

Федеральное государственное автономное образовательное учреждение высшего
образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет: Институт издательского дела и журналистики

Кафедра «Информатика и информационные технологии»

Направление подготовки/ специальность: Информационные системы и технологии

ОТЧЕТ

по проектной практике

Студент: Федукина Мария Павловна Группа: 241-334

Место прохождения практики: Московский Политех, кафедра
«информатика и информационные технологии»

Отчет принят с оценкой _____ Дата _____

Руководитель практики: Рябчикова Анна Валерьевна

Москва 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

1. Общая информация о проекте:
 - Название проекта
 - Цели и задачи проекта
2. Общая характеристика деятельности организации (*заказчика проекта*)
 - Наименование заказчика
 - Организационная структура
 - Описание деятельности
3. Описание задания по проектной практике
4. Описание достигнутых результатов по проектной практике

ЗАКЛЮЧЕНИЕ (*выводы о проделанной работе и оценка ценности выполненных задач для заказчика*)

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЯ (*при необходимости*)

ВВЕДЕНИЕ

1. Общая информация о проекте

Название проекта

Название проекта «CharCreator» - производное от двух английских слов (character – персонаж, герой; creator – создатель). По задумке название должно быть простым в произношении и запоминании как русской, так и английской аудитории, а также отражать суть проекта.

Цели и задачи проекта

Главной целью проекта является создание функционирующего сайта со встроенным визуальным редактором внешности персонажа для DnD. Для выполнения данной цели необходимо не только четко выстроить цепочку задач, но и осознавать, что из себя представляет DnD. DnD – это ролевая настольная игра в жанре фэнтези, в которой есть игроки и мастер. Игроки под руководством мастера отправляются в выдуманное приключение, отыгрывая созданного ими персонажа. Для этого используются специальные листы персонажа, в которые записывается вся информация о нем, в том числе и его внешность. Таким образом, проект CharCreator это визуальный редактор внешности персонажа, изначально специализирующийся именно на игроках DnD. Для реализации данного проекта был поставлен ряд задач:

- Сбор и анализ информации:
 - Опрос ЦА;
 - Анализ трендов и конкурентов;
 - Создание карты пути пользователя;
- Создание и распределение технического задания (далее, ТЗ):
 - ТЗ для дизайнеров проекта;
 - ТЗ для программистов;

- Работа над дизайном:
 - Определение цветового решения;
 - Создание логотипа;
 - Создание макета сайта (как цифрового, так и бумажного);
 - Создание дизайна персонажей;
- Работа над технической частью:
 - Определение языка программирования, домена и т.п.;
 - Разработка сайта и визуального редактора;
- Работа над социальными сетями:
 - Создание группы ВКонтакте;
 - Привлечение аудитории;
 - Оформление постов, работа над их регулярным выходом и поиск нового контента.

Актуальность проекта

Актуальность проекта обусловлена ростом популярности настольной игры DnD. Рост популярности настольной игры приводит к притоку новых игроков, которые не привыкли к традиционной записи статистики своего игрового персонажа и поиску информации об игре в объемной Книге Игрока или Мастера, поэтому стремятся его оптимизировать: именно благодаря этому существуют сейчас такие сайты как «Long Story Short» и «dnd.su». Однако их функционал не подразумевает визуального редактора внешности персонажа. Существуют другие сайты, содержащие в себе подобный функционал, однако, по тем или иным причинам они недостаточно соответствуют требованиям игроков, что доказал проведенный анализ конкурентов и трендов. Именно поэтому создание визуального редактора внешности персонажа является актуальной проблемой.

2. Общая характеристика деятельности организации (заказчика проекта)

Наименование заказчика

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский Политехнический Университет».

Организационная структура

Организационная структура университета включает в себя несколько ключевых компонентов:

- административное управление (ректорат, отделы по учебной и научной деятельности);
- структурные подразделения – факультеты и институты, формирующие образовательный и исследовательский контент;
- центр дополнительного профессионального образования, обеспечивающий реализацию программ вне рамок основной учебной нагрузки.

Описание деятельности

Московский Политех осуществляет образовательную деятельность в рамках программ бакалавриата, магистратуры и дополнительного профессионального образования. Университет предлагает широкий спектр учебных направлений – от технических и инженерных до гуманитарных и цифровых, а также развивает курсы переподготовки и повышения квалификации. Одной из задач, стоящих перед университетом, является необходимость создания более эффективной, централизованной системы администрирования программ

дополнительного образования, что открывает перспективы для цифровизации и автоматизации учебных процессов.

3. Описание задания по проектной практике

В рамках проектной практики было поставлено задание разработать веб-продукт, тематически связанный с основным проектом, который выполнялся на «Проектной деятельности». Практика включала как базовую, так и вариативную части и предусматривала не только техническую реализацию, но и оформление сопроводительной документации.

Проект выполнялся индивидуально. Вся работа – от планирования структуры до наполнения контентом и публикации – велась мной самостоятельно, с использованием репозитория на GitHub (<https://github.com/uneverdreamed/project-practice-site>), что обеспечивало контроль версий и прозрачность хода разработки.

Базовая часть задания включала:

- инициализацию и настройку репозитория проекта;
- создание сайта с помощью статического генератора Hugo;
- реализацию нескольких разделов, включая главную страницу, страницу «О проекте», «Журнал» и «Ресурсы»;
- работу с шаблонами темы оформления (Ananke), адаптацию структуры сайта под цели проекта;
- размещение на сайте материалов проекта: презентации, графических ресурсов, игровых правил и ссылок;
- оформление проектной документации и итогового отчёта.

В процессе работы мной была выполнена настройка сайта на Hugo, подобрана и подключена тема оформления, организована структура контента (в том числе через каталоги content, static и layouts), а также подготовлены страницы с использованием синтаксиса Markdown. В разделе «Журнал» были опубликованы записи, отслеживающие ход разработки, в «Ресурсах» — выложены вспомогательные материалы, необходимые для ознакомления с проектом. Кроме того, были созданы главная страница с

общей информацией о проекте и страница с описанием участников. Все страницы сопровождаются фотографиями и визуально понятным оформлением, соответствующим общей стилистике проекта.

Проект создавался с нуля и стал логическим продолжением работы, выполненной в течение года в рамках основного проекта «CharCreator». В результате была подготовлена полноценная веб-страница, выполняющая функцию цифровой презентации и хранилища материалов проекта.

Вариативная часть включала разработку простой нейросетевой модели на языке Python, способной предсказывать следующий ход противника в игре «Камень-ножницы-бумага» на основе истории предыдущих раундов. Цель заключалась в демонстрации базовых принципов машинного обучения: подготовки данных, построения модели, обучения и оценки её качества.

Работа выполнялась индивидуально. Основные этапы включали:

- **Формирование обучающего датасета:** были собраны и размечены реальные и синтетические данные, представляющие собой последовательности ходов игрока и противника. Дополнительно реализована функция генерации правдоподобных последовательностей на основе вероятностных правил, имитирующих поведение игрока.
- **Предобработка данных и кодирование признаков:** каждый ход преобразовывался в вектор в формате one-hot. На основе истории длиной N (в финальной версии – 5 раундов) формировались входные признаки, учитывающие как действия самого игрока, так и его оппонента.

- **Разработка архитектуры нейросети:** была построена простая двухслойная модель с одним скрытым слоем. Входной слой принимал закодированные данные, скрытый слой активировался функцией sigmoid, выходной – softmax с тремя нейронами (для «Камня», «Ножниц» и «Бумаги»). Использовалась кросс-энтропийная функция потерь и L2-регуляризация.
- **Реализация обучения модели с нуля:** все операции реализованы вручную без использования готовых фреймворков, что позволило глубже понять принципы работы нейросетей.
- **Проведение обучения и анализ результатов:** модель обучалась на 80% данных, а точность проверялась на отложенной тестовой выборке (20%). Итоговая точность на тесте составила около 41%, что существенно превышает случайное угадывание (33%). Также были построены график снижения ошибки и матрица неточностей.
- **Формулировка выводов и ограничений:** дальнейшее повышение точности потребует перехода к более сложным архитектурам расширения обучающего набора. Работа продемонстрировала базовые возможности ИИ в предсказании шаблонного поведения в игре и может быть использована как основа для дальнейших экспериментов.

4. Описание достигнутых результатов по проектной практике

В рамках выполнения проектной практики мной был разработан сайт проекта с использованием генератора статических сайтов Hugo. Это позволило не только ускорить процесс создания сайта, но и получить опыт работы с современными веб-инструментами. Создана и оформлена структура репозитория на GitHub, в том числе добавлен ключевой файл README.md, поясняющий содержание проекта.

Разработан полноценный сайт, включающий пять основных разделов:

- **Главная страница (main)** – краткое описание проекта и его цели;

- **О проекте** (about) – подробный текст о замысле, актуальности, этапах и результатах;
- **Участники** (team) – информация об авторах проекта;
- **Журнал** (blog) – записи, отражающие ход выполнения работы;
- **Ресурсы** (resources) – вспомогательные материалы, включая правила настольной игры, презентацию, ссылки на фигму и соцсети.

Каждая страница оформлена с использованием Markdown и интегрирована в структуру сайта Hugo. Визуально сайт построен на базе темы Ananke, поддерживающей адаптивность и простоту восприятия. Хотя не удалось полностью реализовать задуманный внешний вид, использование Hugo соответствовало методическим рекомендациям, в которых прямо указывается, что генераторы сайтов (в особенности Hugo) являются предпочтительным вариантом при наличии базовых навыков HTML/CSS.

В рамках вариативной части практики, была разработана простая нейросетевая модель на языке Python, предсказывающая следующий ход противника в игре «Камень-ножницы-бумага» на основе истории предыдущих раундов:

- кодировка входных данных в формате one-hot (ходы игрока и соперника за 5 последних раундов);
- двухслойная архитектура: скрытый слой с 16 нейронами и сигмной функцией, выходной слой с функцией softmax;

Дополнительно были реализованы модули, обеспечивающие полноценную работу и оценку модели:

- генерация обучающих данных с ограничением на повтор ходов;
- обучение нейросети с нуля методом обратного распространения ошибки (backpropagation);
- визуализация процесса обучения (график ошибки), а также оценка точности предсказания и построение confusion matrix;
- демо-режим, позволяющий сыграть против обученной модели (отключено по умолчанию).

Приобретенные навыки:

1) Работа с Git и GitHub

- Создание и структурирование репозитория проекта
- Ведение истории изменений и работа с коммитами
- Размещение проекта в удалённом репозитории

2) Работа с генераторами сайтов (Hugo)

- Установка и настройка Hugo
- Использование темы оформления (Ananke), адаптация шаблонов
- Работа с Markdown и параметрами конфигурации (config.toml)
- Создание структуры сайта и страниц на основе шаблонов

3) Основы веб-разработки

- Написание и редактирование HTML и CSS
- Интеграция медиа-контента (изображений, логотипов, фоновых изображений)
- Попытка настройки визуальной части сайта (цветовая схема, фон и т.п.)

4) Python-разработка

- Работа с массивами и матрицами при кодировании входных данных (one-hot)

- Реализация собственной нейросети с нуля, включая прямой и обратный проход
- Создание инструментов для анализа данных и визуализации результатов обучения (matplotlib)

5) Документирование проекта

- Подготовка и оформление текстов для сайта
- Структурирование информации в формате Markdown
- Составление итогового отчета по проектной практике

ЗАКЛЮЧЕНИЕ

Проектная практика стала для меня важным этапом в профессиональном и личном развитии. Она позволила не только применить полученные ранее знания на практике, но и попробовать себя в роли разработчика полноценного цифрового продукта – от проектирования структуры до технической реализации и финального оформления.

Работа над сайтом проекта дала возможность освоить генератор статических сайтов Hugo, научиться организовывать структуру проекта, работать с шаблонами и конфигурационными файлами, а также размещать контент с помощью языка разметки Markdown. Несмотря на то что мне не удалось в полной мере реализовать изначально задуманную визуальную концепцию сайта (в частности, настройку фона и цветовой схемы), я получила полезный опыт адаптации готовых тем и настройки внешнего вида с помощью CSS и параметров Hugo.

Особую ценность имела реализация вариативной части задания – нейросетевой модели на Python. Работа над ней потребовала глубокого погружения в основы машинного обучения, включая подготовку и обработку данных, разработку архитектуры модели и реализацию обучения с нуля. Это позволило не только лучше понять алгоритмы работы нейросетей, но и развить навыки работы с числовыми данными, логикой программирования и визуализацией результатов.

Также в рамках проекта я освоила базовые принципы работы с Git и GitHub, что обеспечило прозрачность и управляемость процесса разработки. Подготовка отчета и оформление сайта помогли систематизировать знания, улучшить навыки технического письма и визуального представления информации.

Несмотря на отсутствие командной работы, выполнение всех этапов проекта в одиночку дало ценный опыт самоуправления, планирования и решения нестандартных задач. Уверена, что этот проект станет важной вехой в моей учебной и будущей профессиональной деятельности. Полученные знания и навыки обязательно найдут применение в дальнейших образовательных и практических задачах, особенно в сфере разработки пользовательских интерфейсов и интеллектуальных систем.

Подтверждаю, что отчет выполнен лично и соответствует требованиям
практики

Федукина Мария Павловна

ФИО

23.05.2025

дата



подпись

СПИСОК ЛИТЕРАТУРЫ

- 1) Купер Нейт, Джи Ким. *Как создать сайт: Пошаговое руководство для начинающих*. — М.: Манн, Иванов и Фербер, 2016.
- 2) *How Many People Play D&D in 2024? (User & Growth Stats)* // Fiction Horizon. — URL: <https://fictionhorizon.com/how-many-people-play-dd/>
- 3) *Hugo Quick Start Guide* — Официальная документация Hugo. — URL: <https://gohugo.io/getting-started/quick-start/>
- 4) *Hugo Documentation* — Полная документация по Hugo. — URL: <https://gohugo.io/documentation/>
- 5) *Ananke Theme for Hugo* — GitHub репозиторий темы Ananke. — URL: <https://github.com/theNewDynamic/gohugo-theme-ananke>
- 6) *Pro Git* (Scott Chacon and Ben Straub). — 2nd Edition. Apress, 2014. — URL: <https://git-scm.com/book/en/v2>
- 7) *Markdown Guide* — Полное руководство по синтаксису Markdown. — URL: <https://www.markdownguide.org/basic-syntax/>
- 8) *Matplotlib Documentation* — Официальная документация библиотеки Matplotlib. — URL: <https://matplotlib.org/stable/index.html>

ПРИЛОЖЕНИЯ

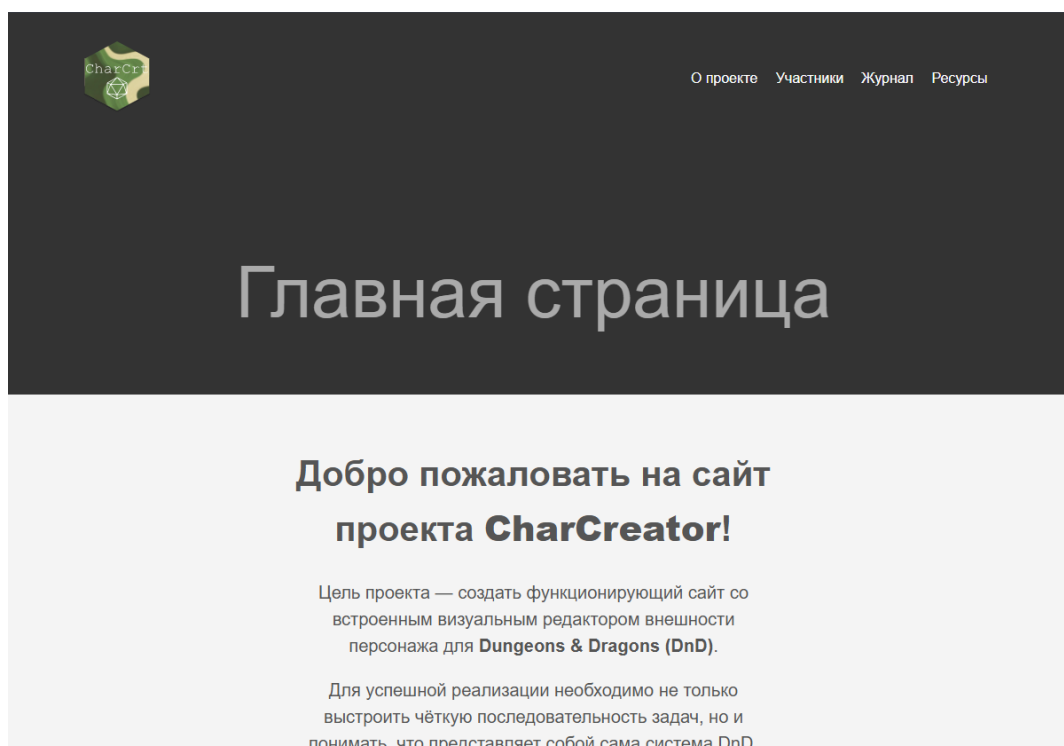


Рисунок 1 - страница main 1

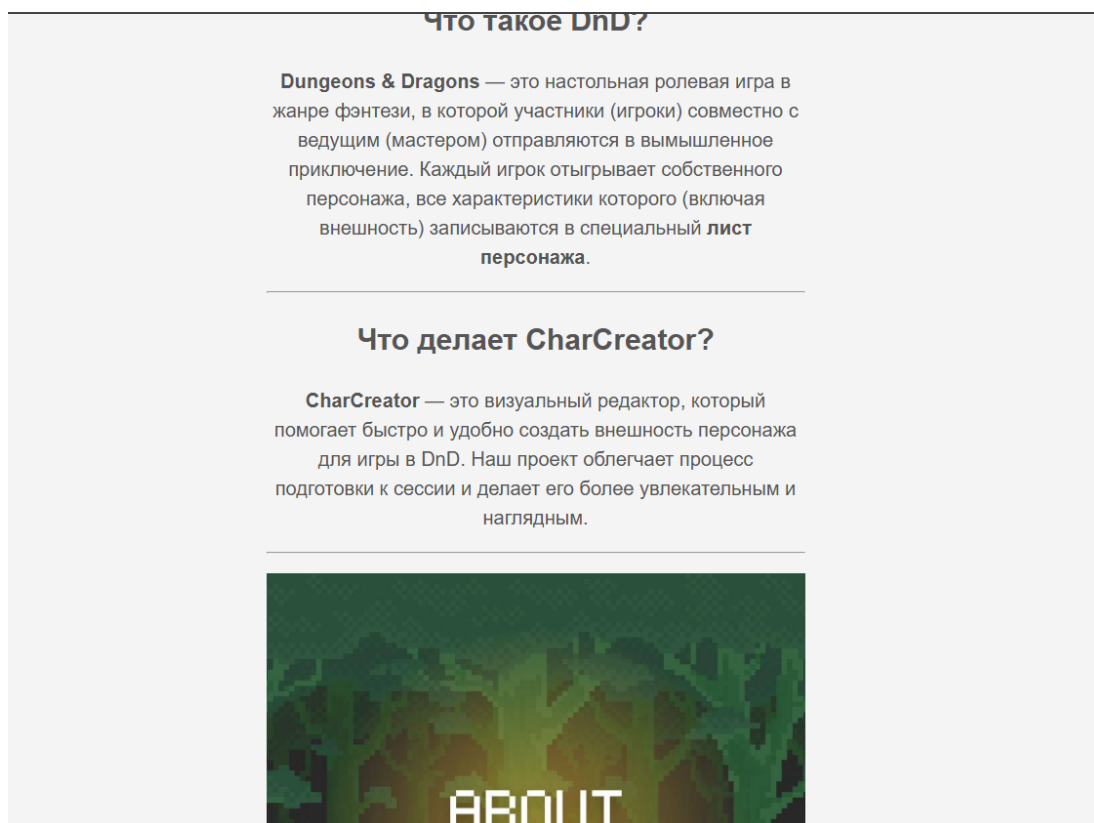


Рисунок 2 - страница main 2



О проекте

ГЛАВНАЯ СТРАНИЦА

О проекте

О проекте CharCreator

CharCreator — это визуальный редактор внешности персонажа для настольной ролевой игры Dungeons and Dragons (DnD). Цель проекта —

Рисунок 3 - страница about 1

О проекте CharCreator

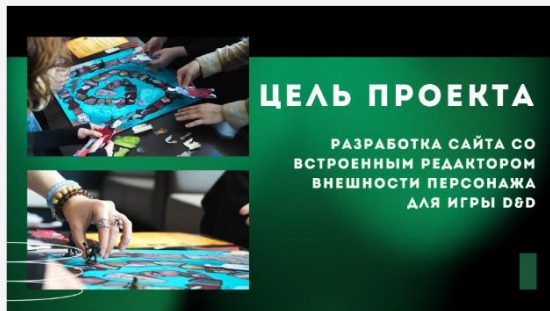
CharCreator — это визуальный редактор внешности персонажа для настольной ролевой игры Dungeons and Dragons (DnD). Цель проекта — упростить процесс создания образа персонажа для игроков, предоставив удобный онлайн-инструмент с элементами визуального редактирования.



Рисунок 4 - страница about 2

Цели проекта

- Создание сайта с визуальным редактором внешности персонажа для DnD.
- Анализ потребностей целевой аудитории (ЦА) и актуальности проекта.
- Разработка дизайна и технической части проекта.



Основные задачи

1. Исследование:

- Анализ ЦА;

Рисунок 5 - страница about 3

- Построение карты пути пользователя.

2. Проектирование:

- Постановка технического задания (ТЗ) для дизайнеров и разработчиков;
- Разработка логотипа, цветовой палитры, макета сайта и дизайна персонажей.

3. Реализация:

- Создание набора деталей внешности;
- Разработка сайта и визуального редактора;
- Поддержка и развитие социальных сетей (например, группы ВКонтакте).

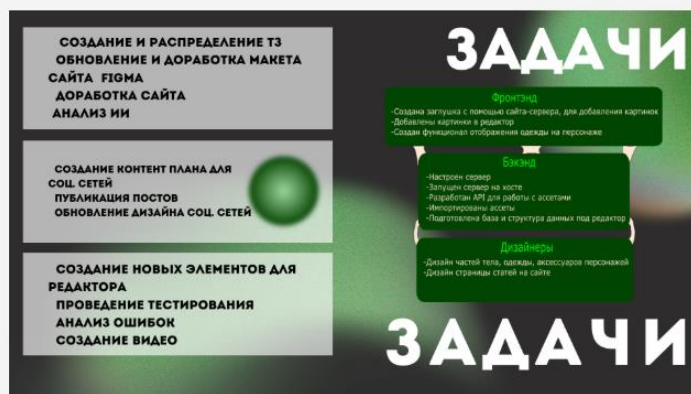


Рисунок 6 - страница about 4

С ростом популярности DnD растёт и потребность в удобных онлайн-инструментах. Существующие ресурсы либо не предоставляют визуального редактора, либо не соответствуют ожиданиям пользователей. Это делает CharCreator актуальным и востребованным решением.



Рисунок 7 - страница about 5

ГЛАВНАЯ СТРАНИЦА

Участники

Участники проекта CharCreator:

Разработчики

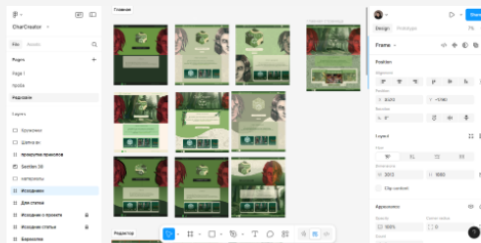
| Имя | Группа |
|-------------------------------------|----------|
| Андреев Кирилл Евгеньевич | 231-334 |
| Бухуров Амирхан Тимурович | 231-336 |
| Демченко Тимофей Юрьевич | 231-331 |
| Едисеев Олег Владимирович | 221-329 |
| Зотов Владимир Павлович | 231-352 |
| Кагарманова Екатерина Александровна | 221-372 |
| Косимов Косимджон Эргашбоевич | 221-331 |
| Луговских Данил Дмитриевич | 221-323 |
| Магомедов Гаджи Магомедрасулович | 231-322 |
| Максимчев Артем Андреевич | 231-3212 |
| Правдолюбова Анастасия Андреевна | 231-331 |
| Пяткин Максим Игоревич | 231-336 |

Рисунок 8 - страница team 1



Дизайнеры интерфейса

| Имя | Группа |
|----------------------------------|---------|
| Попова Екатерина Дмитриевна | 231-821 |
| Правдолюбова Анастасия Андреевна | 231-331 |
| Прокопова Варвара Константиновна | 241-332 |
| Федукина Мария Павловна | 241-334 |



Художники

| Имя | Группа |
|-------------------------------|---------|
| Брыскина Сабина Александровна | 231-023 |
| Денисова Софья Алексеевна | 231-023 |
| Манушина Ксения Дмитриевна | 231-821 |
| Попова Екатерина Дмитриевна | 231-821 |

Рисунок 9 - страница team 2

Художники

| Имя | Группа |
|-------------------------------|---------|
| Брыскина Сабина Александровна | 231-023 |
| Денисова Софья Алексеевна | 231-023 |
| Манушина Ксения Дмитриевна | 231-821 |
| Попова Екатерина Дмитриевна | 231-821 |
| Суздалева Дарья Евгеньевна | 231-023 |
| Хильченко Полина Денисовна | 231-023 |
| Федукина Мария Павловна | 241-334 |



Рисунок 10 - страница team 3

SMM

| Имя | Группа |
|------------------------------|---------|
| Анисимова Алиса Владимировна | 231-023 |
| Бушуева Ирина Дмитриевна | 231-331 |
| Мацегора Юлия Владимировна | 231-023 |
| Федоров Павел Вячеславович | 231-334 |



Рисунок 11 - страница team 4

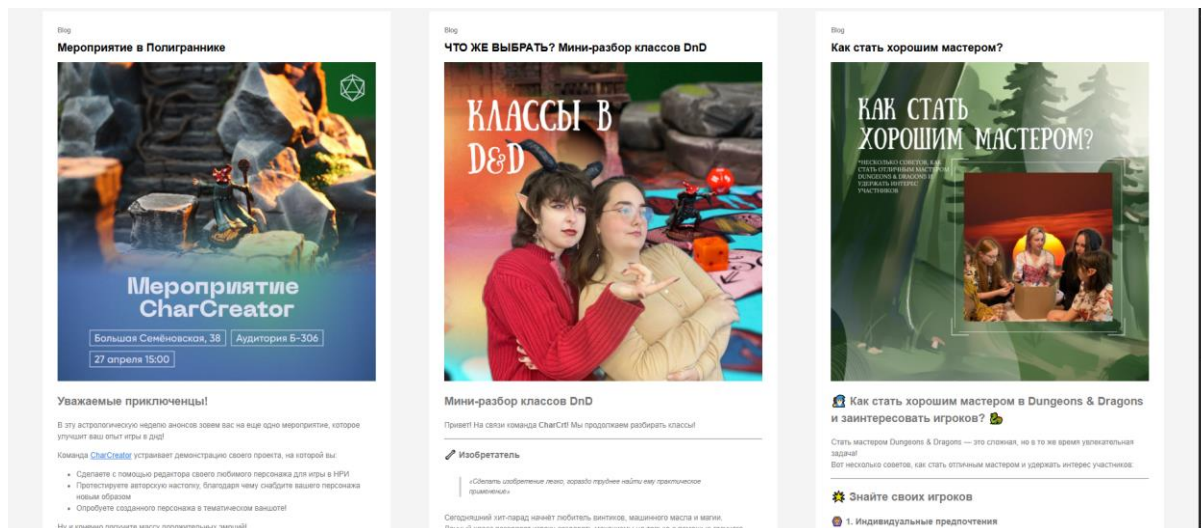


Рисунок 12 - страница blog 1

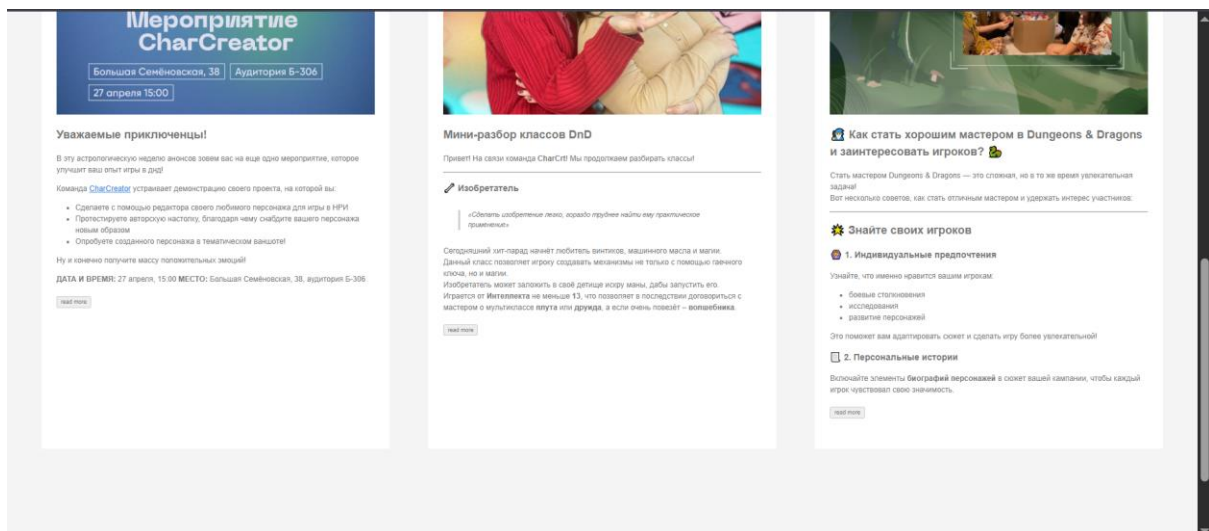


Рисунок 13 - страница blog 2



Рисунок 14 - страница resources 1



Презентация проекта

- [Скачать презентацию \(PowerPoint\)](#)

Рисунок 15 - страница resources 2

```

import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
import matplotlib.pyplot as plt
import random

##### ДАННЫЕ #####
#####
my_moves = ['Б', 'К', 'Б', 'Б', 'Б', 'Н', 'Н', 'К', 'Б', 'Б',
            'Н', 'Б', 'Б', 'Б', 'Н', 'К', 'Б', 'К', 'Б', 'Н',
            'Б', 'К', 'Н', 'Б', 'Н', 'К', 'Н', 'К', 'Б', 'Н',
            'Н', 'К', 'Н', 'Б', 'К', 'Н', 'Б', 'Б', 'К', 'Н',
            'Н', 'К', 'Б', 'Б', 'Б', 'К', 'Н', 'Б', 'Б', 'Н',
            'К', 'Б', 'К', 'Б', 'К', 'К', 'К', 'Б', 'К', 'Б',
            'К', 'Н', 'К', 'К', 'Н', 'Б', 'Б', 'Б', 'Б', 'Б',
            'Н', 'Н', 'Б', 'К', 'К', 'Б', 'Б', 'К', 'Н', 'Б',
            'Б', 'Н', 'Б', 'Н', 'К', 'Н', 'Н', 'К', 'Б', 'К',
            'Б', 'К', 'Б', 'Б', 'К', 'Н', 'Н', 'К', 'Б', 'Н',
            'К', 'К', 'Н', 'Б', 'Б', 'Б', 'К', 'К', 'Н', 'Б',
            'К', 'Б', 'Н', 'Б', 'Б', 'К', 'К', 'Н', 'К', 'Н',
            'Б', 'К', 'Б', 'Б', 'Н', 'К', 'К', 'Б', 'Б', 'К',
            'Н', 'К', 'Н', 'К', 'Б', 'Н', 'Б', 'К', 'К', 'Б',
            'К', 'Б', 'К', 'Н', 'Н', 'Б', 'Н', 'К', 'Н', 'К',
            'Б', 'Н', 'Б', 'К', 'Б', 'К', 'Б', 'Н', 'К', 'Б',
            'К', 'К', 'Н', 'Б', 'К', 'Н', 'Н', 'К', 'Б', 'Б',
            'Б', 'Н', 'Н', 'Б', 'К', 'К', 'Н', 'Н', 'Б', 'К',
            'К', 'Б', 'Н', 'К', 'Н', 'Н', 'Б', 'Н', 'К', 'Б',
            'К', 'К', 'Н', 'Н', 'Б', 'Н', 'К', 'К', 'Б', 'К',
            'К', 'Б', 'К', 'Н', 'Н', 'К', 'Н', 'К', 'Б', 'К',
            'Б', 'К', 'Б', 'К', 'Н', 'Б', 'К', 'Н', 'К', 'Н',
            'Б', 'К', 'Н', 'Н', 'К', 'Б', 'К', 'Н', 'Б', 'Н',
            'К', 'Б', 'Б', 'Н', 'К', 'К', 'Н', 'К', 'Б', 'Н',
            'К', 'Б', 'Н', 'Б', 'К', 'Б', 'К', 'Н', 'Н', 'Б']

opponent_moves = ['Н', 'Н', 'К', 'К', 'Н', 'Н', 'Н', 'Н', 'Н', 'К',
                  'К', 'К', 'К', 'Н', 'Б', 'Н', 'Н', 'Н', 'Б', 'Б',
                  'Н', 'К', 'К', 'Б', 'Н', 'Б', 'Б', 'Н', 'Б', 'К',
                  'Б', 'К', 'К', 'Б', 'Н', 'Б', 'К', 'Б', 'Б', 'Н',
                  'К', 'Б', 'Н', 'Б', 'К', 'Б', 'Б', 'Н', 'Б', 'Б',
                  'Н', 'Н', 'Б', 'Н', 'Н', 'К', 'К', 'Б', 'Н', 'К',
                  'Н', 'Н', 'Б', 'Н', 'К', 'К', 'Н', 'Н', 'К', 'Б',
                  'К', 'Н', 'К', 'Н', 'Б', 'Н', 'К', 'К', 'К', 'Н',
                  'Б', 'Б', 'К', 'К', 'Н', 'Н', 'К', 'К', 'Н', 'К',
                  'Н', 'Б', 'Н', 'Б', 'К', 'К', 'Б', 'Б', 'Н', 'Н',
                  'Н', 'К', 'К', 'Б', 'Н', 'Н', 'Б', 'К', 'К', 'Б',
                  'Б', 'Н', 'Б', 'К', 'Н', 'К', 'Н', 'Н', 'Б', 'Б',
                  'К', 'К', 'Н', 'Н', 'Б', 'Н', 'К', 'К', 'Б', 'К',
                  'Н', 'Б', 'К', 'Н', 'К', 'К', 'Н', 'Н', 'Б', 'Б',
                  'К', 'Н', 'Н', 'Б', 'Н', 'К', 'Н', 'К', 'Б', 'К',
                  'К', 'Б', 'К', 'Н', 'Н', 'К', 'Н', 'Б', 'К', 'Н',
                  'Б', 'К', 'Б', 'К', 'Н', 'Б', 'К', 'Н', 'К', 'Н',
                  'Б', 'К', 'Н', 'Н', 'К', 'Б', 'К', 'Н', 'Б', 'Н',
                  'К', 'Б', 'Б', 'Н', 'К', 'К', 'Н', 'К', 'Б', 'Н',
                  'К', 'Б', 'Н', 'Б', 'К', 'Б', 'К', 'Н', 'Н', 'Б']

def generate_smart_sequence(length=100):
    moves = ['R', 'P', 'S']
    sequence = []

    for i in range(length):
        if len(sequence) >= 3 and sequence[-1] == sequence[-2] ==
sequence[-3]:
            choices = [m for m in moves if m != sequence[-1]]
            next_move = random.choice(choices)
        elif len(sequence) >= 2 and sequence[-1] == sequence[-2]:

```



```

        if random.random() < 0.7:
            choices = [m for m in moves if m != sequence[-1]]
            next_move = random.choice(choices)
        else:
            next_move = sequence[-1]
    else:
        # В остальных случаях — 60% сменить
        if len(sequence) > 0 and random.random() < 0.6:
            choices = [m for m in moves if m != sequence[-1]]
            next_move = random.choice(choices)
        else:
            next_move = random.choice(moves)

    sequence.append(next_move)

    return sequence

smart_my_moves = generate_smart_sequence(500)
smart_opponent_moves = generate_smart_sequence(500)

rps_to_russian = {'R': 'К', 'P': 'Б', 'S': 'Н'}
smart_my_moves = [rps_to_russian[m] for m in smart_my_moves]
smart_opponent_moves = [rps_to_russian[m] for m in smart_opponent_moves]

my_moves += smart_my_moves
opponent_moves += smart_opponent_moves

move_to_int = {'К': 0, 'Н': 1, 'Б': 2}
int_to_move = {0: 'Камень', 1: 'Ножницы', 2: 'Бумага'}

X = [] # история ходов игрока + противника
y = [] # следующий ход противника

history_size = 5 # сколько прошлых ходов учитывать

for i in range(history_size, len(my_moves)):
    my_history = my_moves[i - history_size:i]
    opp_history = opponent_moves[i - history_size:i]
    encoded = []

    for m, o in zip(my_history, opp_history):
        # one-hot игрока
        onehot_my = [0, 0, 0]
        onehot_my[move_to_int[m]] = 1
        encoded.extend(onehot_my)

        # one-hot противника
        onehot_opp = [0, 0, 0]
        onehot_opp[move_to_int[o]] = 1
        encoded.extend(onehot_opp)

    X.append(encoded)
    y.append(move_to_int[opponent_moves[i]])

X = np.array(X)
y = np.array(y)
y_onehot = np.eye(3)[y]

##### архитектура сети
#####

```

```

def sigmoid(x, deriv=False):
    x = np.clip(x, -500, 500)
    if deriv:
        return x * (1 - x)
    return 1 / (1 + np.exp(-x))

def softmax(x):
    exp_x = np.exp(x - np.max(x, axis=1, keepdims=True))
    return exp_x / np.sum(exp_x, axis=1, keepdims=True)

def predict(X):
    l1 = sigmoid(np.dot(X, syn0))
    l2 = softmax(np.dot(l1, syn1))
    return np.argmax(l2, axis=1)

input_size = history_size * 3 * 2 # 3 значения (К,Н,Б) * 2 игрока *
history_size
hidden_size = 16 # нейронов в скрытом слое
output_size = 3 # 3 возможных хода (К, Н, Б)

np.random.seed(42)
syn0 = 2 * np.random.random((input_size, hidden_size)) - 1 # веса вход ->
скрытый слой
syn1 = 2 * np.random.random((hidden_size, output_size)) - 1 # веса скрытый
-> выходной слой

learning_rate = 0.01 # скорость обучения
epochs = 5000 # количество итераций обучения
l2_lambda = 0.01

##### обучение сети
#####
X_train, X_test, y_train, y_test = train_test_split(X, y_onehot,
test_size=0.2, random_state=42)

loss_history = []

for epoch in range(epochs):
    l0 = X_train
    l1 = sigmoid(np.dot(l0, syn0))
    l2 = softmax(np.dot(l1, syn1))

    l2_error = y_train - l2
    l2_delta = l2_error # для softmax + cross-entropy

    l1_error = l2_delta.dot(syn1.T)
    l1_delta = l1_error * sigmoid(l1, deriv=True)

    syn1 += (l1.T.dot(l2_delta) - l2_lambda * syn1) * learning_rate
    syn0 += (l0.T.dot(l1_delta) - l2_lambda * syn0) * learning_rate

    loss = np.mean(np.abs(l2_error))
    loss_history.append(loss)

    if epoch % 1000 == 0:
        print(f"Epoch {epoch}, Loss: {loss:.4f}")

##### оценка модели #####
def evaluate_model(X, y_true_onehot):
    y_pred = predict(X)

```

```

y_true = np.argmax(y_true_onehot, axis=1)
accuracy = accuracy_score(y_true, y_pred)
cm = confusion_matrix(y_true, y_pred)
return accuracy, cm

train_accuracy, train_cm = evaluate_model(X_train, y_train)
test_accuracy, test_cm = evaluate_model(X_test, y_test)

print(f"\nTrain Accuracy: {train_accuracy:.2%}")
print(f"Test Accuracy: {test_accuracy:.2%}")

plt.plot(loss_history)
plt.title("График ошибки при обучении")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.show()

##### ИНТЕРАКТИВНОЕ ДЕМО
#####

# def play_vs_ai():
#     print("\nДемо-режим: игра против ИИ")
#     print("Вводите свои ходы (К - камень, Н - ножницы, Б - бумага, Q -
выход) ")
#
#     # история последних ходов (игрока и ИИ)
#     game_history = []
#
#     while True:
#         # Ввод хода игрока
#         player_move = input("\nВаш ход (К/Н/Б/Q): ").upper()
#         if player_move == 'Q':
#             print("Игра завершена.")
#             break
#         if player_move not in ['К', 'Н', 'Б']:
#             print("Некорректный ввод! Используйте К, Н или Б.")
#             continue
#
#         # если накопилось достаточно истории, делаем предсказание
#         ai_move = None
#         if len(game_history) >= history_size:
#             encoded = []
#             for m, o in game_history[-history_size:]:
#                 onehot_my = [0, 0, 0]
#                 onehot_my[move_to_int[m]] = 1
#                 encoded.extend(onehot_my)
#
#                 onehot_opp = [0, 0, 0]
#                 onehot_opp[move_to_int[o]] = 1
#                 encoded.extend(onehot_my)
#
#             prediction = predict(np.array([encoded]))[0]
#             ai_move = int_to_move[prediction]
#         else:
#             # если истории мало случайный ход
#             ai_move = random.choice(['К', 'Н', 'Б'])
#
#         result = "Ничья"
#         if (player_move == 'К' and ai_move == 'Н') or \
#             (player_move == 'Н' and ai_move == 'Б') or \

```

```

#         (player_move == 'Б' and ai_move == 'К'):
#             result = "Вы победили!"
#     elif (ai_move == 'К' and player_move == 'Н') or \
#           (ai_move == 'Н' and player_move == 'Б') or \
#           (ai_move == 'Б' and player_move == 'К'):
#         result = "ИИ победил!"
#
#     game_history.append((player_move, ai_move))
#
#     print(f"\nИИ выбрал: {ai_move}")
#     print(f"Результат: {result}")
#     print(f"История (последние {min(3, len(game_history))} раунда):")
#     for i, (p, a) in enumerate(game_history[-3:]):
#         print(f"Раунд {len(game_history) - 3 + i + 1}: Вы - {p}, ИИ - {a}")
# play vs ai()

```

Приложение 16 – листинг нейросети

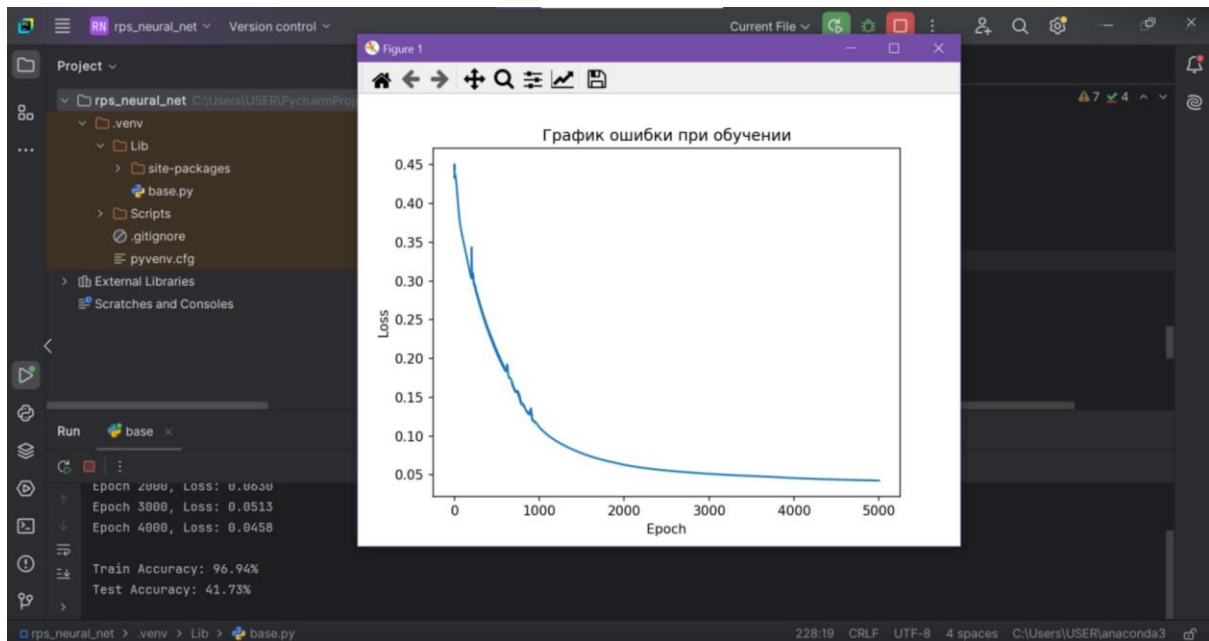


Рисунок 16 - вывод при запуске