

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Основная профессиональная образовательная программа
Направление подготовки 09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»
форма обучения – очная

Курсовая работа

«Управление проектом по созданию искусственного интеллекта с
использованием машинного обучения»

Обучающегося 3 курса
Войтина Евгения Вячеславовича

Научный руководитель:
Кандидат физико-математических наук, ассистент
Жуков Николай Николаевич

Санкт-Петербург
2019

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ОСНОВНАЯ ЧАСТЬ	5
1. Теоретические аспекты разработки программного проекта с использованием машинного обучения	5
2. Описание разработки программного проекта	8
2.1 Организация работы.....	8
2.2 Разработка скрипта для перемещения	11
2.3 Разработка скрипта для следования	14
3. Вывод.....	15
ЗАКЛЮЧЕНИЕ	16
ЛИТЕРАТУРА	17
ПРИЛОЖЕНИЕ А	19
ПРИЛОЖЕНИЕ Б.....	23
1. ОБЩИЕ СВЕДЕНИЯ.....	25
1.1. Полное и краткое наименование проводимых работ	25
1.2. Наименование организации – заказчика	25
1.3. Список ответственных исполнителей.....	25
1.4. Плановые сроки начала и окончания работ по редизайну ресурса.....	25
1.5. Основные сведения об организации – заказчике.....	26
2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ РЕСУРСА.....	26
2.1 Назначение программного проекта	26
2.2 Цель создания ресурса	26
2.3 Целевая аудитория.....	26
2.4 Основные задачи ресурса	26
3. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОЕКТУ	26
3.1 Требование к стилистическому оформлению проекта	26
3.2 Требования к графическому дизайну программного проекта	26
3.3 Требования к средствам просмотра программного проекта	27
4. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОЕКТУ	27
4.1 Требование к функциональности агентов	27

ВВЕДЕНИЕ

В данной курсовой работе для написания проекта по созданию искусственного интеллекта с использованием машинного обучения были разработаны окружение для агента машинного обучения и код скрипта, регулирующий его поведения во время обучения.

Машинное обучение – это создание искусственного интеллекта с помощью алгоритмов, способных обучаться. Работа с машинным обучением актуальна как для учащихся и студентов, так и для профессионалов различных областей, от ракетостроения до систем распознавания. Использование машинного обучения позволяет облегчить функционирование многих систем, в том числе игровых.

Цели данной курсовой работы заключаются в описании процесса разработки проекта по созданию искусственного интеллекта с использованием машинного обучения, анализе существующих методологий разработки программных проектов, связанных с машинным обучением и определением оптимальной методологии.

Для достижения поставленной цели курсовой работы требуется решить следующие задачи:

- Разработка проекта:
 1. Разработка окружения, в котором ИИ мог бы функционировать.
 2. Разработка программного кода, регулирующего поведение агента для обучения перемещению.
 3. Разработка программного кода, регулирующего поведение агента для обучения следованию.

- Написание отчета с описанием процесса разработки.

Объектом данного исследования является разработка искусственного интеллекта.

Предметом исследования является использование машинного обучения в разработке искусственного интеллекта.

ОСНОВНАЯ ЧАСТЬ

1. Теоретические аспекты разработки программного проекта с использованием машинного обучения

Теоретический материал: чтобы лучше понять термины, использованные в данной работе, обратимся к теории.

Искусственный интеллект — в общем смысле это область информатики, которая занимается разработкой интеллектуальных компьютерных систем, то есть систем, обладающих возможностями, которые мы традиционно связываем с человеческим разумом, — понимание языка, обучение, способность рассуждать, решать задачи и т. д.

Машинное обучение — обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.

Обучение с подкреплением — один из способов машинного обучения, в ходе которого испытываемая система (агент) обучается, взаимодействуя с некоторой средой.

Существует множество методологий разработки программных проектов [1], далее будут описаны наиболее подходящие методологии для данного проекта и приведены их отличительные особенности.

RUP (Rational United Process):

- Концентрация на выполнении требований заказчиков к исполняемой программе (анализ и построение модели прецедентов).
- Ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки.

- Компонентная архитектура, реализуемая и тестируемая на ранних стадиях проекта.
- Работа над проектом в сплочённой команде, ключевая роль в которой принадлежит архитекторам.

Для данного проекта представленная методология не подходит по причине деления на роли.

Agile:

- Неформальные отношения важнее задокументированных. То есть устные договоренности между сотрудниками, между заказчиком и исполнителем важнее всего.
- Работающий продукт — главная оценка прогресса. Важны не инструменты, решения, производительность и изящество, а тот факт, что все запланированные возможности реализованы.

Данная методология не подходит т.к. при разработке проекта важнее был не работающий продукт, а именно описание его разработки и демонстрация работы с разными инструментами в процессе решения различных задач.

Spiral — это сложная организация жизненного цикла ПО, которая фокусируется на раннем выявлении и уменьшении проектных рисков. Разработка начинается в небольшом масштабе, решаются локальные задачи, оцениваются риски и пути их уменьшения. Следующий шаг охватывает более комплексные задачи — следующий виток спирали.

Данная методология лучше всего может подойти для разработки подобного проекта, ведь в ней нет четких указаний отношений между участниками и процесс разработки «от меньшего к большему» позволяет избежать большего количества проблем с кодом и его выполнением машиной.

Scrum — в основе концепции лежат «спринты». Спринт — короткая итерация, строго ограниченная по времени (обычно 2–4 недели). В это время минимизируется длительность совещаний, но увеличивается их частота.

Данная методология также подошла бы для разработки данного программного проекта. Спринты позволяют сконцентрироваться на конкретной задаче и её выполнении. Также данная методология позволяет разным участникам работать над разными задачами без постоянной привязки [4].

2. Описание разработки программного проекта

В данном разделе представлена разработка проекта и организация разработки.

Используемое оборудование: ПК, программное обеспечение «Unity», библиотека для использования машинного обучения «ml-agents», язык программирования C#.

Для организации разработки проекта были использованы следующие сервисы: GitHub.

Для создания программного продукта было решено использовать элементы из двух методологий Scrum и Spiral. Из Scrum были взяты спринты и частые встречи команды, из Spiral работа «по спирали».

2.1 Организация работы

Для достижения поставленных целей был выбран следующий план действий: необходимо разработать виртуальное окружение, в котором агент будет действовать, необходимо разработать код, приводящий агента в движение и обучающий нахождению пути до поставленной цели, необходимо разработать код, позволяющий агенту следовать за поставленной целью без непосредственно достижения её. Выполнение данных задач было решено делать частично параллельно, т.к. было очевидно, что придется вносить изменение в предыдущую работу в связи с проблемами в следующей [4].

Первостепенно было решено разработать окружение, ведь именно в ней придется действовать агенту в будущем, в тот момент, когда будет разработан минимальный набор окружения для работы агента можно переходить к следующим шагам.

Следующим шагом идет разработка кода обучения агента движению и достижению цели. Именно он будет стоять в основе следующего кода – кода обучения следованию.

В результате сложилась следующая диаграмма:

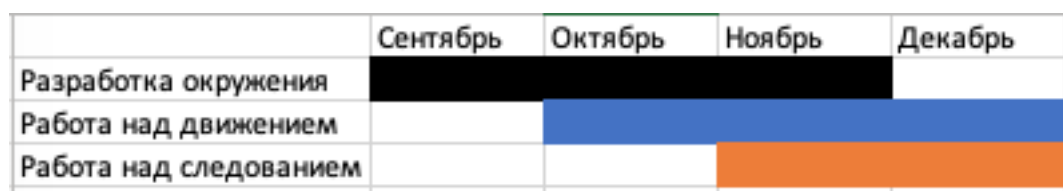


Рисунок 1

Для организации совместной работы была выбрана система GitHub. Был создан репозиторий, в который любой работающий над проектом мог загрузить свои изменения проекта либо обновить свой локальный проект в соответствии с последними изменениями других участников.

В GitHub использовались Issues (проблемы):

🐛	NullReferenceException when using rayperception	bug	#14 by mbuet2ner was closed 27 days ago	👍 Increment 2: Fi...
🚀	Add vision to MouseAgent	enhancement	#13 by mbuet2ner was closed 24 days ago	👍 Increment 2: Fi...
🐛	Fix physics of MouseAgent	bug	#12 by mbuet2ner was closed 24 days ago	👍 Increment 2: Fi...
🚀	Fix mouse orientation after import		#11 by mbuet2ner was closed on Nov 3	
🚀	Make mouse find the restaurant	enhancement	#10 by mbuet2ner was closed 24 days ago	👍 Increment 2: Fi...
🐛	Some objects disappear while in play mode	bug	#9 by mbuet2ner was closed 7 days ago	👍 Increment 2: Fi...
🚀	Add border to the environment	enhancement	#7 by mbuet2ner was closed on Oct 15	👍 Increment 2: Fi... good first issue help wanted
🐛	MouseAgent not moving	bug	#5 by mbuet2ner was closed on Oct 11	👍 Increment 1: Si... help wanted
🚀	Wire everything together: Trained agent for the current environment	enhancement	#4 by mbuet2ner was closed on Oct 11	👍 Increment 1: Si... help wanted
🚀	Create basic story	enhancement	#3 by mbuet2ner was closed on Oct 11	👍 Increment 1: Si... help wanted
🚀	Adapt AI environment	enhancement	#2 by mbuet2ner was closed on Oct 11	👍 Increment 1: Si... help wanted
🚀	Create virtual playground of "maze"	enhancement	#1 by mbuet2ner was closed on Sep 22	👍 Increment 1: Si... help wanted

Рисунок 2

Команда состояла из 3 человек, четкого разделения обязанностей не было, поэтому каждый член группы работал над любой объявленной в GitHub проблеме.

Встречи с заказчиком проходили раз в месяц, что позволяло держать его в курсе всех обновлений проекта, но в тоже время не встречаться слишком часто и избавиться от ненужных непродуктивных встреч.

2.2 Разработка скрипта для перемещения

В данном разделе представлена реализации логики поведения агента при обучении задачи перемещения.

Для разработки скрипта обучения для перемещения необходимо существование окружения для обучения. Оно представлено следующим образом:

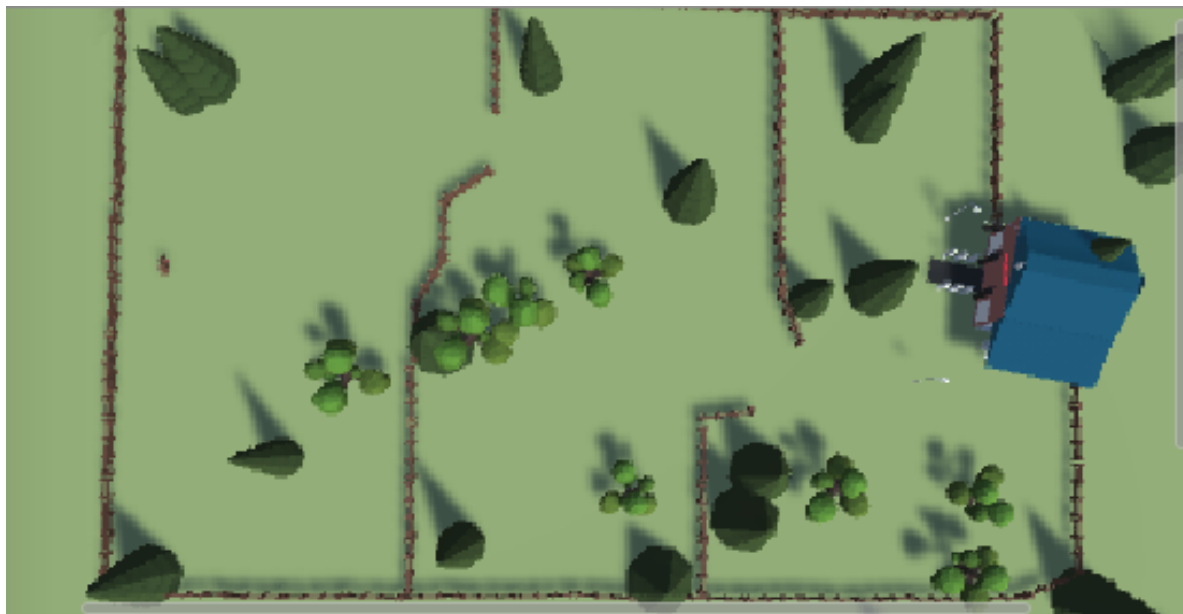


Рисунок 3

Справа находится здание, которое необходимо достичь агенту. Агент находится слева. Между ними расположены препятствия, которые необходимо избегать.

Необходим скрипт поведения, обеспечивающий агенту возможность взаимодействия с окружением, понимание того, что ему необходимо сделать, какие его действия являются правильными, а какие нет [7].

Скрипт представлен в ПРИЛОЖЕНИЕ А, ниже следуют поясняющие комментарии к нему.

Важным является данный элемент кода:

```
public override void CollectObservations()  
{
```

```

        // Target and Agent positions
        AddVectorObs(Target.localPosition);
        AddVectorObs(this.transform.localPosition);

        // Add raycast perception observations for fences and goals
        float rayDistance = 15f;
        float[] rayAngles = { 20f, 60f, 90f, 120f, 160f };
        string[] detectableObjects = { "fence", "goal" };
        AddVectorObs(rayPerception.Perceive(rayDistance, rayAngles, detectableObjects, 0f,
0f));

        // Add velocity observation
        Vector3 localVelocity =
transform.InverseTransformDirection(agentRigidbody.velocity);
        AddVectorObs(localVelocity.x);
        AddVectorObs(localVelocity.z);
    }

```

Здесь объявляются вещи, которые агент может «видеть»: агент знает свою позицию в пространстве и позицию своей цели, агент наблюдает, что находится рядом с ним с помощью «лучевого зрения», он выпускает перед собой лучи, которые, сталкиваясь с другими объектами, сообщают ему класс этого объекта.

Одним из самых важных, является система поощрений для агента. В данном случае, агент получает небольшое поощрение каждый раз, когда подходит немного ближе к цели. Каждый свой шаг агент получает небольшое пенальти, что сделано с целью достичь более быстрого выполнения поставленной задачи. Как только сумма накопленных пенальти достигла значения меньше -5, агент перезагружается, аккумулируя таким образом полученный опыт для дальнейшего обучения на своих ошибках или успехах [8].

```

    if (distanceToTarget > 1.42f)
    {
        AddReward(1.0f / distanceToTarget);
    }

    // Determine state
    if (GetCumulativeReward() <= -5f)

```

```

{
    // Reward is too negative, give up
    Done();
}
else
{
    // Encourage movement with a tiny time penalty and pdate the score text display
    AddReward(-.001f);
}

```

Также была продумана система поощрений за достижение цели и пенальти за столкновение с другими объектами.

```

private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.CompareTag("goal"))
    {
        // visualize collision with goal
        // remember to set gizmos in unity scene view
        Debug.DrawRay(collision.contacts[0].point, collision.contacts[0].normal,
Color.green, 1, true);
        AddReward(5f);
        Done();
    }
    else if (collision.gameObject.CompareTag("fence"))
    {
        // visualize collision with fence
        Debug.DrawRay(collision.contacts[0].point, collision.contacts[0].normal,
Color.red, 1, true);
        AddReward(-.01f);
    }
    else if (collision.gameObject.CompareTag("sign") && signFirstHit == true)
    {
        // visualize collision with sign
        Debug.DrawRay(collision.contacts[0].point, collision.contacts[0].normal,
Color.green, 1, true);
        AddReward(1f);
        signFirstHit = false;
    }
}

```

2.3 Разработка скрипта для следования

В данном разделе представлена реализация логики поведения агента при обучении задаче следования.

Очевидна разработка этого скрипта на основе скрипта для перемещения. Именно поэтому работа над этим скриптом началась позже, также уже существовало окружение для обучения.

Для достижения преследования цели агентом достаточно изменить систему поощрений для агента:

```
if (distanceToTarget > 1.42f)
{
    AddReward(1.0f / distanceToTarget);
    if (distanceToTarget > 20f)
    {
        AddReward(-10f);
        Done();
    }
}
if (distanceToTarget <= 1.4f)
{
    if (distanceToTarget >= 1.1f)
    {
        AddReward(1.0f);
    }
    else
    {
        AddReward(-0.5f);
        Done();
    }
}
```

Основной идеей изменений является заставить агента находится рядом с целью, но не сталкиваться с ней, именно для этого были введены штрафы для слишком близкого нахождения к цели и перезагрузка агента с огромным штрафом при слишком большом удалении от цели

3. Вывод

Выбранная система организации работы над проектом показала себя неэффективно. Перед командой стояла задача разработать полноценное окружение мышей, которое имеет свое расписание дня, которому следует.

Поставленная задача была выполнена не полностью. По окончании работы было разработано окружение, включающее в себя 2 мыши, перед первой стояла задача нахождения пути из лабиринта, перед второй стояла задача следования за первой.

Такой неудовлетворяющий результат связан с разрозненностью команды и отсутствием распределения задач по участникам. Не был выбран лидер команды, который ставил бы задачи перед другими. В результате каждый участник команды принимал на себя минимальную ответственность.

Для подобных проектов хорошо подойдет выбранная методология, однако обязательно обозначение лидера или же другого контролирующего элемента.

ЗАКЛЮЧЕНИЕ

При написании данной курсовой работы оставленные задачи решились не полностью. Посредством командной работы над проектом были достигнуты не все цели. Посредством документирования шагов работы была достигнута вторая цель – анализ проделанной работы и выводы из результатов.

В результате выполненной работы был разработан проект по созданию искусственного интеллекта с использованием машинного обучения, а именно: разработано окружение для работы ИИ, разработан скрипт для обучения перемещению, разработан скрипт для обучения следованию. Также был задокументирован процесс разработки для последующего отчета и анализа организации.

Получившейся проект возможен к доработке и поможет при работе над схожими проектами. Получившийся анализ организации процесса разработки поможет в будущем организовать процесс правильно с максимальной эффективностью.

ЛИТЕРАТУРА

1. Архипенков С. Лекции по управлению программными проектами. / www.arkhipenkov.ru. Москва. 2009
2. Балашов А.И. Управление проектами: учебник и практикум/ А.И. Балашов, Е.М. Рогова, М.В. Тихонова, Е.А. Ткаченко; под общей редакцией Е.М. Роговой. – М.: издательство Юрайт. 2015. – 383с.
3. В. А. Благодатских, В. А. Волнин, К. Ф. Посакалов. Стандартизация разработки программных средств. Учебное пособие. Москва. Финансы и статистика. 2005г, 288 стр
4. Гецци К., Джазайери М., Мандриоли Д. Основы инженерии программного обеспечения. 2-е изд.: Пер. с англ. – СПб: БХВ-Петербург, 2005. _ 832 с. 148 стр.
5. Котов. С.Л. Нормирование жизненного цикла программной продукции. –М.: ЮНИТИ- ДАНА. 2002. –143с.
6. Мостовой Я.А. Лекции по технологии разработки программного обеспечения. Учебное пособие. Издательство ПГУТИ. Самара.2014г.178 с.
7. Макконел С. Совершенный код. Мастер класс/ Пер. с англ. – М.: Издательско-торговый дом «Русская редакция»; СПб.: Питер, 2005. – 896 стр.
8. Орлик С. Инженерия ПО / <http://sorlik.blogspot.com>, <http://SWEBOK.Sorlik.ru>
9. Тьетар Р.-А. Менеджмент пер. с франц. – СПб: Издательский дом «Нева», 2003, - 96с.
10. Терехов А.Н. Технология программирования: учеб. Пособие / А.Н. Терехов. 2-е изд., -М.: Интернет Университет информационных технологий; БИНОМ. Лаборатория знаний, 2007.

11. Боэм Б.У. Инженерное проектирование программного обеспечения:
Пер. с англ.— М.: Радио и связь. 1985. — 512с.

ПРИЛОЖЕНИЕ А

```
using System.Collections.Generic;
using UnityEngine;
using MLAgents;
public class MouseAgent : Agent
{
    [Header("MouseAgent Settings")]
    public Transform Target;
    public float moveSpeed = 1f;
    public float rotateSpeed = 2f;
    private Rigidbody agentRigidbody;
    private RayPerception rayPerception;
    private Vector3 startPos;
    private bool signFirstHit = true;
    /// Initialize MouseAgent
    void Start()
    {
        agentRigidbody = GetComponent<Rigidbody>();
        rayPerception = GetComponent<RayPerception>();
        startPos = transform.localPosition;
    }
    /// <summary>
    /// Resets the agent to starting position
    /// Resets velocity
    /// </summary>
    public override void AgentReset()
    {
        transform.localPosition = startPos;
        this.agentRigidbody.angularVelocity = Vector3.zero;
        this.agentRigidbody.velocity = Vector3.zero;
    }
    /// <summary>
```

```

    /// Observations by the agent
    /// Target and own position, vision of "fence" and "goal"
    /// </summary>
    public override void CollectObservations()
    {
        // Target and Agent positions
        AddVectorObs(Target.localPosition);
        AddVectorObs(this.transform.localPosition);

        // Add raycast perception observations for fences and goals
        float rayDistance = 15f;

        float[] rayAngles = { 20f, 60f, 90f, 120f, 160f };

        string[] detectableObjects = { "fence", "goal" };

        AddVectorObs(rayPerception.Perceive(rayDistance, rayAngles,
        detectableObjects, 0f, 0f));

        // Add velocity observation
        Vector3 localVelocity =
        transform.InverseTransformDirection(agentRigidbody.velocity);

        AddVectorObs(localVelocity.x);
        AddVectorObs(localVelocity.z);
    }

    /// <summary>
    /// The actions that MouseAgent is able to do
    /// Moving and rotating
    /// </summary>
    /// <param name="vectorAction"></param>
    /// <param name="textAction"></param>
    public override void AgentAction(float[] vectorAction, string textAction)
    {
        // Determine the rotation action
        float rotateAmount = 0;

        if (vectorAction[1] == 1)
        {
            rotateAmount = -rotateSpeed;
        }

        else if (vectorAction[1] == 2)

```

```

    {
        rotateAmount = rotateSpeed;
    }

    // Apply the rotation (3D movement and rotation)
    Vector3 rotateVector = transform.up * rotateAmount;

    agentRigidbody.MoveRotation(Quaternion.Euler(agentRigidbody.rotation.eulerAngles
    + rotateVector * rotateSpeed));

    // Determine move action
    float moveAmount = 0;
    if (vectorAction[0] == 1)
    {
        moveAmount = moveSpeed;
    }
    else if (vectorAction[0] == 2)
    {
        moveAmount = moveSpeed * -.5f; // move at half-speed going backwards
    }

    // Apply the movement
    Vector3 moveVector = transform.forward * moveAmount;
    agentRigidbody.AddForce(moveVector * moveSpeed,
    ForceMode.VelocityChange);

    // Rewards
    float distanceToTarget = Vector3.Distance(this.transform.position,
    Target.position);

    //Debug.LogFormat("Distance to target: {0}", distanceToTarget);

    // Getting closer to target
    if (distanceToTarget > 1.42f)
    {
        AddReward(1.0f / distanceToTarget);
    }

    // Determine state
    if (GetCumulativeReward() <= -5f)
    {
        // Reward is too negative, give up
        Done();
    }

```

```

    }
    else
    {
        // Encourage movement with a tiny time penalty and pdate the score
text display
        AddReward(-.001f);
    }
}

/// <summary>
/// If collision with "goal" or "fence"
/// </summary>
/// <param name="collision"></param>
private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.CompareTag("goal"))
    {
        // visualize collision with goal
        // remember to set gizmos in unity scene view
        Debug.DrawRay(collision.contacts[0].point,
collision.contacts[0].normal, Color.green, 1, true);
        AddReward(5f);
        Done();
    }
    else if (collision.gameObject.CompareTag("fence"))
    {
        // visualize collision with fence
        Debug.DrawRay(collision.contacts[0].point,
collision.contacts[0].normal, Color.red, 1, true);
        AddReward(-.01f);
    }
    else if (collision.gameObject.CompareTag("sign") && signFirstHit ==
true)
        // visualize collision with sign
        Debug.DrawRay(collision.contacts[0].point,
collision.contacts[0].normal, Color.green, 1, true);
        AddReward(1f);
    }
}

```

```
signFirstHit = false;  
}  
}
```

ПРИЛОЖЕНИЕ Б

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»**

Техническое задание

**«Проект по созданию искусственного интеллекта с
использованием машинного обучения»**

Оглавление

ВВЕДЕНИЕ.....	3
ОСНОВНАЯ ЧАСТЬ.....	5
1. Теоретические аспекты разработки программного проекта с использованием машинного обучения	5
2. Описание разработки программного проекта	8
2.1 Организация работы.....	8
2.2 Разработка скрипта для перемещения.....	11
2.3 Разработка скрипта для следования	14
3. Вывод.....	15
ЗАКЛЮЧЕНИЕ	16
ЛИТЕРАТУРА	17
ПРИЛОЖЕНИЕ А	19
ПРИЛОЖЕНИЕ Б.....	23
1. ОБЩИЕ СВЕДЕНИЯ.....	25
1.1. Полное и краткое наименование проводимых работ.....	25
1.2. Наименование организации – заказчика	25
1.3. Список ответственных исполнителей	25
1.4. Плановые сроки начала и окончания работ по редизайну ресурса.....	25
1.5. Основные сведения об организации – заказчике	26
2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ РЕСУРСА	26
2.1 Назначение программного проекта	26
2.2 Цель создания ресурса	26
2.3 Целевая аудитория	26
2.4 Основные задачи ресурса	26
3. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОЕКТУ	26
3.1 Требование к стилистическому оформлению проекта	26
3.2 Требования к графическому дизайну программного проекта	26
3.3 Требования к средствам просмотра программного проекта	27
4. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОЕКТУ	27
4.1 Требование к функциональности агентов.....	27

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Полное и краткое наименование проводимых работ

Полное наименование проводимых работ – разработка программного проекта по созданию искусственного интеллекта с использованием машинного обучения.

Краткое наименование системы – программный проект по созданию ИИ.

1.2. Наименование организации – заказчика

Заказчик – Университет Оулу, факультет электроинженеринга.

1.3. Список ответственных исполнителей

Войтин Евгений Вячеславович – студент, разработчик проекта, сборщик системы и материала для наполнения, ответственный за реализацию проекта.

1.4. Плановые сроки начала и окончания работ по редизайну ресурса

Общий срок работ по разработке программного проекта по созданию ИИ составляет 120 календарных дней.

- Создание окружения для обучения ИИ на движке Unity – 90 календарных дней;
- Создание программного кода для обучения ИИ перемещаться и достигать цель – 90 календарных дней;
- Создание программного кода, для обучения ИИ перемещаться и следовать за целью – 60 календарных дней;
- Отладка выявленных ошибок – 120 календарных дня;

1.5. Основные сведения об организации – заказчике

Приоритетные направления деятельности:

- обучение;
- повышение квалификации;

2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ РЕСУРСА

2.1 Назначение программного проекта

Основным назначением программного проекта является предоставление платформы для исследования и экспериментов.

2.2 Цель создания ресурса

Целью создания ресурса является исследование поведения ИИ и эксперименты над ним.

2.3 Целевая аудитория

Студенты ИТ направлений.

Преподаватели ИТ направлений.

2.4 Основные задачи ресурса

1. Исследовательские;

3. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОЕКТУ

3.1 Требование к стилистическому оформлению проекта

Поскольку проект выполняется на базе движка Unity и библиотеке к нему, то в особом оформлении нет нужды, целевая аудитория способна ознакомиться с документацией данных средств и самостоятельно использовать проект.

3.2 Требования к графическому дизайну программного проекта

Конкретных требований к графическому дизайну также не существует. Дизайн должен быть интуитивно понятным, агенты должны быть легко

отличимыми от окружения, их действия должны быть очевидными, если они перемещаются, их перемещение должно отражаться и т.д.

3.3 Требования к средствам просмотра программного проекта

- Ресурс должен разрабатываться под базовое разрешение экрана 800х600 пикселей и выше;
- Ресурс должен обеспечивать корректное отображение данных для всех соотношений сторон экрана.

4. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОЕКТУ

Проект должен обеспечивать исследователям возможность тестирования возможностей ИИ.

Проект должен содержать несколько агентов-мышей, действующих независимо друг от друга. Должен быть разработан интерфейс для подключения «чистого» ИИ для обучения на основе поведения заранее обученных агентов.

4.1 Требование к функциональности агентов

Агенты-мыши должны перемещаться по окружению независимо от того, где они тренировались.

Агенты-мыши должны находить путь до цели.

Агенты-мыши должны уметь следовать за целью, без помех для нее.

