

Федеральное государственное автономное образовательное учреждение высшего
образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий
Кафедра «Информационные системы и технологии»

Направление подготовки: 09.03.02 АСОиУ

ОТЧЕТ

по проектной практике

Студент: Козак Юлия Андреевна 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 | 3 |
| Общая информация о проекте | 4 |
| Общая характеристика деятельности организации | 4 |
| Описание задания по проектной практике..... | 6 |
| Описание задания по проектной практике..... | 9 |
| Базовая часть..... | 9 |
| Взаимодействие с организацией-партнёром | 11 |
| Вариативная часть | 15 |
| Заключение | 20 |
| Список литературы..... | 21 |

Введение

Отчет представляет результаты проектной практики, выполненной в рамках образовательной программы по направлению «Информатика и информационные технологии». Практика проводилась на базе кафедры «Информатика и информационные технологии» Московского Политехнического университета и была направлена на разработку веб-сайта в соответствии с дисциплиной «Проектная деятельность», а также на изучение способов разработки telegram-ботов при помощи Phyton в рамках вариативной части задания.

Общая информация о проекте

Название проекта: Компьютерная игра для одного человека «ECollapse» (I курс).

Цель проекта: Создание игры с процедурно генерируемым миром, где игроку предстоит выживать, адаптируясь к постоянно меняющимся условиям, эволюционировать и взаимодействовать с уникальными экосистемами, предоставляя игроку бесконечно разнообразный опыт.

Задачи проекта:

1. Дизайн концепции игры
2. Разработка базового прототипа
3. Создание системы эволюции
4. Внедрение экосистем и симуляции
5. Добавление уникальных условий
6. Тестирование и балансировка
7. Разработка финальной версии

Общая характеристика деятельности организации

В паспорте проекта заказчик отсутствует, но фактически им можно считать Московский Политех, факультет информационных технологий

Структура организации:

ФИТ готовит специалистов в области программирования, искусственного интеллекта, кибербезопасности, анализа данных и других IT-направлений. Факультет сочетает фундаментальное образование с практико-ориентированным подходом, сотрудничая с ведущими IT-компаниями.

Работа сотрудников:

1. *Преподаватели и научные работники* ведут лекции, семинары и лабораторные работы, руководят студенческими проектами и стартапами.

2. *Администрация* организует учебный процесс (расписание, документация), координирует практики и стажировки, обеспечивает работу IT-инфраструктуры факультета

3. *Сотрудники базовых кафедр* (Яндекс, СберТех, Лаборатория Касперского) курируют реальные проекты студентов

Обучение студентов:

1. Основные направления

2. Практика:

- Стажировки в IT-компаниях (с 1–2 курса)
- Участие в хакатонах и конкурсах
- Работа в студенческих лабораториях

Описание деятельности Московского политеха, факультета информационных технологий:

Факультет информационных технологий (ФИТ) Московского Политеха — это современный образовательный центр, готовящий специалистов для цифровой экономики. Факультет реализует программы бакалавриата, магистратуры и аспирантуры по ключевым IT-направлениям: разработка ПО, искусственный интеллект, кибербезопасность, большие данные и компьютерные системы. Обучение сочетает фундаментальную подготовку с практикой — студенты работают с реальными проектами в партнерстве с ведущими компаниями (Яндекс, СберТех, Лаборатория Касперского).

ФИТ активно развивает научные исследования в области машинного обучения, анализа данных и защиты информации. Студенты получают доступ к современным лабораториям, участвуют в хакатонах и стартап-проектах. Факультет также предлагает программы двойных дипломов с зарубежными вузами и курсы от IT-корпораций, что позволяет выпускникам строить успешную карьеру в российской и международной IT-индустрии.

Описание задания по проектной практике

Задачи, выполненные в рамках проектной деятельности.

Участие в работе проектной команды:

В рамках проектной деятельности по разработке компьютерной игры с процедурной генерацией и элементами выживания я принимал активное участие в работе команды из 28 человек. Моя роль заключалась в следующем:

1. Аналитическая часть. Создание и проведение опроса для анализа целевой аудитории.

Данная задача выполнялась в команде из пяти человек. Мы обсудили, на что важно сделать акцент, чтобы полученная информация была наиболее полезной и доступной, ведь её после нас анализировала другая группа. Затем каждый предлагал вопрос, который считает важным задать. Таким образом был сформирован финальный вариант опросника.

2. Техническая часть. Зависимость камеры от игрока.

Данная задача выполнялась в команде из трех человек. Было важно, чтобы все команды справились с поставленными задачами, поэтому тех, кто разбирался в теме лучше, ставили с теми, кто разбирается хуже. Таким образом происходил обмен знаниями, что полезно для каждого человека в команде.

В нашу задачу входило: прописать движение персонажа (ходьба, бег, прыжки) и закрепить камеру так, чтобы она следовала за героем. Первым делом мы посмотрели обучающие ролики и обсудили способы реализации, а затем приступили к работе. Каждый раз, когда у кого-то возникал вопрос, остальные помогали решить проблему. Работа была выполнена всего за пару дней благодаря сплоченности команды

Планирование.

Так как над проектом работает довольно много человек, абсолютно каждая задача выполнялась в группе. Для того, чтобы точно успеть в установленные сроки, я и ребята из моей группы списывались сразу, когда получали задачу, чтобы выбрать дни и время для ее выполнения. Это помогало собрать всю команду сразу, избегая наложения планов.

Что касается лично моей работы, я разбивала задачу на подзадачи, чтобы она не казалась неподъемной. Также делала перерыв на 10-15 минут раз в полтора часа, чтобы не слишком уставать от работы. Планы записывала в заметки телефона и ставила напоминания.

Проблемы, с которыми столкнулась, и решения.

1. Не укладывалась в то время, на которое изначально рассчитывала

Решение: стала выделять время с небольшим запасом на случай непредвиденных ситуаций

2. Пропадающая мотивация

Решение: делилась с близкими прогрессом работы, получала поддержку с их стороны

Участие в проекте позволило:

1. Улучшить навыки командной работы.

В процессе работы мы много дискутировали, учились правильно аргументировать свои предложения и приходить к единому мнению без конфликтов.

2. Укрепить технические навыки:

Я изучила основные возможности Unity и научилась применять их в игровой разработке.

Итог

Мои soft skills сильно повысились за время работы в команде. Я научилась конструктивной обратной связи с участниками команды, активно участвовала во всех обсуждениях и мозговых штурмах с аргументацией позиций. Научилась лучше планировать свое время и правильно расставлять приоритеты.

Описание задания по проектной практике

Базовая часть

Создание и структура репозитория.

На платформе GitHub был создан публичный репозиторий для удобного управления версиями проекта. Его структура включает:

- README.md — основной файл с описанием практики;
- reports — отчёт по практике;
- site — исходный код сайта.

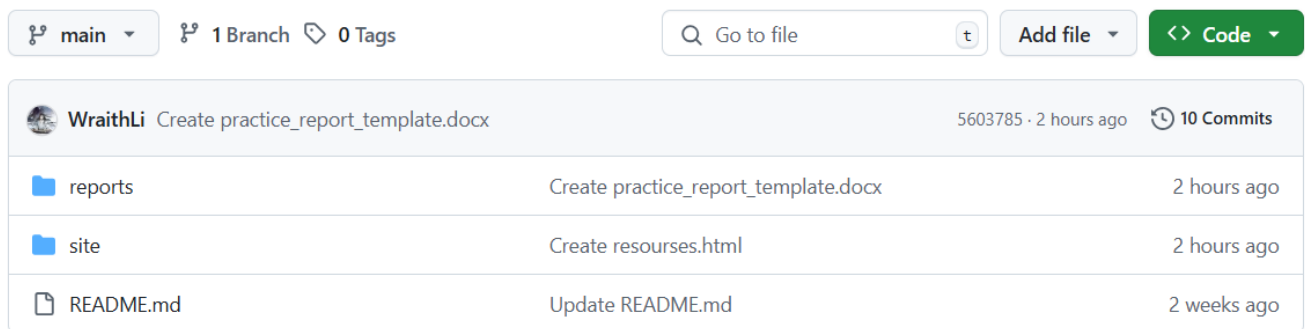


Рисунок 1 - мой репозиторий

Работа с Git: фиксация изменений.

Каждое значимое изменение фиксируется (commit) в три этапа:

1. Добавление в индекс (git add).

- Чаще всего используется `git add .` (добавляет все изменённые файлы).

2. Создание коммита (git commit -m "описание").

- Комментарий помогает быстро находить нужные версии через `git log`.

3. Отправка на удалённый репозиторий (git push).

- Вместо `git push --all origin` использовалась более точная команда `git push origin branch:branch` для избежания конфликтов.

Документация в Markdown.

Все текстовые файлы проекта (.md) написаны в Markdown — удобном формате для статей, отчётов и документации. README.md — ключевая информация о практике.

Создание статического веб-сайта.

Работа состояла из нескольких этапов:

1. Сбор информации
2. Подбор цветов и стилей
3. Написание сайта

Цвета и стили

Для сайта были выбраны кофейные оттенки, потому что они хорошо подходят под концепцию разрабатываемой игры и довольно приятны глазу.

Все кнопки, ссылки и карточки реагируют на наведение мыши.

Присутствует адаптивная верстка для корректного отображения сайта на различных устройствах.

Все углы скруглены, что делает интерфейс мягче и соответствует современным трендам.

Шапка не исчезает при скроллинге.

Взаимодействие с организацией-партнёром

Встреч с организацией-партнером не было, зато была экскурсия по павильону «Умный дом на ВДНХ, что соответствует тематике IT.

Билеты были взяты заранее, поэтому нас встретил гид, который рассказал о концепции «умного дома» и «умного города», их значении в современном мире. Затем он вкратце рассказал, какие технологии будут продемонстрированы.

Основная часть экскурсии

Экскурсия проходит в нескольких тематических зонах, где можно не только увидеть, но и потрогать, попробовать, поуправлять технологиями.

Начали мы с зоны 5G, в которой также есть игровая с ретро-компьютерами. Здесь вы после экскурсии можете в мир развлечений и игр 90-ых, такие как Супер-Марио, Диггер, Змейка и другие.



Рисунок 2 - компьютер из игровой зоны

По пути в следующую зону была Фотозона «Цифровые технологии Москвы», посвященная 30-летию Рунета, где мы тоже остановились, чтобы увидеть, насколько же быстро развиваются технологии.



Рисунок 3 - ретро половина



Рисунок 4 - современная половина

Следующая локация – «Интеллектуальный транспорт».

Здесь нам показали на примере цифровых двойников, как развивалось метро Москвы, а ещё мы лично протестировали систему биометрической оплаты проезда.



Рисунок 5 - турникет метро

В «Цифровом здравоохранении» экскурсовод познакомил с тем, как цифровые технологии меняют сферу здравоохранения. Мы рассмотрели три ключевых раздела:

- искусственный интеллект (ИИ), который помогает врачу устанавливать диагноз;
- единая медицинская информационно-аналитическая система (ЕМИАС);
- мобильное приложение «ЕМИАС.ИНФО».

ЕМИАС объединяет все медицинские учреждения города, предоставляя доступ к электронным медицинским картам и упрощает взаимодействие между пациентами и врачами. А приложение «ЕМИАС.ИНФО» позволяет быстро получить информацию о своем здоровье и записаться на прием.

В интерактивной зоне представлен анатомический стол «Пирогов», с помощью которого можно было изучить строение человеческого организма в 3D-формате.



Рисунок 6 - стол "Пирогов"

Выставка «Цифровые технологии Москвы. К 30-летию Рунета».

Здесь расположен таймлайн с интерактивными экспонатами, которые погружают посетителей в историю развития Рунета и изменений московских сервисов в формате было/стало.



Рисунок 7 - еще один вариант компьютера 90-ых



Рисунок 8 - самая первая версия всем известного сайта mos.ru

Вариативная часть

Разработка telegram-бота на Python

Задачи, которые необходимо выполнить в ходе работы:

1. Ознакомиться с разновидностями telegram-ботов и выбрать тематику для своего.
2. Изучить необходимый для разработки материал.
3. Предоставить готовый продукт.

Разновидности telegram-ботов:

1. Сервисные боты

- Погодные: прогноз на день/неделю, данные о влажности, ветре.
- Конвертеры валют: актуальные курсы, отслеживание изменений.
- Переводчики: быстрый перевод слов и текстов.

2. Бизнес-боты

- CRM-боты: автоматизация работы с клиентами, интеграция с CRM.
- Для интернет-магазинов: обработка заказов, платежей, рекомендации товаров.

3. Образовательные боты

- Тьюторы: уроки, тесты, контроль прогресса.
- Энциклопедии: быстрый поиск информации по запросу.

4. Развлекательные боты

- Игровые: мини-игры, логические задачи.

- Квизы: викторины на разные темы.

5. Информационные боты

- Новостные: подборка актуальных новостей по темам.
- Агрегаторы: сбор данных из разных источников.

6. Технические боты

- Мониторинг серверов: уведомления о сбоях.
- Управление задачами: создание, напоминания, учёт времени.

В качестве задания для вариативной части практики я выбрала развлекательного бота, а именно бота для игры в «Кто я?» Такой выбор обусловлен, в первую очередь, тем, что на проектной деятельности наша команда занимается разработкой игры, так что я решила не отходить от развлекательной тематики. Также данный бот не слишком сложен в реализации, что позволило уложиться в установленные сроки.

Структура проекта

```
├── main.py      # Основной код
├── .env         # Токен бота
└── characters.json # База персонажей
```

Этапы разработки:

| Наименование | Код |
|--------------|-----|
|--------------|-----|

| | |
|--|---|
| <p>Подключение необходимых библиотек</p> | <pre>import json import random from aiogram import Bot, Dispatcher, types from aiogram.filters import Command, StateFilter from aiogram.fsm.state import State, StatesGroup from aiogram.fsm.context import FSMContext from aiogram.fsm.storage.memory import MemoryStorage from dotenv import load_dotenv import os</pre> |
| <p>Инициализация</p> <p>Загрузка переменных окружения (токен бота)</p> <p>Загрузка списка персонажей из JSON-файла</p> <p>Настройка хранилища состояний (MemoryStorage)</p> | <pre># Загрузка переменных окружения load_dotenv() # Загрузка персонажей with open("characters.json", "r", encoding="utf-8") as f: CHARACTERS = json.load(f) # Настройка бота TOKEN = os.getenv("BOT_TOKEN") bot = Bot(token=TOKEN) storage = MemoryStorage() dp = Dispatcher(storage=storage) # Состояния игры class GameState(StatesGroup): playing = State()</pre> |
| <p>Основные команды</p> <p>/start - приветственное сообщение</p> <p>/play - начать новую игру</p> <p>/surrender - сдаться (только в игре)</p> <p>/stats - показать статистику побед</p> <p>/help - помощь (только вне игры)</p> | <pre># Команда /start @dp.message(Command("start")) async def start(message: types.Message): await message.answer("👋 Привет! Я бот для игры 'Кто я?' \n\n" "Я загадаю персонажа, а ты должен угадать его, задавая вопросы, " "на которые можно ответить 'Да', 'Нет' или 'Не знаю'. \n\n" "Например: \n" "- Ты человек? \n" "- Ты из фильма? \n" "- Ты существуешь в реальной жизни? \n\n" "Чтобы начать игру, напиши /play") # Команда /play - начало игры @dp.message(Command("play")) async def play_game(message: types.Message, state: FSMContext): await state.set_state(GameState.playing) character = random.choice(CHARACTERS) await state.update_data(character=character,</pre> |

| | |
|---|--|
| | <pre> questions=0) await message.answer(f"🎲 Игра началась! Я загадал персонажа.\n" "Задавай вопросы, на которые можно ответить 'Да', 'Нет' или 'Не знаю'.\n\n" "Чтобы сдать, напиши /surrender") # Статистика @dp.message(Command("stats")) async def show_stats(message: types.Message): wins = user_stats.get(message.from_user.id, 0) await message.answer(f"📊 Твоё количество побед: {wins}") # Работает ВНЕ игры команда help @dp.message(Command("help"), ~StateFilter(GameState.playing)) async def help_cmd(message: types.Message): await message.answer("Помощь: /play - начать игру") </pre> |
| <p>Обработка вопросов</p> <p>Основной игровой цикл анализирует вопросы и дает ответы. Содержит более 20 вопросов</p> | <pre> @dp.message(GameState.playing) async def handle_question(message: types.Message, state: FSMContext): data = await state.get_data() character = data["character"] questions = data["questions"] + 1 question = message.text.lower() # Проверка на команды if question.startswith("/"): return # Обновляем статистику вопросов await state.update_data(questions=questions) # Проверяем, угадал ли игрок персонажа if character.lower() in question: user_id = message.from_user.id user_stats[user_id] = user_stats.get(user_id, 0) + 1 await message.answer(f"🎉 Правильно! Это {character}!\n" f"Ты угадал за {questions} вопросов.\n\n" f"Всего побед: {user_stats[user_id]}\n" f"Сыграем ещё? /play") await state.clear() return </pre> |

| | |
|--------------------|---|
| | <pre> # Логика ответов на вопросы (только если персонаж не угадан) answer = "Не знаю" if "человек" in question: answer = "Да" if character in ["Леон Кеннеди", "Шерлок Холмс", "Джульетта", "Тони Старк", "Карлсон", "Капитан Джек Воробей", "Джокер", "Человек-паук", "Геральт", "Рапунцель"] \ else "Нет" elif "животное" in question or "зверь" in question or "кот" in question: answer = "Да" if character in ["Кот в сапогах"] else "Нет" elif "мульт" in question or "мультик" in question: answer = "Да" if character in ["Губка Боб Квадратные Штаны", "Шрек", "Рапунцель", "Кот в сапогах", "Карлсон"] else "Нет" await message.answer(answer) </pre> |
| Запуск бота | <pre> if __name__ == "__main__": dp.run_polling(bot, skip_updates=True) </pre> |

Данный проект был модифицирован: добавлены подсказки. Если они становятся необходимы игроку, ему достаточно воспользоваться командой [/hint](#)

Заключение

В ходе проектной практики были успешно выполнены все поставленные задачи, включая базовую и вариативную части. Освоены ключевые технологии веб-разработки: работа с Git для контроля версий, создание структурированного репозитория, оформление документации в Markdown, а также разработка адаптивного веб-сайта с использованием HTML и CSS.

Особое внимание было уделено созданию telegram-бота: изучены их виды и особенности. Приобретены навыки работы с состояниями и командами. Это позволило глубже понять все тонкости разработки ботов.

Практика также способствовала развитию навыков планирования и самообучения. В результате был создан функциональный прототип сайта по игре «ECollapse», отвечающий современным требованиям юзабилити и дизайна.

Приобретённый опыт в области веб-разработки и написании бота с нуля станет прочной основой для дальнейшего роста в сфере информационных технологий .

Список литературы

1. GitHowTo URL: <https://githowto.com/ru> (дата обращения: 10.05.2025).
2. Бесплатный учебник по Git и GitHub // HTML academy URL: https://htmlacademy.ru/blog/html_old (дата обращения: 10.05.2025).
3. Язык разметки Markdown: шпаргалка по синтаксису с примерами // Skillbox URL: <https://skillbox.ru/media/code/yazyk-razmetki-markdown-shpargalka-po-sintaksisu-s-primerami/> (дата обращения: 20.05.2025).
4. Основы CSS // MDN Web Docs URL: https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Styling_the_content (дата обращения: 15.05.2025).
5. Основы HTML // MDN Web Docs URL: https://developer.mozilla.org/ru/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content (дата обращения: 15.05.2025).
6. Изучение HTML: руководства и уроки // MDN Web Docs URL: https://developer.mozilla.org/ru/docs/Learn_web_development/Core/Structuring_content (дата обращения: 17.05.2025).
7. CSS styling basics // MDN Web Docs URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics (дата обращения: 17.05.2025).
8. Руководство по Python для создания Telegram-ботов // Python Telegram Bot URL: <https://python-telegram-bot.readthedocs.io> (дата обращения: 19.05.2025).
9. Обучающий курс по созданию Telegram-ботов. // Udemу URL: <https://www.udemy.com/course/telegram-bots> (дата обращения: 19.05.2025).