ПРОЕКТ

«СОЗДАНИЕ 2D ИГРЫ НА PYQT»

ВЫПОЛНИЛ:

ПЕТРУНИН АЛЕКСАНДР МИХАЙЛОВИЧ

СОДЕРЖАНИЕ

1. Введение……………………………………………………………

1.1. Актуальность…………………………………………………….

1.2. Проблема………………………………………………………..

1.3. Цели и задачи……………………………………………………

1.4. Планируемый результат (продукт) …………………………….

2. Основная часть………………………………………………………

2.1.Реализация…………………………………………………………

2.2.Функции приложения…………………………………………………

3. Выводы ……………………………………………………………...

4. Приложения ……………………………………………...…………

* 1. **Актуальность**

Данный проект создан в период изучения нового модуля Pygame. Актуальность этого проекта заключается в его образовательной ценности для автора.

* 1. **Проблема**

Изучение модуля Pygame.

* 1. **Цели и задачи**

Цель: закрепить знания по модулю Pygame в процессе создания творческого проекта, улучшить навыки программировнаия.

Задачи:

1. Продумать концепт простой 2D игры;
2. Подготовить визуальное оформление;
3. Подготовить музыкальное сопровождение;
4. Создать классы основных объектов и событий игры;
5. Создать интерфейс для взаимодействия пользователя;
6. Сформировать основной игровой цикл;
   1. **Планируемый продукт**

2D игра

**2.1 Реализация**

## 2.1.1 Краткие сведения

В качестве основы для игры было принято взять 2D шутер от 3го лица во вселенной Звездных Войн, где главным объектом является космический корабль, цель которого – перевезти данные из пункта А в пункт Б.

Препятствием для игрока является динамическое астеройдное поле.

Игрок располагает простым управлением корабля в виде клавиш (W,A,S,D) для передвижения, ЛКМ для стрельбы стандартными лазерами и комбинацией W+ПКМ для усиленной атаки.

**2.1.2 Визуальное и аудио оформление**

Текстуры корабля и астеройдов были найдены в сети Интернет в открытых источниках.

Изображения бластерных выстрелов, кнопки меню, стартовый и конечный экраны, анимации полета выполнены с помощью графического редактора GIMP.

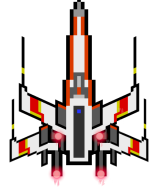
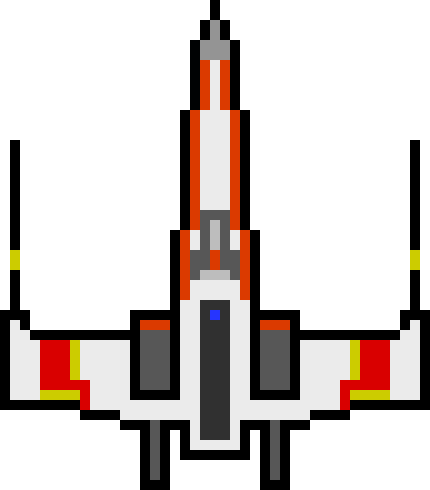
Аудиодорожки были взяты из открытых источников и обработаны в звуковом редакторе. С помощью него производилась оптимизация звука выстрелов, а также сведение фоновых музыкальных композиций.

Модуль Pygame требует различные аудио форматы для звуков и музыки. Для конвертирования форматов аудио файлов использовался функционал того же редактора.

**2.1.3 Основные классы**

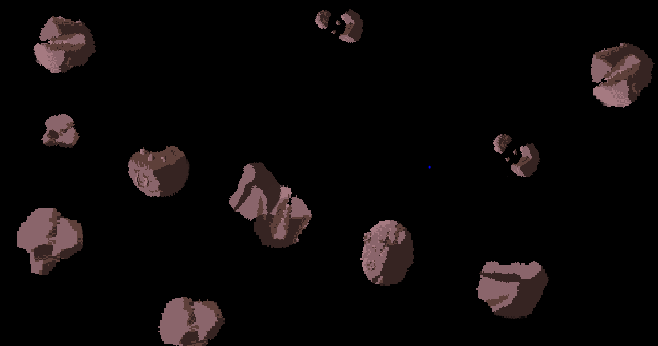
1. Класс «Hero»

Объект этого класса инициализируется при старте игры. Основными методами класса являются передвижение, перезарядка, стрельба, получение урона.



1. Класс «Asteroid»

Объект этого класса инициализируется при наступлении определенной фазы в игре. Наносит урон игроку. Взрывается при достаточном повреждении.





1. Класс «Asteroid wave»

Объект этого класса представляет собой «контейнер-генератор» астеройдов с методом update() для обновления состояния сгенерированных объектов класса «Asteroid».

1. Класс «Shot»

Объект этого класса –снаряд с определенной мощностью и типом заряда. Он генерируется в методе класса «Hero» - shooting().

Все выше описанные классы наследуются от класса Sprite модуля Pygame.

1. Класс «MeasureLine»

Объект этого класса является информационной полосой, необходимой для информирования игрока о текущих состояниях прочности корабля, щита и проценте выполнения задания.

линия.PNG

1. Класс «Button»
2. Класс «Menu»

Эти классы создают интерфейс, отслеживают клики игрока, выполняют необходимые игровые действия (старт, пауза, новая игра)



****

**2.1.4 Игровой цикл**

В игровом цикле перед началом самой игры вызываются функции prepare\_field() и update\_lvl().

Функция prepare\_field удаляет все возможные спрайты из массивов и создает новые объекты классов Hero и AsteroidWave. Здесь включается фоновое музыкальное сопровождение.

Функция update\_lvl() принимает в аргумент данные об уровне и переключает фазы игры. Информация о тестовом уровне представлена в виде строки из символов “a”(фаза астеройдного поля), “-”(пустая фаза), “X”(конец игры).

В течение игрового цикла обрабатываются взаимодействия игрока с клавиатурой и мышью, вызываются основные методы главных объектов, срабатывает таймер на обновление фазы уровня, при необходимости вызывается функция отображения интерфейса.

Для выхода из игрового цикла и игры в целом создана функция terminate().

**3.Выводы**

Мной была создана 2D игра. Планируемый продукт получен. Цель проекта достигнута, так как в процессе его создания я закрепил и развил свои навыки программирования, изучил новый модуль, научился оценивать и решать сложные задачи.

Проект можно развивать, добавляя новых врагов и увеличивая вариативность комбинаций фаз игры.