

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«МИРЭА – Российский технологический университет»**

# РТУ МИРЭА

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ**

по дисциплине

«Технологии программирования в среде Python»

по теме «Создание 2D игр в среде Python»

**Выполнила:** Смолькова М. С.

**Группа:** БСБО-05-18

**Преподаватель:** Тарланов А. Т.

**Москва 2020**

В задании курсовой требовалось реализовать проект на языке программирования Python. В качестве проекта в моём случае было выбрано создание игры, т. к это наиболее оптимальный способ, чтобы познакомиться с особенностями и технологиями данного языка программирования. Выбор пал на аркаду с возможностью сохранения результата, меню, анимацией, сбором и звуковым сопровождением.

В языке программирования Python для создания игр хорошо подходит библиотека Pygame, поэтому именно она была выбрана за основу создания проекта.

После импорта библиотеки для начала работы требовалось инициализировать все модули Pygame с помощью команды pygame.init(), далее можно было приступать к созданию игры.

При создании было задействовано множество модулей данной библиотеки и возможностей языка Python.

1. Загрузка изображений и их обязательная последующая конвертация (благодаря ей всё работает намного быстрее, при выполнении игры это ускоряет отрисовку поверхностей и позволяет избежать лаги)
2. Подключение собственного шрифта, предварительно загруженного в корневую папку проекта.
3. Использование классов для структуризации объектов игры. В игре имеется класс Player, класс Menu, Класс Attack, класс Items (препятствия, монеты). В каждом из них содержатся инициализатор, обычные методы, реализующие конкретные цели, статические методы, которые не требуют создания объекта для вызова.
4. Использование удобного метода для работы с поверхностями get\_rect, который позволил, без проблем реализовать столкновения игрока с различными объектами.
5. Обработка нажатий клавиш с клавиатуры.

В игре реализовано управление персонажем с клавиатуры, в момент нажатия персонаж либо движется вперёд, либо вверх, либо атакует.

1. Обработка нажатий кнопок мыши.

Реализовано управление меню с помощью нажатия левой кнопки мыши. Благодаря использованию метода get\_rect при создании кнопок меню, отработка нажатия на них была создана с помощью проверки левой и правой части кнопки с координатой мыши по оси X и верхней и нижней части с координатой мыши по оси Y.

1. Звуковое сопровождение.

В игре присутствует музыкальное (Стартовое меню и конец игры) и звуковое сопровождение (Сама игра). Музыкальное сопровождение реализовано с помощью метода pygame.mixer.music.load(). Данный метод позволяет загрузить необходимую музыку, так после этого можно установить желаемую громкость воспроизведения и установить количество повторов воспроизведения музыки. Звуки реализованы с помощью метода pygame.mixer.Sound().

1. В игре предусмотрен главный таймер, который ограничивает максимальное число кадров, тем самым обеспечивает плавную игру и контролирует трату системных ресурсов.
2. Реализация движения объектов и персонажа.

Для того чтобы персонаж двигался, при нажатии на клавишу, отвечающую за движение вперёд, начинает двигаться фон и сами объекты, но никак не персонаж. Бесшовный по вертикали фон зациклен, поэтому повторяется бесконечно и создается видимость движения вперёд. Движущиеся объекты создаются за пределами экрана, после того, как они проходят вдоль экрана и становятся невидимыми, их можно удалить, очистив тем самым список созданных объектов.

1. Графический интерфейс и сохранение, сброс результатов

В течении игры пользователь должен знать о состоянии своих показателей, для этого в проекте реализован графический интерфейс персонажа. Шкала здоровья, шкала энергии, собранные монеты, пройденное расстояние. Шкала здоровья падает при столкновении с препятствием. Шкала энергии падает при атаке и постепенно восстанавливается после её окончания.

Собранные монеты суммируются и сохраняются после выхода из игры, так же, как и пройденное расстояние, если побит старый рекорд, новый перезаписывает старый. Для сохранения был создан файл и были использованы библиотеки Python для работы с файлами. Благодаря этому можно продолжить игру после выхода из неё. В игре так же присутствует возможность сброса всех результатов для нового старта.

1. Анимация и создание объектов

Для создания подвижных объектов, в игре были использованы события, происходящие по таймеру, через определённое время (pygame.USEREVENT, pygame.time.set\_timer()). В нашем случае они обновляли картинку с одной на другую по истечении данного времени.

Для создания препятствий и объектов тоже были использованы события, через одинаковый промежуток времени генерировался новый объект.

В итоге, получаем игру, в который у нас есть дракон, который облетает препятствия, поджигает их и собирает монеты, цель игры – продержаться как можно дольше и не умереть.