

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Северский технологический институт –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(СТИ НИЯУ МИФИ)

Проект

Разработка компьютерной игры.

Северск 2025

1. Цель проекта

Освоить полный цикл разработки программного обеспечения от проектирования и кодирования до контроля версий и публикации.

2. Технологический стек:

- Язык: **Python** с виртуальным окружением **venv**.
- Фреймворк: **Pygame**.
- Среда разработки: **VS Code**.
- Контроль версий: **Git / GitHub**.

3. Основная часть.

Аннотация

Данный проект посвящен детальному изучению процесса разработки компьютерной игры «Змейка» на языке программирования Python с применением библиотеки Pygame.

Работа охватывает весь жизненный цикл проекта, включая:

- Проектирование;
- Реализация;
- Тестирование;
- Размещение готового приложения на платформе GitHub.

Особое внимание уделено использованию интегрированной среды разработки Visual Studio Code (VS Code).

Содержание итогового отчёта

1. Отчёт по лабораторной работе должен быть оформлен в соответствии с требованиями СТИ НИЯУ МИФИ.

2. Отчет по проделанной работе готовится в текстовом редакторе и предоставляется для проверки в электронном виде в формате электронного документа. В отчете должны быть принципиальные и структурные схемы исследуемых элементов и звеньев, структурные схемы моделирования и графики переходных процессов, построенные по результатам моделирования, а также необходимые выводы по работе.

3. Отчет содержит 8 практических работ, каждая из которых посвящена отдельному этапу разработки игры «Змейка»: от планирования и проектирования до финального тестирования и публикации проекта. Каждая практическая работа сопровождается скриншотами, описанием хода работы.

[**<Итоговый_отчёт_№группы-ФИО>.docx**](#)

[**<Итоговый_отчёт_№группы-ФИО>.pdf**](#)

[**<Итоговый_отчёт_№группы-ФИО>.pptx**](#)

ПРИМЕР:

[**<Итоговый_отчёт_295-Иванов ИИ>.docx**](#)

[**<Итоговый_отчёт_295-Иванов ИИ>.pdf**](#)

[**<Итоговый_отчёт_295-Иванов ИИ>.pptx**](#)

4. • Каждая практическая работа оформляется в отдельной ветке (pr1, pr2, ...), после проверки — мердж в main.

5. В отчете указывать ссылку на коммит, скриншоты, ключевые фрагменты кода и пояснения.

Содержание проекта

Практическое задание №1 Проектирование проекта: структурная и функциональная схемы.

Практическое задание №2 Настройка среды разработки и подготовка к проекту.

Практическое задание №3 Создание базовой структуры проекта.

Практическое задание №4 Игра генератор случайных чисел.

Практическое задание №5 Реализация дополнительного функционала проекта.

Практическое задание №6 Создание дистрибутива.

Практическое задание №7 Документирование проекта и размещение в репозитории GitHub

Практическое задание №1

Название:

Проектирование проекта: структурная и функциональная схемы.

Цель:

- Изучение принципов проектирования;
- Формулирование требований к проекту;
- Составление структурной и функциональной схем.

Задание:

1. Ознакомится с техническим заданием игры «Змейка» и основными требованиями к ее реализации.
2. Составить структурную схему проекта (какие файлы, модули, классы будут в проекте).
3. Составить функциональную схему (основные функции, их взаимодействие).

Шаги:

1. Изучить информацию из технического задания
2. Разработать структурную схему (пример, в виде блок-схемы или списка модулей: main.py, game.py, snake.py, food.py, и т.д.).
3. Разработать функциональную схему (основные функции: инициализация игры, главный цикл, обработка событий, движение змейки, генерация еды, проверка столкновений, отрисовка).

Отчет:

Раздел отчёта. Практика №1. Настройка среды разработки и подготовка к проекту.

Практическое задание №2

Название:

Настройка среды разработки и подготовка к проекту.

Цель:

Установить и настроить инструменты для разработки (Python, VS Code, Pygame), создать репозиторий на GitHub.

Задание:

1. Установить Python 3.10+ и VS Code.
2. Настроить виртуальное окружение (venv).
3. Установить библиотеку pygame через pip.
4. Создать репозиторий на GitHub с названием snake-game и инициализировать локальный проект.

Шаги:

1. Проверить версию Python: `python --version`.
2. Установить зависимости: `pip install pygame`.
3. Добавить в `.gitignore` файлы виртуального окружения и кэша.
4. Сделать первый коммит с описанием "Initial setup".

Отчет:

Раздел отчёта. Практика №2. Настройка среды разработки и подготовка к проекту.

Скриншоты установки Python/VS Code, вывод терминала с установкой Pygame, ссылка на GitHub-репозиторий. Активация виртуального окружения. Код файлов, скриншот запущенного окна, коммит в GitHub с описанием «Initial setup».

Практическое задание №3

Название:

Создание базовой структуры проекта.

Цель:

Реализовать начальную структуру проекта и подключить Pygame.

Задание:

1. Создать файлы:
 - src/main.py (запуск игры),
 - src/config.py (настройки: SCREEN_WIDTH, SNAKE_SPEED),
 - src/game.py (класс Game с методами init, run).
2. Реализовать инициализацию окна Pygame.

Шаги:

- В main.py импортировать класс Game из game.py.
- Добавить обработку события закрытия окна (кнопка "X").
- Проверить запуск окна без логики игры.

Отчет:

Раздел отчёта. Практика №3. Создание базовой структуры проекта.
Код файлов, скриншот запущенного окна, коммит в GitHub с описанием «Basic project structure».

Практическое задание №4

Название:

Игра генератор случайных чисел.

Цель:

Добавить в git игру генератор случайных чисел.

Задание:

Шаги:

Отчет:

Раздел отчёта. Практика №4. Игра генератор случайных чисел.

Код файлов, скриншот запущенного окна, коммит в GitHub с описанием «Random Number Generator».

Практическое задание №5

Название:

Реализация дополнительного функционала проекта.

Цель:

Добавить обработку столкновений (со стенами и с собой), подсчет очков и управление скоростью.

Задание:

1. Реализовать класс `Snake` с атрибутами:
 - `body` (список координат сегментов),
 - `direction` (вектор движения).
2. Добавить обработку клавиш (WASD или стрелки).
3. Организовать обновление позиции змейки в цикле.
4. Реализовать проверку столкновений змейки с границами экрана и с собой.
5. Реализовать подсчет очков и их отображение.

Шаги:

1. В классе `Snake` добавить метод `check_collision`, который проверяет, не столкнулась ли змейка с границами или с собой.
2. В главном цикле при столкновении завершать игру.
3. Добавить переменную счета, увеличивать ее при поедании еды и отображать на экране.

Отчет:

Раздел отчёта. Практика №5. Создание базовой структуры проекта.
Код файлов, скриншот запущенного окна, коммит в GitHub с описанием «Basic project structure».

Практическое задание №6

Название:

Создание дистрибутива.

Цель:

1 Научиться создавать самостоятельные исполняемые файлы (.exe) из Python-скриптов с использованием инструментов PyInstaller и Auto PY to EXE, что позволяет запускать Python-приложения на компьютерах без установленного интерпретатора Python.

Задание:

Установка дополнительных пакетов:

Перед началом работы необходимо установить следующие пакеты:

```
pip install pyinstaller  
pip install auto-py-to-exe
```

Задание 1: Подготовка тестового Python-скрипта

Задание 2: Создание EXE файла с помощью PyInstaller

Задание 3: Создание EXE файла с помощью Auto PY to EXE

Задание 4: Создание EXE файла для приложения с внешними зависимостями.

Задание 5: Создание EXE файла для вашего проекта «Змейка».

Отчет:

Раздел отчёта. Практика №5. Создание дистрибутива.

Практическое задание №7

Название:

Документирование проекта и размещение в репозитории GitHub.

Цель:

Опубликовать проект с полной документацией.

Задание:

1. Подготовить презентацию по шаблону.

Отчет:

Ссылка на GitHub-репозиторий со всеми файлами, содержимое README.md, коммит «Final release».