**Электронный конспект**по игре на pygame  
выполнила студентка группы ИСиП 23-01-1п Беспрозванных Ю.А.

1. **Особенности библиотеки Pygame, изученные в ходе работы над проектом**

В ходе работы над проектом были изучены следующие особенности библиотеки Pygame:

* Инициализация и настройка окна игры:
* Использование pygame.init() для инициализации всех модулей Pygame.
* Создание игрового окна с помощью pygame.display.set\_mode().
* Управление частотой кадров с помощью pygame.time.Clock().

Работа с изображениями:

* Загрузка изображений с помощью pygame.image.load().
* Использование convert() и convert\_alpha() для оптимизации отрисовки изображений.
* Размещение изображений на экране с помощью screen.blit().

Работа с текстом:

* Создание шрифтов с помощью pygame.font.Font().
* Рендеринг текста с помощью render().
* Размещение текста на экране с использованием прямоугольников (get\_rect()).

Обработка событий:

* Использование pygame.event.get() для получения событий (например, нажатия клавиш или закрытия окна).
* Обработка нажатий клавиш с помощью pygame.key.get\_pressed().
* Коллизии и взаимодействие объектов:
* Проверка столкновений с помощью colliderect().
* Логика взаимодействия объектов (например, столкновение игрока с предметами).

Таймеры и время:

* Использование pygame.time.get\_ticks() для отслеживания времени.
* Управление временем для увеличения сложности игры (например, ускорение объектов).

1. **Структура проекта в файловой системе на диске**

Проект представляет собой папку, которая содержит следующие элементы:

* Основной файл скрипта игры (например, main.py): Содержит весь код игры, включая инициализацию, игровой цикл, обработку событий и отрисовку.
* Папка с ресурсами (например, assets): Изображения: detective.png, teapot.png, candlestick.png, wooden\_box.png, code\_game\_back.jpg, code\_game\_back\_floor.jpg.
* Шрифты: prstartk.ttf.
* README.md (опционально):
* Описание проекта, инструкции по запуску, список зависимостей.
* Файл зависимостей (например, requirements.txt):
* Список библиотек, необходимых для запуска проекта (например, pygame).

1. **Виртуальное окружение: что это, как и когда его нужно создавать?**

Виртуальное окружение — это изолированная среда для разработки Python-проектов, которая позволяет управлять зависимостями проекта отдельно от глобальных установленных пакетов.

Зачем нужно:

* Чтобы избежать конфликтов версий библиотек между проектами.
* Для упрощения переноса проекта на другие компьютеры.

Как создать:

* Установить virtualenv или использовать встроенный модуль venv
* Активировать виртуальное окружение:
* Установить зависимости:

Когда создавать:

* В начале работы над новым проектом.
* При использовании сторонних библиотек.

1. **Особенности использования среды разработки**

Для работы над проектом была выбрана среда разработки VS Code. Её особенности:

Поддержка Python:

* Интеграция с Python через расширение Python.
* Подсветка синтаксиса, автодополнение, отладка.

Управление виртуальным окружением:

* Возможность автоматического выбора интерпретатора из виртуального окружения.

Интеграция с Git:

* Удобное управление версиями проекта.

1. **Приоритеты структуры программного кода**

При создании игрового проекта на Pygame важно соблюдать линейную последовательность:

* Инициализация Pygame и создание окна:
* Загрузка ресурсов (изображения, шрифты).
* Создание игровых объектов и их начальных состояний.
* Основной игровой цикл.
* Обработка событий.
* Обновление состояния игры.
* Отрисовка объектов.
* Завершение игры.
* Обработка выхода из игры.
* Освобождение ресурсов.

1. **Назначение циклов, условий, данных, структур данных и функций**

Циклы:

* while True: — основной игровой цикл, который обновляет состояние игры и отрисовывает кадры.
* Используется для непрерывного выполнения игры.

Условия:

* if event.type == pygame.QUIT: — обработка закрытия окна.
* if keys[pygame.K\_UP]: — обработка нажатий клавиш.
* Используются для управления логикой игры.

Данные и структуры данных:

* Переменные (hero\_rect, current\_speed) — хранение состояния объектов.
* Списки (например, для хранения нескольких объектов).
* Используются для управления состоянием игры.

Функции:

* display\_score() — отображение счёта.
* reset\_game() — сброс состояния игры.
* Используются для организации кода и повторного использования логики.

1. **Что можно или нужно изменить?**

Улучшение читаемости кода:

* Разделить код на модули (например, отдельный файл для ресурсов, для логики игры).
* Добавить комментарии и документацию.

Добавление новых механик:

* Введение новых типов препятствий.
* Добавление бонусов (например, временное замедление объектов).

Оптимизация:

* Использование спрайт-групп для управления объектами.
* Оптимизация загрузки ресурсов.

Улучшение пользовательского интерфейса:

* Добавление меню с настройками.
* Улучшение отображения счёта и других элементов интерфейса.