеет свои данные, которые из монотонной входящей команды разбиваются на конкретные значения: имя игрока, количество его денег, ставка и карты на руках.

Как говорилось ранее, данное приложение имеет файл для настройки, позволяющий изменять размер окна, установку/снятие загрузочного экрана, выбор различных рубашек карт, тему фона. Все изображения для спрайтов хранятся отдельно до тех пор, пока сцена не будет разрушена и загружаются по мере надобности.

2.2 Устройство менеджера интеллектов

Игровой интеллект подключается к серверу, как обычный пользователь, имея сокет, логин и пароль. Для предотвращения избыточного количества приложений с игровыми интеллектами, было решено объединить их в модуль, который является контейнером для экземпляров игровых интеллектов с возможностью управления ими. Это позволит централизованно собирать информацию и статистику о каждом экземпляре интеллекта, что, несомненно, упростит процесс обучения искусственной нейронной сети. Менеджер интеллектов – это приложение, реализующее централизованное управление процессом обучения игровых интеллектов.

В зависимости от размеров игровой комнаты, инициализируются требуемые экземпляры игровых интеллектов. Далее проходит авторизация каждого из них. При попадании в общее меню, один из экземпляров группы с подает запрос на создание комнаты. При успешном создании комнаты, вся группа экземпляров интеллекта переходит в нее и занимает соответствующие места за игровым столом. Далее все игровые интеллекты отсылают команду о готовности к игре. Так как на игровой процесс в дальнейшем менеджер интеллектов никак не влияет, его возможности и задачи заключаются в следующем: формирование групп экземпляров, добавление/удаление экземпляра в группу, формирование запросов на создание комнаты, установление флага готовности экземпляров к игре.

2.2.1 Структура менеджера игровых интеллектов

Приложение менеджера игровых интеллектов было разбито на отдельные модули:

- модуль обучения игрового интеллекта, основанный на нейронной сети;
- модуль централизованного управления экземплярами игровых интеллектов;
- модуль игровых интеллектов.

Каждый модуль в свою очередь разбивается на отдельные классы. Все классы максимально защищены друг от друга: реализована инкапсуляция данных. Диаграмма классов представлена на рисунке 2.7.

Как можно заметить, класс Program является точкой входа в программу, имеющий такие экземпляры классов, как GeneticAlgorithm и BotManager. Это приложение позволяет управлять поведением игровых интеллектов и их обучением вручную, что позволяет без дополнительного вмешательства в исходный код программы конфигурировать настройки обучаемых интеллектов.

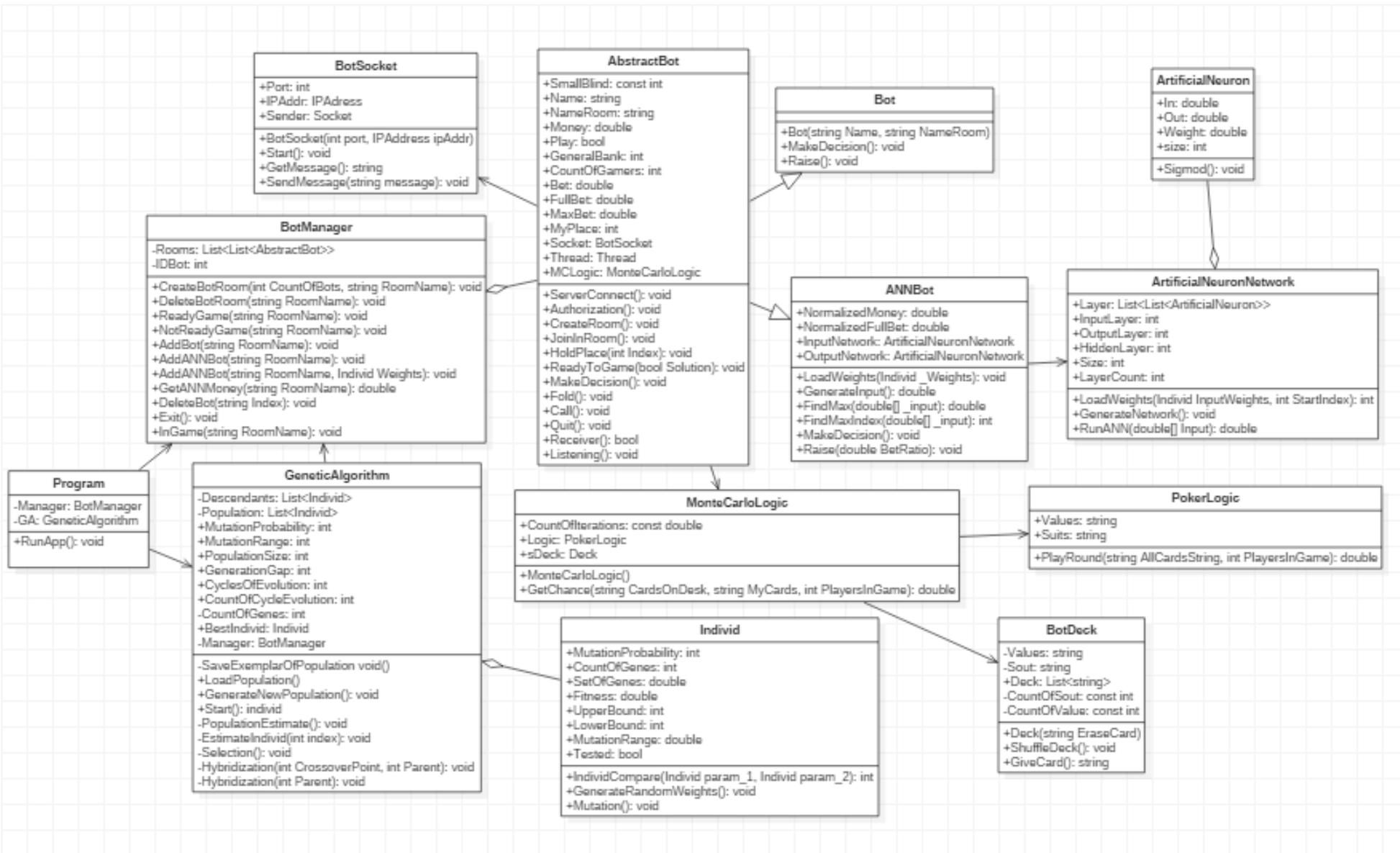


Рисунок 2.7. UML-диаграмма приложения менеджера игровых интеллектов

2.2.2 Модуль обучения игрового интеллекта, основанного на искусственной нейронной сети

Модуль обучения искусственной нейронной сети использует методы генетического алгоритма. Сам модуль состоит из следующих классов: GeneticAlgorithm и Individ. Класс Individ содержит в себе поля и методы, полностью описывающие один индивид в популяции. Индивид содержит поля: вероятность мутации, количество генов, требующихся для инициализации индивида, стартовый диапазон случайных значений при инициализации набора генов, набор самих генов, оценка индивида, влияние мутации на отдельную хромосому гена. Наиболее важным полем индивида является набор генов, представляющий собой набор весов для нейронной сети. Этот модуль является «учителем» для игрового интеллекта, основанного на искусственной нейронной сети. Соответственно, игровой алгоритм использует полученные эволюционным алгоритмом настройки для принятия какого-либо решения.

Класс GeneticAlgorithm реализует методы генетического алгоритма. У класса генетического алгоритма есть важные свойства: вероятность мутации для генов индивидов, диапазон изменения генов при мутировании, размер популяции, разрыв поколений, количество циклов обучения. Все поля при инициализации являются ключевыми параметрами, которые загружаются отдельно из файла и могут быть изменены по требованию эксперта. По мере обучения индивиды, прошедшие оценивание, сохраняются, для того, чтобы можно было прерывать этап обучения и через некоторое время возобновлять его без потери информации о популяции.

Процесс обучения индивида начинается с оценки приспособленности каждого индивида в популяции. Для проведения этой процедуры используется количество оставшихся денежных средств на балансе экземпляра игрового алгоритма. Чем больше средств, тем лучше особь. Для этого экземпляр проводит по несколько игр с различным количеством

соперников. После оценки всей популяции происходит селекция, скрещивание и мутация отдельных индивидов. Эти этапы реализованы в методах: PopulationEstimate, EstimateIndivid, Selection, Hybridization, Mutation.

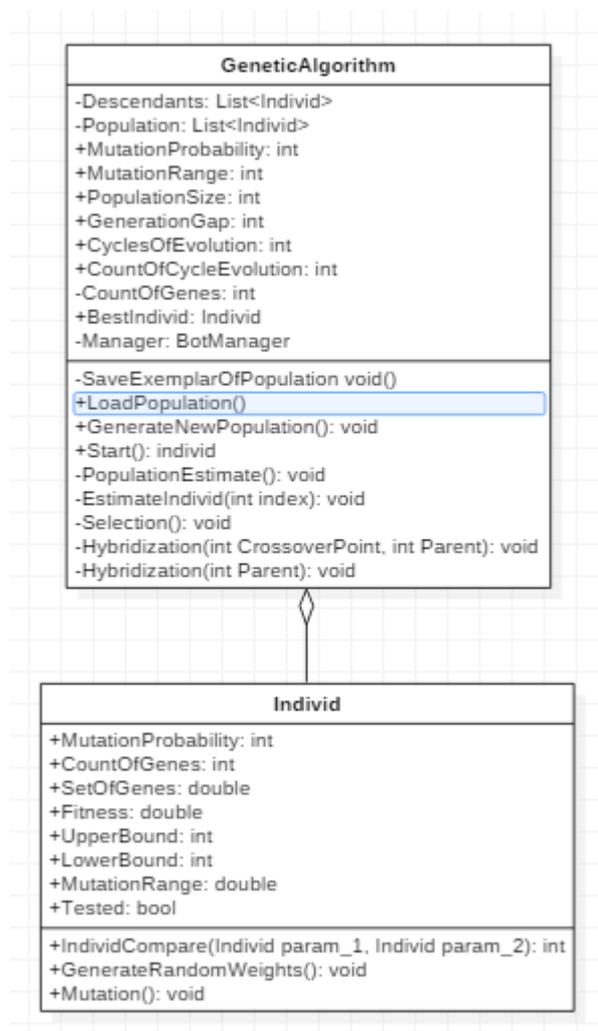


Рисунок 2.8. UML-диаграмма модуля обучения игрового интеллекта основанного искусственной нейронной сети

2.2.3 Модуль централизованного управления экземплярами игровых интеллектов

Данный модуль предназначен для управления экземплярами игрового интеллекта, моделирующими поведение человека. Соответственно, этот модуль является центром управления группами объектов игровых интеллектов. Ключевыми задачами являются: инициализация экземпляров, формирование групп и имитирование поведения объектов таким образом, чтобы сервер не различал людей и игровые интеллекты.

Имитация заключается в обработке команд, запрашивающих авторизацию, создание комнаты, удаление комнаты, вход и выход из комнаты. Все эти методы реализованы в родительском классе игрового интеллекта. Отдельные элементы не знают о существовании друг друга, поэтому нужен контейнер, который бы мог централизованно управлять, как отдельными экземплярами, так и их группами. Так как управлять комнатой может только ее создатель, соответственно необходимо знать, кто ее владелец.

Основные методы: `CreateBotRoom`, `DeleteBotRoom`, `ReadyGame`, `NotreadyGame`, `AddBot`, `AddANNBot`, `DeleteBot`, `InGame`. Метод `CreateBotRoom` создает комнату с определённым названием, на определенное количество игроков и рассаживает игровые интеллекты по местам автоматически. `DeleteBotRoom` выводит все экземпляры игровых объектов из комнаты, очищает их и отправляет запрос на удаление данной комнаты. Методы `ReadyGame` и `NotreadyGame` приводят к готовности игровые алгоритмы и запускают или останавливают игровой процесс. После создания комнаты можно добавить или убрать объект игрового алгоритма. Добавление осуществляется как в процессе игры, так и к началу следующего раунда. Существует несколько видов игровых алгоритмов. При добавлении интеллектов в игровую комнату указывается необходимый тип. Удаление играющего объекта осуществляется путём отправки удаляемому игроку

команды NotreadyGame, после чего он удаляется его из игры. Функция InGame проверяет состояние игровых комнат и собирает информацию, необходимую для генетического алгоритма.

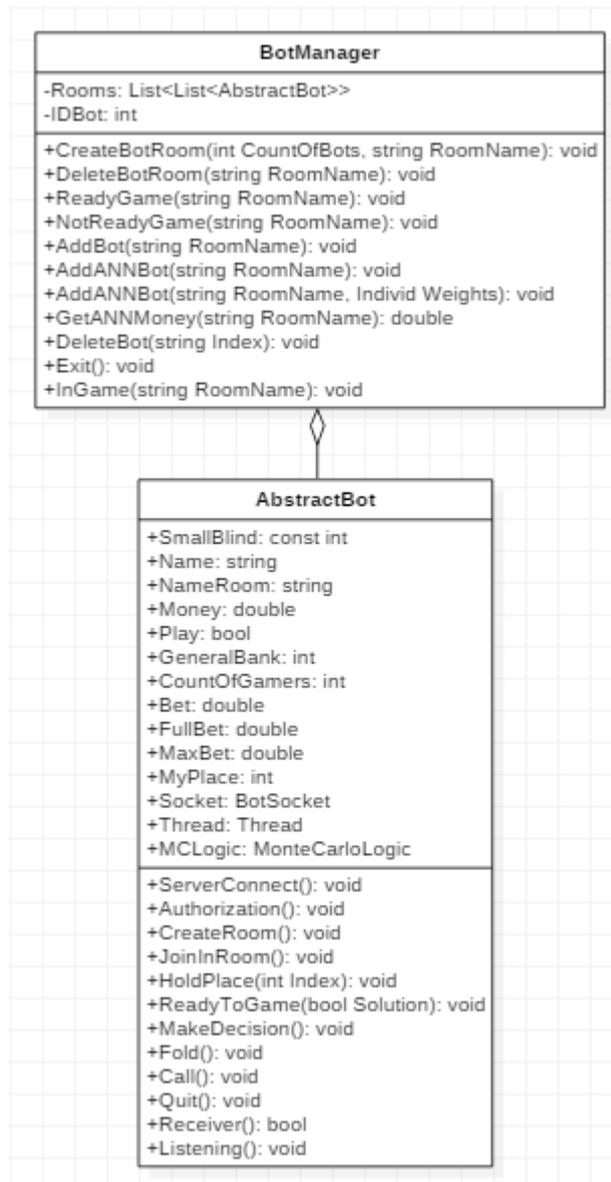


Рисунок 2.9. UML-диаграмма модуля централизованного управления экземплярами игровых интеллектов

2.2.4 Модуль игровых интеллектов

Как говорилось ранее, каждый индивид подвергается процессу оценки. Для того чтобы оценить конкретную особь, нужно провести некоторое количество игр с игровым интеллектом, который принимает решения согласно примитивной покерной стратегии. Таким образом, модуль игрового интеллекта разделяется на две части: игровой интеллект, основанный на искусственной нейронной сети и игровой интеллект принимающий решение по определенному закону. Так как два этих модуля будут иметь множество общих методов, следует реализовать абстрактный игровой алгоритм, являющийся родительским для этих подмодулей. Наследуемые классы будут различаться лишь методами принятия решения для конкретной ситуации.

Абстрактный игровой интеллект имеет поля: имя, имя комнаты, в которой находится экземпляр, количество средств на счету, флаг состояния, текущая ставка, полная ставка, максимальная ставка, номер места за игровым столом и сокет. Необходимо чтобы экземпляры этого класса общались с сервером посредством установленной системы команд и запоминали свое текущее состояние на сервере.

Оба наследуемых алгоритма используют численный метод Монте-Карло, для подсчета вероятности выигрыша. Метод Монте-Карло предполагает получение большого числа реализаций стохастического процесса, который формируется таким образом, чтобы его вероятностные характеристики совпадали с аналогичными величинами решаемой задачи.

В данном случае случайной величиной является выигрыш или поражение игрока при известном наборе карт. При этом все карты, которые также участвуют в эксперименте, но их значения достоверно не известны, выбираются из оставшейся колоды случайным образом. Из общего числа экспериментов запоминается число тех случаев, когда игрок победил. Это число делится на общее число проведённых тестов, что в результате даёт вероятность выигрыша.

Наследуемые алгоритмы различаются в подходах принятия решения.

Игровой алгоритм, принимающий решение по формуле опирается только на процент выигрышных игр полученных при вычислении метода Монте-Карло:

$$\text{Выигрыш} = \text{Доля выигранных игр} * \text{банк}$$

Если выигрыш меньше текущей полной ставки, принимается решение сбросить карты. Если выигрыш больше текущей полной ставки не более чем на большой блайнд, будет решено поддержать эту ставку. В другом случае принимается решение поднять ставку на большой блайнд.

Принятие решений и структура игрового алгоритма, основанного на искусственной нейронной сети, базируется на изучении статьи [17].

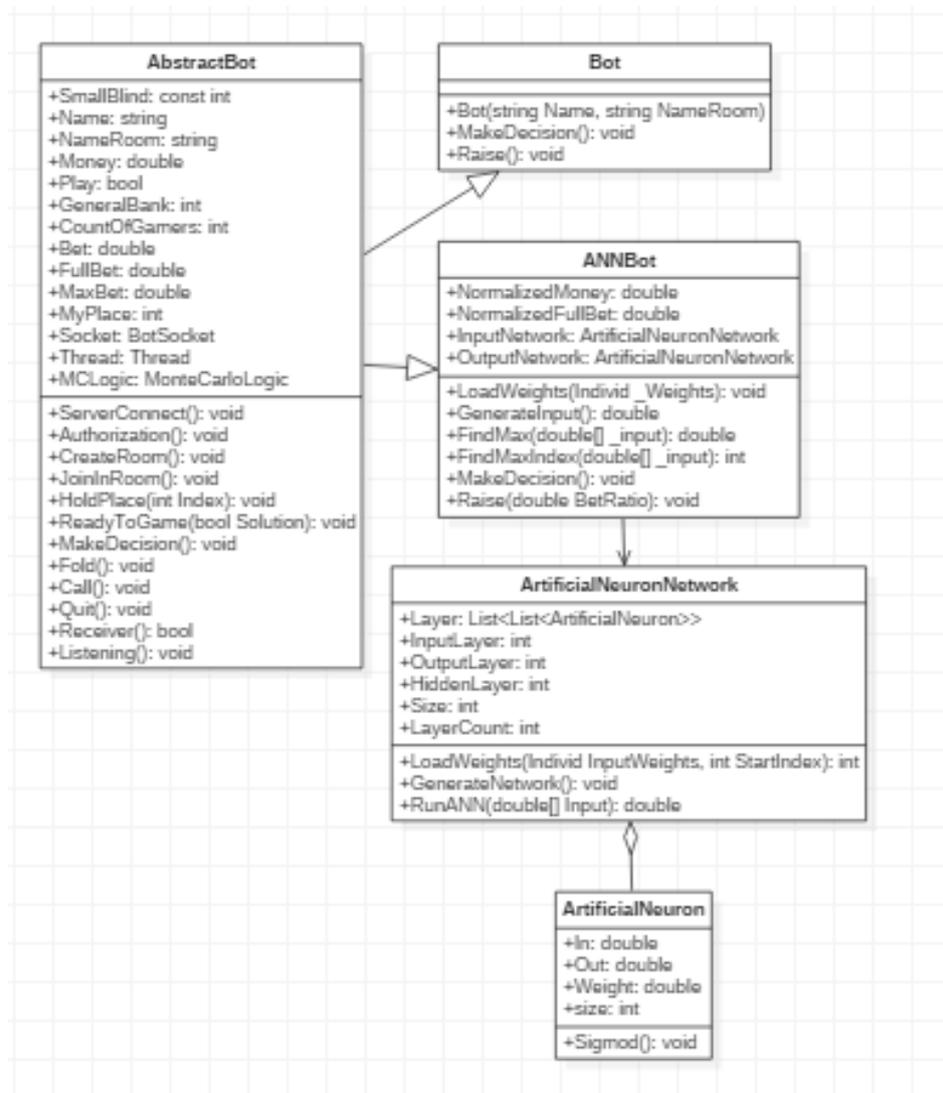


Рисунок 2.10. UML-диаграмма модуля игровых интеллектов

3. ОПИСАНИЕ ПРЕДЛАГАЕМЫХ РЕШЕНИЙ

Эволюционный алгоритм оптимизации является алгоритмом случайно-направленного поиска [18] и применяется в основном там, где сформулировать задачу в пригодном виде для быстрых алгоритмов невозможно. Применение генетического алгоритма необходимо для обучения нейронных сетей, вычисляющих сложные функции. Главное достоинство этого метода заключается в подходе «от общего к частному». При начальном этапе эволюции изменяется значительная часть пространства весов. Через некоторое количество итераций генетический алгоритм будет сужать пространство поиска, тем самым увеличивает приспособленность особи. Так же важным достоинством является большой набор параметров, влияющих на качество и скорость обучения искусственных нейронных сетей.

На сегодняшний день уже существует алгоритм, играющий в игру «Покер» с максимальной долей выигрыша, но игра может проходить только с лимитированными ставками и только один на один. Для поставленной задачи требуется обучить нейронную сеть, которая сможет играть в игру «Покер» с разным количеством игроков. Важно подобрать такие параметры генетического алгоритма, при которых поиск выдаст наиболее приспособленную особь за кратчайшее время.

3.1 Кодирование особи генетического алгоритма

Как известно, генетический алгоритм работает с популяциями, которые в свою очередь состоят из особей. В этой работе генетический алгоритм используется для обучения нейронной сети. Обучение искусственной нейронной сети заключается в настройке весов связей между нейронами. Таким образом, в качестве индивида в популяции выступает набор весов, записанный последовательно.

3.2 Оценивание особей в популяции

Было решено, что функцией приспособленности будет разница между начальным и конечным количеством денежных средств. Для того, чтобы

найти требуемую разницу, необходимо провести некоторое количество игр с разным количеством соперников, что требует значительного времени вычислений. Время вычисления в рассматриваемом случае сократить невозможно по причине моделирования случайных событий методом Монте-Карло. Игровой алгоритм, основанный на искусственной нейронной сети, использует возможную долю победы для принятия решения. Соответственно, чем меньше моделируется случайных событий, тем больше погрешность принятого решения.

3.3 Оператор кроссинговера

При скрещивании было решено использовать разные операторы кроссинговера: одноточечный и двухточечный кроссинговер. При исследовании влияния этих параметров будет оцениваться качество и скорость обучения нейронной сети. Операторы имеют разную разрушающую способность, следовательно, они имеют разную степень влияния на формирование новой популяции.

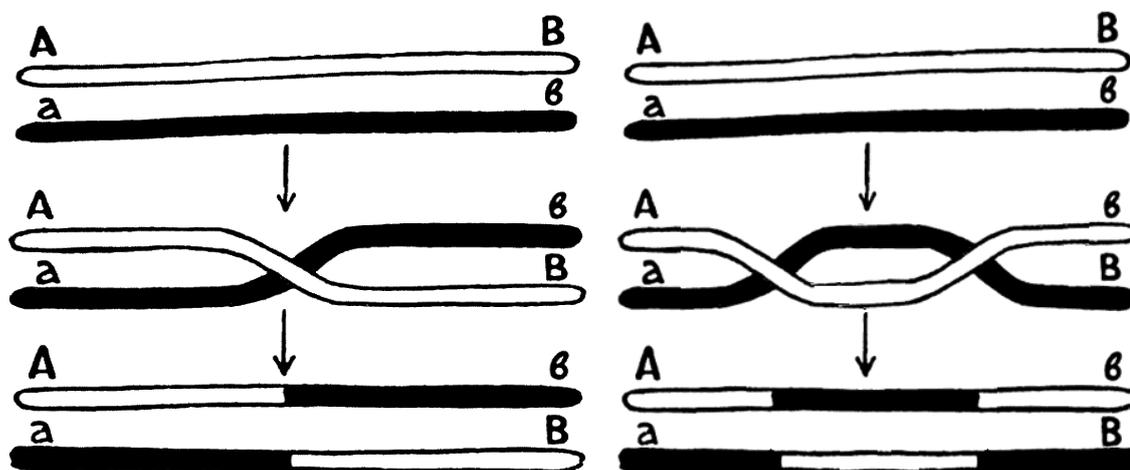


Рисунок 14. Операторы кроссинговера

Также можно было бы использовать BLX- α или арифметический оператор. Особенность этих операторов заключается в том, что они позволяют формировать гены в генотипе потомка, которые отличаются на некоторую величину от тех генов, которые были в генотипах родителей. Такой подход формирования новой особи увеличит пространство поля

поиска, что выгодно отличает его от операторов кроссинговера. Однако BLX- α , или арифметический операторы, вычислительно более сложные, т. е. на формирование новой популяции они тратят больше времени, чем более простые операторы кроссинговера. Учитывая, что в концепции генетического алгоритма присутствует оператор мутации, основной задачей которого является расширения пространства поля поиска оптимальных решений, использование вычислительно сложных операторов кроссинговера не целесообразно.

3.4 Оператор мутации

После скрещивания в новой популяции у каждой особи есть шанс мутации каждого гена в хромосоме. Оператор мутации расширяет область поиска наилучшей особи. Ключевыми параметрами оператора являются сила и шанс мутации. Сила мутации влияет на то, насколько сильным будет изменение конкретного гена. Чем больше сила, тем больше будет модифицирован ген. Шанс мутации необходимо подобрать так, чтобы он оптимально влиял на изменение популяции в целом.

В практике эксплуатации генетических алгоритмов часто возникает проблема вырождения популяции, когда вся популяция сходится к одной особи. Существуют два метода решения этой проблемы. Первый - увеличение шанса мутации, что может сильно повлиять на начальном этапе эволюции. Второй - при вырождении популяции, добавлять новых особей, тем самым не влияя на особей при начальном этапе эволюции.

4. ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Ключевой целью этой научно-исследовательской работы является получение качественного экземпляра игрового интеллекта, имитирующего поведение игрока при игре в покер. Особенностью такого игрового интеллекта должно стать умение приспосабливаться к различным игровым ситуациям и разному количеству игроков за столом.

Одним из входных параметров обоих нейронных каскадов является величина, показывающая долю выигрыша, полученную из набора случайных величин. На вычисление такого параметра уходит достаточно продолжительный промежуток времени. Соответственно, с учётом затрат времени на определение вероятности выигрыша, игровой интеллект имеет гораздо меньший отрезок времени для принятия решения, чем обычный игрок. Как известно, особью для генетического алгоритма будет набор весов каскадов нейронных сетей, каждая из которых решает свою задачу. Поэтому структура этих каскадов может отличаться друг от друга.

Для того чтобы определить наиболее оптимальное количество скрытых слоев и количество нейронов для них, необходимо указать диапазон стартовых значений при инициализации каждой особи. Слишком большой диапазон может усугубить обучение путем добавления экземпляров, неспособных к приспособлению или неспособных вычислить результат за определенное время. Соответственно, для каждой структуры нейронной сети необходимо опытным путем подобрать оптимальный диапазон стартовых значений.

Помимо этого у генетического алгоритма также есть важные параметры, влияющие на скорость и качество получения лучшей особи: операторы кроссинговера и мутации, разрыв поколений, сила мутации. Невозможно исследовать эти параметры по отдельности, так как при изменении хотя бы одного из них, можно получить радикальные изменения. Для решения такой задачи было принято рассмотреть наиболее интересные

наборы параметров генетического алгоритма и параметры каскадов нейронных сетей, проходящие эволюционное обучение. Тем не менее, рассмотреть все варианты не представляется возможным, поэтому некоторые из параметров устанавливались с помощью использования теоретических обоснований руководителей проекта.

4.1 Описание подхода к обучению каскадов нейронной сети

Для того чтобы эффективно подобрать параметры обучения, необходимо провести ряд опытных экспериментов с разными наборами параметров, отличавшихся друг от друга. В результате было подобрано 10 наборов с параметрами, которые представлены в таблице 4.1.

Структура каскадов может отличаться друг от друга, так как они выполняют разные задачи. Обучение необходимо проводить комплексное, потому что разбиение его на части может не дать эффективного результата, удовлетворяющего требованиям данной исследовательской работы. Так как задача имеет большую вычислительную сложность, эволюционное обучение необходимо ограничить в количестве исполняемых итераций. Каждый набор проходил 1000 итераций эволюционного обучения, что позволяет исследовать несколько наборов параметров за определенное количество времени. Вследствие чего был получен ряд показателей эффективностей каждого набора и были определены лучшие индивиды каждого из них. Особей в популяции было 100 для всех наборов параметров.

Таблица 4.1 – Параметры выбранных интеллектов

№ набора параметров	Количество скрытых слоёв первого каскада	Количество скрытых слоёв второго каскада	Количество нейронов первого каскада	Количество нейронов второго каскада	Разрыв поколений, (%)	Шанс мутации, (%)	Сила мутации
1	1	2	600	600;1200	20	5	0,05
2	1	3	1000	400;600;800	20	2	0,1
3	2	2	400;550	700;1000	10	1	0,1
4	2	2	600;500	400;500	15	5	0,01
5	2	3	200;600	400;600;800	10	2	0,1
6	2	3	800;300	600;1000;300	15	2	0,05
7	3	2	500;500;500	800;800	25	5	0,1
8	3	2	250;500;750	400;800	10	2	0,03
9	3	3	750;500;250	200;500;800	10	1	0,1
10	3	3	125;500;1000	250;500;1000	20	3	0,07

4.2 Результаты экспериментов

Все игровые экземпляры, полученные в ходе обучения, оценивались по разнице денежных средств до начала тестирования и после. Каждая особь, полученная в ходе эволюции, проходила оценивание путем проведения игр с различным количеством соперников: один, три, пять. Так же при проведении только лишь одной серии игр можно получить ошибочный результат. Так как в игре многое зависит от случайных величин, найти один игровой алгоритм с высокой функцией приспособления будет не совсем корректно. Экземпляр может получить высокий показатель приспособленности за счет везения, поэтому имеет смысл на одних наборах параметров проводить несколько серий игр с разным количеством соперников и усреднять значение разницы средств, полученных в ходе тестирования.

Таблица 4.2 – Результаты обучения интеллектов

№ интеллекта	Показатель приспособленности	Разница между текущим и минимальным значением
1	-985	0
2	-852	133
4	-546	439
7	-468	517
9	-264	721
6	12	997
8	125	1110
3	534	1519
10	1450	2435
5	2120	3105

В теории максимальный выигрыш, который мог получить интеллект, на этапе оценивания, составляет 33750. Минимальное количество средств, оставшееся после тестирования, может составлять -11250. В общем, при оценивании проходило 15 игр: 5 игр с одним противником, 5 игр с тремя противниками, 5 игр с пятью противниками.

Судя по результатам, те решения, показатель приспособленности которых менее -500 либо не способны к обучению за такое количество итераций, либо не способны обучаться вовсе. Вариант, что такому решению ни разу не выпал шанс уменьшить потери средств исключается ввиду достаточно большой серии проведенных тестов.

Об экземплярах, которые обучились не проигрывать свой стартовый капитал, можно судить, что структура нейронной сети и параметры обучения выбраны успешно.

Результаты, представленные в таблице, показывают лишь приближенные результаты получившихся игровых экземпляров. Так как количество игр в ходе оценивания было невелико, есть шанс, что некоторым экземплярам повезло, и они получили высокую оценку приспособленности. Тем не менее, даже на 1000 итерациях уже были получены результаты имитирующие поведение игрока в покер не хуже линейного алгоритма. Это говорит о том, что если расширить диапазон параметров обучения и увеличить количество итераций эволюции, можно получить более эффективное решение.

Из полученных результатов следует вывод: слишком простая структура сети не способна к решению такого рода задач. Также можно отметить, что более сложные структуры каскадов с большим количеством нейронов справляются с решением такой проблемы наиболее эффективно.

5. МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

5.1 Введение

На сегодняшний день каждое научное исследование необходимо оценивать с различных точек зрения, одной из которых является коммерческая ценность разработки. Каждый проект имеет потребность в финансировании для коммерциализации и изучения полученных результатов. Это важно самому разработчику для представления перспектив проекта.

Проект «PokerSharp» является многопользовательским клиент-серверным приложением, особенность которого заключается в игровом интеллекте, основанном на искусственной нейронной сети, способному играть на высоком уровне с различным количеством соперников. Отсюда можно сделать вывод о том, что проект, в своем роде, является уникальным, так как на текущий момент не существует подобных алгоритмов.

В игровом приложении реализована самая популярная разновидность покера - безлимитный тexasский холдем. Выбор такой разновидности был направлен на привлечение большой публики с возможностью конкурирования с устоявшимися проектами. Приложение является компьютерной игрой, что, несомненно, является положительной чертой, в связи с тем что аудитория увлекающаяся компьютерными играми огромна. Так же в приложении реализован такой механизм, как нейронная сеть, способная имитировать поведение игрока в покер. Такая составляющая может привлечь аудиторию увлекающейся научными исследованиями такого направления как: нейронные сети, эволюционные алгоритмы и все производные от них.

В текущем разделе основная цель является проектирование и создание конкурентоспособных технологий, соответствующих ресурсоэффективности и ресурсосбережению на сегодняшний день. Достижение такой цели можно с

помощью оценки коммерческой перспективности и потенциала проведения научного исследования, используя инструменты и расчеты SWOT-анализа.

5.2 SWOT-анализ

SWOT - Strengths (наличие сильных сторон), Weaknesses (наличие слабых сторон), Opportunities (возможности) и Threats (угрозы) — это комплексный анализ научно-исследовательского проекта, где сильные и слабые стороны определяют то насколько проект приспособлен сам влиять на факторы внутренней среды или на самого себя, а возможности и угрозы, которые могут повлиять на проект из вне. Внешними факторами могут быть законы о торговле или рыночная структура. Объектами SWOT-анализа может быть что угодно: научная сфера, организации, отдельные специалисты. SWOT-анализ осуществляется за несколько этапов.

Основной задачей первого этапа является определение наличия слабых и сильных сторон научно-исследовательского проекта и их описание. Так же необходимо определить возможности и угрозы в процессе реализации, которые могут проявиться в виде факторов внешней среды. Чтобы работать с таким инструментом, необходимо владеть определенными знаниями понятий.

1. Сильные стороны. Сильными сторонами называется набор факторов, определяющие наличие конкурентоспособных сторон научно-исследовательского проекта. Уникальность проекта с точки зрения конкурентоспособности выделяет его среди других, что определенно является сильной стороной и у него есть отличительные преимущества. Соответственно, сильными сторонами можно назвать все ресурсы и возможности, которые имеет руководство научного исследования и могут эффективно использовать их при достижении определенно поставленных целей. Необходимо при описании сильных сторон учитывать их с различных точек зрения людей, кто в нем участвует: руководители проекта и их подчиненные.

Вопросы, на которые должны отвечать поставленные аргументы:

- Какие технические преимущества вы имеете по сравнению с конкурентами?
- Что участники вашего проекта умеют делать лучше всех?
- Насколько ваш проект близок к завершению по сравнению с конкурентами?

2. Слабые стороны. Именно эти стороны определяют внутренние факторы, негативно влияющие на успеваемость научного проекта и ограничивающие его в развитии. Такие факторы могут ограничивать скорость достижения определенных целей. Ими могут быть: недостаточность ресурсов, плохая мотивация участников, нехватка знаний по сравнению с его конкурентами. Соответственно, слабые стороны необходимо учитывать при сравнении с конкурентами, а точнее с их превосходством. Чтобы проанализировать недостатки, они должны отвечать на такие вопросы:

- Что можно улучшить?
- Что делается плохо?
- Чего следует избегать?

3. Возможности. Возможности это внешние факторы, которые могут положительно повлиять на проект в любой момент времени. Такие факторы как изменение рыночной ситуации или отток аудитории, изменившие свои потребности от проектов соответствующего типа. Все эти возможности влияют на достижение поставленных целей и укрепление конкурентоспособности. Для определения возможностей требуется ответить на набор вопросов, помогающих в их формулировании:

- В чем состоят благоприятные рыночные возможности?
- Какие интересные тенденции отмечены?
- Какие потребности, пожелания имеются у покупателя, но не удовлетворяются конкурентами?
- Какие перспективы возможны на рынке?

Необходимо провести поиск благоприятных ниш, учитывая тот фактор, что их доступность времена. Другими словами, нужно занимать на рынке такие места, чтобы иметь преимущество, как можно дольше и прогнозировать свое существование наперед. Факторы, влияющие на успех проекта, могут быть такими:

- изменения в технологической сфере и на рынке - как мирового, так и регионального масштаба;
- изменения правительственной политики в отношении отрасли, где проводится научное исследование;
- изменения социальных стандартов, профиля населения, стиля жизни и т.д.

4. Угроза. Возможности это определение благоприятных внешних факторов влияющих на успех научно-исследовательского проекта, соответственно угрозы это обратная сторона, представляющая нежелательные ситуации, тенденции или изменения внешних факторов. Такие факторы замедляют или вовсе останавливают развитие проекта, впоследствии разрушающие его, негативно влияют на конкурентоспособность. Соответственно, все угрозы имеют последствие, которые наносят вред проекту. Чтобы не допустить этого, нужно проанализировать их, а для этого необходимо ответить на соответствующие вопросы:

- Какие вы видите тенденции, которые могут уничтожить ваш научно-исследовательский проект или сделать его результаты устаревшими?
- Что делают конкуренты?
- Какие внешние факторы, препятствующие развитию, стоят перед вашим проектом?
- Изменяются ли требуемые спецификации или стандарты на результаты научного исследования?
- Угрожает ли изменение технологии положению вашего проекта?

- Имеются ли у руководства проекта проблемы с материально-техническим обеспечением?

В ходе исследования и реализации продукта было выявлено ряд сильных сторон.

Во-первых, при реализации клиенткой части приложения был выбран сильный графический движок, который соответствует всем техническим параметрам современного оборудования. В разработке учитывались все оптимизации для улучшения качества взаимодействия всех систем. В этом приложении оптимизации проходили на низком уровне, что подчеркивает эффективность использования ресурсов без лишних простоев. Уже из этого фактора вытекают другие положительные стороны усиливающие проект относительно других конкурентов.

Во-вторых, серверная часть реализована таким образом, что не ограничивает проект его в масштабируемости, т. е. количество обрабатываемых пользователей на сервере может быть неограниченное количество. При этом данная система автоматизирована так, что при добавлении вычислительных средств, не требуется изменение конфигурационных файлов, а тем более исходного кода приложения.

В-третьих, приложение «PokerSharp» находится в свободном доступе, что означает отсутствия изъятия каких-либо средств при скачивании и эксплуатации. Но для необходимости в финансировании есть встроенные покупки, которые пользователь может не использовать.

В-четвертых, как уже описывалось ранее, проект имеет ключевую особенность в игровом алгоритме имитирующего поведение игрока. При правильной настройке этого алгоритма, он может учитывать даже настроение противников. Это может привлечь достаточно большую аудиторию увлекающуюся не только играми, но и наукой.

Еще одной ключевой сильной стороной проекта является его защищенность. При передаче данных используются только зашифрованные пакеты. Все конфигурационные файлы, так же имеют алгоритмы

шифрования. Для защиты сервера, используются технологии, оберегающие его от чрезмерных запросов от злоумышленников.

Несомненно, как с сильными сторонами у любого проекта есть и слабые стороны.

Одним из недостатков является ограниченность разработчиков, что замедляет процесс развития проекта. При малом количестве участников, самим участникам, в таком масштабном проекте сначала приходится изучать большой объем информации, для его применения. Это также влияет и на малое количество взглядов при решении технических проблем, ограничивая в различных путях для решения.

Для исследования и обучения экземпляра игрового алгоритма требуется очень большой ресурс времени. Помимо этого сам алгоритм нахождения оптимального решения имеет случайную вероятность, что может привести к сильному замедлению в развитии проекта.

Неопытность самих разработчиков так же является слабой стороной проекта. Отсюда возникает и неуверенность в самом проекте по причине случайных событий (таких как описывалось выше: неопределенность в нахождении оптимального решения). Поэтому приходится привязываться к существующим проектам, рассматривать их решения, для того, чтобы подстроится под них и только потом улучшать свои методы.

Отсутствие финансирования у проекта так же имеет замедлительный характер. Финансирование влияет на ускоренное обучение эволюционного алгоритма на высокотехнологичных вычислительных системах, которые не находятся в свободном доступе. Так же говорилось о встроенных покупках, которые не востребованы у пользователей приложения.

При отсутствии лицензионного программного обеспечения есть ограничения для выпуска конечного продукта на различных платформах

операционных систем, что ограничивает приложение в соответствующей аудитории.

Если говорить о возможностях научно-исследовательского проекта, то можно выявить благоприятные черты внешних факторов.

Главной и ключевой особенностью относительно конкурентов является привлечение большей аудиторией с другими интересами, помимо тех, что есть у аудитории конкурентов. Где уникальность проекта является внутренним фактором, влияющая на внешний фактор.

Еще одним не маловажно фактором играет относительно высокая цена встроенных игровых покупок у конкурентов, а так же невозможность из-за этого управлять всем игровым функционалом. Другими словами наше приложение более доступно, относительно других конкурентов.

Помимо того что было перечислено, приложение «PokerSharp» имеет более скромные системные требования, при большей эффективности. Что предоставляет появление дополнительного спроса.

Угрозами для проекта может быть:

Несвоевременное финансирование может сыграть неблагоприятную роль. При недостаточном наличии средств исследование проекта может замедлиться или вообще приостановиться.

Отсутствие рекламы тоже играет важную роль, так как даже при уникальности о проекте могут просто не знать.

И последнее, в настоящее время серверная часть проекта находится на сервере кафедры ВТ, который в свою очередь, дает сбои.

Рекомендуется результаты первого этапа SWOT-анализа представлять в табличной форме.

Таблица 5.1 – Матрица SWOT

	<p>Сильные стороны научно-исследовательского проекта:</p> <p>С1. Максимальная оптимизация проекта.</p> <p>С2. Простая масштабируемость серверной площадки.</p> <p>С3. Свободный доступ.</p> <p>С4. Уникальный игровой алгоритм, имитирующий поведение игрока в покер.</p> <p>С5. Максимальные способы защиты проекта от злоумышленников и прочего программного обеспечения.</p>	<p>Слабые стороны научно-исследовательского проекта:</p> <p>Сл1. Малый штат сотрудников.</p> <p>Сл2. Неизвестное время предположительного исследования.</p> <p>Сл3. Малый опыт у разработчиков проекта.</p> <p>Сл4. Отсутствие финансирования проекта.</p> <p>Сл5. Отсутствие лицензионного соглашения программного обеспечения.</p>
<p>Возможности:</p> <p>В1. Более востребованный спрос у аудитории, чем у конкурентов.</p> <p>В2. Относительно высокая стоимость</p>	<p>В1В2С3. При низкой привязанности платного игрового контента к игровому взаимодействию, можно привлечь значительную часть аудитории.</p>	<p>В1Сл2. При понижении числа пользователей, может увеличиться время исследования, так как при обучения алгоритмов, так же</p>

<p>продукта конкурента. ВЗ. Более высокие технические требования у приложений конкурентов, при соответствующей производительности.</p>	<p>В1В2С4. Уникальные проекты всегда имеют спрос, так как несут в себе особый характер. ВЗС2. При достижении высокого уровня абстракции модулей серверной части можно производить различные расширения, без вмешательства в исходный код программы. ВЗС5. Низкие технические требования позволяют усилить методы защиты и добавить обработку исключений.</p>	<p>используется игра с простыми игроками. В1Сл4. Продукт может не окупить себя встроенными покупками.</p>
<p>Угрозы: У1. Несвоевременное финансирование ТПУ. У2. Неизвестность проекта. У3. Форс-мажорные обстоятельства серверной площадки</p>	<p>У1С2. Отсутствие финансирования мешает масштабируемости проекта. УЗС1. В коде могут присутствовать незамеченные исключения, что может приводить к сбоям работы.</p>	<p>У1Сл1Сл2. При плохом финансировании штат проекта может сократиться, время исследования увеличивается. У1Сл5. Так отсутствие лицензионного соглашения ограничивает проект в</p>

	УЗС2. При отключения света на серверной площадке может сгореть оборудование.	его распространении. У2Сл2. Малое количество пользователей затягивает процесс обучения интеллектов. УЗСл1. Штат сотрудников может сократиться в любой момент.
--	--	---

Второй этап состоит в выявлении соответствия сильных и слабых сторон научно-исследовательского проекта внешним условиям окружающей среды. Это соответствие или несоответствие должны помочь выявить степень необходимости проведения стратегических изменений.

В рамках данного этапа необходимо построить интерактивную матрицу проекта. Ее использование помогает разобраться с различными комбинациями взаимосвязей областей матрицы SWOT.

Таблица 5.2 – Интерактивная матрица проекта

Сильные стороны проекта						
Возможности проекта		C1	C2	C3	C4	C5
	B1	0	+	-	-	-
	B2	-	+	0	+	-
	B3	+	+	-	+	+
Слабые стороны проекта						
Угрозы проекта		Сл1	Сл2	Сл3	Сл4	Сл5
	У1	+	+	-	-	0
	У2	-	+	-	+	-
	У3	+	+	-	-	-

5.3 Вывод

Методы, которые использовались в этом разделе, позволяют изучить коммерческую ценность разработки. Исследование слабых и сильных сторон позволяет правильно спроектировать проект для конкурентной среды с возможностью успешного закрепления на рынке. Изучение воздействий внешних факторов крайне необходимо чтобы избежать приостановления исследования проекта и усилить связь возможностей с сильными сторонами. Так же данный раздел показал, что проект конкурировать с существующими проектами. В перспективе планируется устранение слабых сторон и минимизация воздействия угроз на проект.

6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

6.1 Влияние на общество

В данной главе будут рассмотрены влияние конечного продукта на потребителя и методы защиты его от пагубных воздействий.

Конечным продуктом для пользователя будет являться приложение, визуально отображающее и симулирующее игру в покер. Соответственно, данное приложение предназначается для значительной аудитории. Данное приложение является как представителем типичной компьютерной игры, так и азартной игры. Так как приложение находится в свободном доступе, его потребителями могут являться как люди разных возрастных категорий (от детей до стариков), так и люди с различным психическим и эмоциональным состоянием. Поэтому следует рассмотреть два аспекта проблемы: игроманию, связанную непосредственно с азартными играми и компьютерную зависимость, которая представляет собой увлечение видеоиграми и компьютерными играми.

Помимо этого в продукте используются элементы искусственного интеллекта. Следовательно, необходимо рассмотреть возможные пагубные последствия и методы их решения, использованные в приложении, оградив пользователя от излишних психических воздействий и эмоциональных напряжений.

6.2 Игровая зависимость

В современном мире все больше и больше людей увлекаются компьютерными играми. Из-за развития игровой компьютерной индустрии растет проблема игровой зависимости. Это связано с изобретением новых и усовершенствованием старых технологий. Графика более реалистична и красочна. Окунаясь в виртуальную жизнь, уже не хочется возвращаться в реальность. Данное приложение является как представителем типичной компьютерной игры, так и азартной игры. Поэтому следует рассмотреть два аспекта проблемы: игроманию, связанную непосредственно с азартными

играми и компьютерную зависимость, которая представляет собой увлечение видеоиграми и компьютерными играми.

Игромания (игровая зависимость, лудомания) – вид психологической зависимости, которая представляет собой потребность в азартных играх, при этом утрачивается интерес к материальным, семейным, социальным ценностям. В последнее время проблема азартных игр приобрела исключительно важное значение в связи с повсеместным распространением игорного бизнеса. Помимо этого существуют другие различные азартные игры, но основная часть является компьютерной. На сегодняшнее время компьютерные игры распространены повсеместно. И проблема игровой и компьютерной зависимости не является новой.

Вследствие этого увлечение к азартным играм пришло не в одну семью, став серьезной угрозой разрушения семейного благополучия, и человеческой личности. Выходя из-под контроля, проблема стала не только частной, но и глобальной, распространяясь во все уголки мира. Входя в азарт, человек теряет контроль над временем и самим собой, что можно рассмотреть на примере казино (где специально нет окон, чтобы не дать игроку понять, сколько времени он тратит на свое увлечение, тем самым сбивая его биологические часы) или на примере компьютерных игр (где есть раскрытый сюжет, реалистичная графика, позволяющая полностью окунуться в виртуальный мир).

Люди, которые не могут реализовать себя в жизни, но имеют успех в компьютерных играх, стараются как можно сильнее забыть реальность и погрузиться в тот мир, где им комфортнее. Они перестают воспринимать игру как развлечение, придавая ей всю большую значимость, забывая свое реальное окружение, становясь более обособленными и изолированными от общества и реальной жизни. Количество таких людей по всему миру составляет более 3%, что является значительной угрозой для человечества в целом, которую необходимо решать немедленно при помощи различных

специалистов, такие как врачи узких специальностей (психотерапевты, психологи, наркологи, социологи и так далее).

Азартные игры поглощают людей всех возрастных, половых и социальных категорий. Однако в различных государствах есть собственные характерные черты. Таким образом, в Америке, к примеру, в азартных играх (в основном, данная игра и лотереи) принимают участие вплоть до 80% людей пенсионного возраста и только лишь 4% молодых людей. В государствах постсоветского пространства азартными играми занимаются, в основном, дети и молодое поколение, а люди пенсионного возраста, как норма, такого рода подчиненностью никак не страдают.

Данное приложение является мультиплатформенным, это означает, что оно может запускаться на различных операционных системах. Так же приложение способно запускаться не только на персональных компьютерах, но и на смартфонах, планшетах и т.п. То есть зона распространения данного продукта не ограничена. Помимо этого продукт находится в свободном доступе (является бесплатным), что значительно расширяет аудиторию. К ней будут относиться и такие люди, которые либо не желают платить за игру, либо не имеют на это возможности (например, подростки и малообеспеченное население). Почти у каждого сейчас имеется компьютер или смартфон, который не только делает жизнь проще и мобильнее, но и подвергает своего обладателя не малой опасности. Вследствие этого под фактор риска попадают люди от подросткового возраста до 45-летних людей. Только те, кто не играет в компьютерные игры, не попадают в зону риска данного приложения.

Таким образом, не одна открытость и простая достигаемость приложения является источником потенциальной опасности. Данный продукт основан по сюжету знаменитого азартного приложения Texas hold'em. Суть приложения заключается в игре в покер. Карточные игры еще с давних времен являлись причиной сильной зависимости. Азартные игры – игры, не имеющие четкой стратегии, основанные на случайности. У человека

появляется вера в удачу. «А вдруг повезет?!» Возникает сильное возбуждение, внутреннее оживление, необоснованно повышенное настроение и неутолимое желанием играть.

В данном приложении клиент имеет возможность общаться и с другими людьми. Можно провести аналогию с социальной сетью. Здесь так же можно общаться с друзьями, найти новых знакомых по интересам не только в самой игре, но и жизни и так далее. С одной стороны у человека есть собеседники, он общается и ничего не предвещает беды, но может наступить тот момент, когда просто сотрется грань между виртуальным и реальным миром. Виртуальные друзья становятся дороже настоящих, человек начинает жить, не отходя от монитора. Так одинокие в реальности люди «с головой» уходят в социальные сети, становясь по-настоящему зависимыми.

Так же данная игра предусматривает то, что люди будут играть не только сами с собой, в процесс включен искусственный интеллект. Поскольку искусственный интеллект является самообучаемым, то есть по мере своего существования, он накапливает полученные знания и улучшает навыки игры, по завершении своего обучения он сможет стать достаточно сильным противником, что может только подогреть интерес у игроков. Желая снова и снова выиграть у машины, человек не замечает, что просто не может оторваться от процесса. Поэтому наличие искусственного интеллекта так же является усугубляющим фактором, приводящим к зависимости.

Исходя из вышеперечисленного, в разработанном приложении очевиден риск формирования игровой зависимости у большого числа людей из той аудитории, которая намерена использовать данный продукт. Чтобы как можно больше обезопасить клиента от нежелательного воздействия были реализованы определенные методы.

Каждый клиент имеет собственную статистику, показывающую количество часов, которые человек проводит за игрой и частоту его пребывания за нею. Это позволит каждому игроку контролировать себя.

Ведь, увлекшись игрой, человек теряет счет времени. Данный метод позволит клиенту самостоятельно отследить часы своего пребывания за игрой и при необходимости (если человек превысил допустимую норму времени) оповестит его об этом. Это уведомление придет на почту конкретному клиенту. Если человек находится в игре больше положенного времени в сутки, то система исключит его из игры. Но это может вызвать бурю недовольства со стороны игроков. И чтобы приложение успешно реализовалось, и было по нраву как можно большему числу людей, система исключает игрока на небольшой промежуток времени (30 минут).

Так же с согласия пользователя собираются личные данные о нем, которые вводит непосредственно сам игрок или берутся с профиля в определенной социальной сети. Таким образом, будут известны как минимум пол и возраст пользователя. Это позволит установить разное количество допустимых часов в сутки для определенных возрастных групп, ограничит несовершеннолетних от игры на реальные деньги. Если человек не обращает внимания на предупреждения о том, что его часовой лимит периодически исчерпывается, система оповестит об этом людей, непосредственно связанных с этим игроком (родные и близкие). Дети будут контролируемы родителями, а люди, которые уже склонны к игровой зависимости, смогут своевременно получить помощь от близких.

Система контролирует не только время пребывания в игре, но и его потраченные деньги. Ведется статистика числа выигрышей и проигрышей. Если число проигрышей раз за разом увеличивается, и число проигранных денег непрерывно растет, опять же будут приходиться уведомления пользователю или его близким. В крайнем случае, игрока придется заблокировать.

Все вышеперечисленные методы будут иметь место, если сам пользователь или его окружение, которое так же приходят уведомления, принимают во внимание все факторы, которые ведут непосредственно к игровой зависимости. В противном случае, если все меры безопасности

будут нарушены и не соблюдены, клиент будет заблокирован данным приложением.

6.3 Искусственный интеллект

За последние десятилетия прогресс развитие техники сделало такой скачок вперед, который сложно было вообразить ранее. Все это заслуги Научно-технического прогресса - взаимообусловленного, поступательного развития науки и техники, производства и сферы потребления. Коренные изменения произошли по всему миру. Происходит поступательный процесс повышения эффективности техники и ее производительности, усложнения и обновления технических средств. Жизнь становится более простой и мобильной, включая в себя все больше и больше новых возможностей. Мы уже не можем представить свою жизнь без компьютера или мобильного телефона. Это неотъемлемые части нашего существования. Однако научно-технический прогресс представляет обществу и серьезную угрозу для человеческих жизней.

Злоупотребление преимуществами научно-технического прогресса, по мнению исследователей, может привести к созданию тоталитарного строя, во главе которого будет привилегированная господствующая элита, обладающая большими техногенными возможностями, и множество людей могут оказаться фактически рабами перед ее властью. Даже при надлежащем контроле весьма вероятен такой исход. Если же такой контроль будет невозможно реализовать, то человечество может прийти к разным рода катастрофам (ядерной, экологической, социальной и т.п.).

Исходя из этого, можно убедиться, что научно-технический прогресс несет в себе не только пользу для человечества, но и представляет огромную опасность для него. Поэтому на первый план в настоящее время выходит задача гуманистического использования достижений научно-технического прогресса в интересах общества, в интересах духовного обогащения всего человечества.

Следует рассмотреть возможное воздействие данного приложения на человека с точки зрения влияния техники и искусственного интеллекта на человечество в целом.

Характерной чертой нашего приложения является наличие искусственного интеллекта, который основан на нейронных сетях. Его обучение подобно тому, как родители обучают своих детей. То есть верные действия и решения будут одобрены, а погрешности будут выявлены и исправлены. Таким образом, постепенно получая и накапливая знания, интеллект саморазвивается, в дальнейшем же сможет работать наравне с реальным человеком, даже превосходя его. В данном приложении работа искусственного интеллекта заключается в утверждении и принятии решений при игре в покер.

Искусственный интеллект, внедренный в данное приложение, работает как некий черный ящик, то есть никто не знает, что там внутри как он будет действовать. Так как система является самообучаемой, ничто не запретит ей обучиться не только игре в покер, но и чему-то другому. Именно поэтому велика вероятность того, что действия этого механизма будут непредсказуемы. А непредсказуемость не является элементом безопасности, скорее наоборот. Искусственный интеллект заточен на получение как можно большей прибыли, то есть выигрыш. Поэтому возможно то, что чтобы определить или предугадать действия своих напарников, механизм научится анализировать поведения других игроков, их эмоциональное состояние: волнение или неуверенность, что еще больше будет подсказкой к ключу успеха. Непредугаданные действия могут быть такими, что ИИ научится влиять на игроков или подчинять их своей воле (например, будут заставлять сбрасывать все карты – сдаваться). Люди будут выходить из состояния равновесия, из провоцируют, выводят на конфликт. Заядлым игроманам, возможно, будут сложно смириться с вечными проигрышами и гнетом ИИ, что может привести к непредугаданным последствиям. Например, люди со

слабой психикой могут прийти в стадию высокого эмоционально напряжения, впасть в депрессию или же закончить жизнь самоубийством.

Подытожить все можно тем, что этот проект максимально защищен со стороны разработчиков и не допускает возможность некорректного поведения искусственного интеллекта. Все аспекты, которым был посвящён раздел социальной безопасности, рассматривает возможное негативное влияние приложения на аудиторию и методы, которые предотвращают это. Соответственно, немало важную роль играет человеческий фактор, который невозможно предугадать, т.е. поведение человека необъяснимо и не поддается каким либо законам. Проект будет находиться в постоянной поддержке, для того чтобы можно было предотвратить, то что не было описано. Так же хотелось бы отметить, что экземпляры игровых интеллектов будут проходить качественное исследование прежде, чем выпускать их в открытый доступ для игроков.

ЗАКЛЮЧЕНИЕ

В рамках проделанной работы была детально рассмотрена проблема создания искусственных игровых интеллектов. Рассмотрены различные варианты нейросетевых интеллектов, моделирующих поведение игрока в покер. Для тестирования и изучения интеллектов были реализованы различные среды, в том числе многопользовательское клиент-серверное приложение и менеджер игровых интеллектов.

Полученные результаты доказали эффективность применения генетических алгоритмов для обучения искусственных нейронных сетей, предназначенных для решения трудно формализуемых задач. Рассмотренные варианты интеллектов отличались различными наборами параметров искусственной нейронной сети и генетического алгоритма.

Несмотря на то, что рассмотреть все возможные варианты комбинаций параметров ИНС и генетического алгоритма не представляется возможным, с применением эвристического подхода были определены наиболее интересные наборы параметров, имеющие ярко выраженные черты. Например, рассматривались нейронные сети различным количеством межнейронных связей. При этом количество межнейронных связей варьировалось как количеством нейронов в слое, так и количеством скрытых слоёв. Аналогичным образом подбирались параметры генетического алгоритма: изучалось влияние новообразующего фактора на скорость обучения ИНС.

Полученные результаты показали, что в зависимости от структуры ИНС кардинальным образом меняется её способность к обучению и решению поставленной задачи. Слабыми решениями оказались искусственные нейронные сети с минимальным количеством скрытых слоёв: однослойные каскады показали полную неспособность к обучению. Лучшие результаты были получены у ИНС с большим количеством межнейронных связей, при этом внутренняя структура каскадов этих сетей имеет ступенчатый вид:

каждый последующий слой имеет большее количество нейронов, чем предыдущий.

Существенным недостатком метода обучения ИНС, применённого в данной работе, является скорость получения результатов: для того, чтобы получить приемлемые для оценки результаты необходимо провести очень большое количество испытаний. Сам процесс испытаний интеллектов также является неоптимальным и требует существенной доработки.

В качестве наиболее перспективных направлений развития данной работы можно выделить следующие:

1. Исследование вариантов интеллектов, основанных на искусственных нейронных сетях с большим количеством межнейронных связей и ступенчатой структурой каскадов.

2. Оптимизация процесса обучения. Необходимо сократить время, затрачиваемое на тестирование приспособленности особей в популяции. Для этого в первую очередь следует отказаться от численного метода Монте-Карло, который используется в текущем решении. Возможно, имеет смысл ввести третий каскад в общую структуру ИНС, который будет отвечать за вычисление вероятности выигрыша.

3. Использованы не все возможности графического «движка». Клиентское приложение может быть модифицировано, как с точки зрения пользовательского интерфейса, так и с точки зрения оптимизации использования компьютерных ресурсов.

СПИСОК ПУБЛИКАЦИЙ

1. Особенности разработки игрового интеллекта на основе искусственной нейронной сети [Электронный ресурс] / В. В. Иванцов, А. Т. Зиганшин // Молодежь и современные информационные технологии : сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 12-14 ноября 2014 г. в 2 т. / Национальный исследовательский Томский политехнический университет (ТПУ), Институт кибернетики (ИК) ; ред. кол. Е. А. Сикора [и др.]. — Томск: Изд-во ТПУ, 2014. — Т. 2. — [С. 136-137]. — Заглавие с титульного экрана. — Свободный доступ из сети Интернет. — Adobe Reader. Режим доступа: <http://www.lib.tpu.ru/fulltext/c/2014/C04/V2/061.pdf>

2. Особенности разработки многопользовательского клиент-серверного приложения на графическом движке Unity 2D [Электронный ресурс] / В. В. Иванцов, А. Т. Зиганшин // Молодежь и современные информационные технологии : сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 12-14 ноября 2014 г. в 2 т. / Национальный исследовательский Томский политехнический университет (ТПУ), Институт кибернетики (ИК) ; ред. кол. Е. А. Сикора [и др.]. — Томск: Изд-во ТПУ, 2014. — Т. 2. — [С. 134-135]. — Заглавие с титульного экрана. — Свободный доступ из сети Интернет. — Adobe Reader. Режим доступа: <http://www.lib.tpu.ru/fulltext/c/2014/C04/V2/060.pdf>

3. Разработка и использование графических ресурсов для реализации игрового приложения с использованием Unity 2D [Электронный ресурс] / В. В. Иванцов, А. Т. Зиганшин // Молодежь и современные информационные технологии : сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 12-14 ноября 2014 г. в 2 т. / Национальный исследовательский Томский политехнический университет (ТПУ), Институт кибернетики (ИК) ;

ред. кол. Е. А. Сикора [и др.]. — Томск: Изд-во ТПУ, 2014. — Т. 2. — [С. 235-236]. — Заглавие с титульного экрана. — Свободный доступ из сети Интернет. — Adobe Reader. Режим доступа: <http://www.lib.tpu.ru/fulltext/c/2014/C04/V2/111.pdf>

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Зубков Н. В., Сидоров С. Г. Нейронные сети. Их обучение и использование // «Математическое моделирование и информационные технологии»: Материалы региональной научно-технической конференции студентов и аспирантов. – 2011. – С. 54.
2. Формальный нейрон, 2002, Электронный ресурс: <http://nnet.chat.ru/neuron.html>.
3. Генетический алгоритм. Просто о сложном, 2011, Электронный ресурс: <https://habrahabr.ru/post/128704/>.
4. T. Cormen, Ch. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms (2009) // 3rd edition. – Cambridge, Massachusetts: MIT Press. – 1313 p.
5. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы // Под ред. В. М. Курейчика. – 2-е изд., исправл. и доп. – М.:ФИЗМАТЛИТ, 2010. – 368 с.
6. Сетевое программирование для разработчиков игр. Часть 1: UDP vs. TCP | SavePearlHarbor, 2014, Электронный ресурс: <http://savepearlharbor.com/?p=209144>.
7. Реферат - Протокол TCP/IP, 2011, Электронный ресурс: www.ronl.ru/referaty/informatika-programmirovanie/104449.
8. Обмен данных в сети, 2010, Электронный ресурс: http://djamaev-mtt.ucoz.ru/publ/lekcii/kompjuternye_seti/obmen_dannykh_v_seti/3-1-0-3.
9. Изучение Unity, 2016, Электронный ресурс: <https://unity3d.com/ru/learn>.
10. Unreal Engine 3 Features, 2016 Электронный ресурс: <https://www.unrealengine.com/previous-versions>.
11. CryENGINE® 3 | Crytek, 2016, Электронный ресурс: <http://www.crytek.com/cryengine/cryengine3/overview>.

12. "томский политехнический университет" В. Г. Спицын, Ю. Р. Цой представление знаний в информационных системах, 2007, Электронный ресурс: <http://rushkolnik.ru/docs/114/index-7094759.html?page=7>.
13. Оценивание популяции, 2015, Электронный ресурс <http://lektsiopedia.org/lek3-15849.html>.
14. Известия ЮФУ. Технические науки, 2005, Электронный ресурс: <http://izv-tn.tti.sfedu.ru/>.
15. Представление знаний в информационных системах (стр. 12), 2008, Электронный ресурс: <http://pandia.ru/text/80/016/47780-12.php>.
16. Иванцов В. В. Разработка и программная реализация нейросетевого алгоритма для принятия решений в интеллектуальной игре «Покер». – 2016
17. Спицын В. Г., Цой Ю. Р. Применение искусственных нейронных сетей для обработки информации. – 2007. – С. 3.
18. Покерные боты: просто о сложном. История и социализация, 2016, Электронный ресурс: <http://cop.glavpoker.ru/blogs/document22886.phtml>.