

Пояснительная записка к проекту

Название проекта: «Космическая баталия»

Автор проекта: Котов Сергей Валерьевич, группа Бсбо-07-22

Запускающий файл: main.py / Космическая баталия.exe

Проект представляет собой игру, сделанную с помощью библиотеки pygame. Игра является 2-D шутером с видом сверху в стилистике старых пиксельных игр.

Использованные библиотеки: screeninfo, pygame, random, time, os.

В проекте присутствуют 5 классов интерфейсов, 5 классов сущностей и 4 прочих класса.

Реализованный функционал:

- Основной игровой цикл, включающий 7 игровых состояний;
- анимированный задний фон, звуковое сопровождение;
- адаптивный размер экрана;
- система сохранения результатов игры;
- доска рекордов;
- система набора очков, основанная на стадии игры;
- пауза с возможностью досрочного прекращения игры;
- анимация смерти игрока;
- прогрессия в игровом процессе: улучшение корабля игрока, увеличение количества врагов и скорости их появления.

Классы игровых сущностей наследуются от основного класса BaseEntity (см. рис. 1), являющегося pygame спрайтом. В методе инициализации задается изображение и размер спрайта, маска для считывания столкновений и скорость сущности. Также в классе реализован метод перемещения по смещению. Все сущности попадают в базовую группу спрайтов сущностей, которая очищается при завершении игрового процесса.

```

BaseSpriteGroup = pygame.sprite.Group()

class BaseEntity(pygame.sprite.Sprite):
    def __init__(self, pos: tuple, speed: float, image="Images/Enemy.png", xScaleMultiplier=70,
                  yScaleMultiplier=80, spriteGroup=None):
        if spriteGroup is not None:
            super().__init__(spriteGroup, BaseSpriteGroup)
        else:
            super().__init__(BaseSpriteGroup)
        tt = TIME_TICK / 10
        self.posX = pos[0]
        self.posY = pos[1]
        self.speedX = speed / tt * SCALE_X
        self.speedY = speed / tt * SCALE_Y

        self.image = pygame.image.load(image)
        self.image = pygame.transform.scale(self.image, (xScaleMultiplier*SCALE_X, yScaleMultiplier*SCALE_Y))

        self.rect = self.image.get_rect()
        self.rect.center = self.posX, self.posY
        self.mask = pygame.mask.from_surface(self.image)

    def move(self, movement):
        self.posX += movement[0]
        self.posY += movement[1]
        self.rect.center = self.posX, self.posY

```

Рисунок 1: Родительский класс игровых сущностей
Описание игрового процесса

При запуске игры, пользователь попадает на экран описания управления, затем переходит в меню. В меню игрок может выйти из игры, начать новую игру или посмотреть 5 лучших игровых рекордов.

Во время игры, пользователь может перемещаться по карте, уклоняясь от вражеских снарядов и стрелять по появляющимся врагам. У игрока 3 жизни, потеря которых приведет к завершению игры. Убивая определенное количество врагов, игрок улучшает свой корабль (повышает скорость стрельбы). Улучшать корабль возможно до пяти раз. При получении урона стадия улучшения откатывается на 1 назад вплоть до первой стадии. Счетчик убийств для улучшения так же сбрасывается. Количество врагов со временем увеличивается. Набор очков вычисляется по количеству уничтоженных противников и стадии игры, на которой они были уничтожены.

По окончании игры, игрок будет перемещен на экран окончания игры, где может ввести свое имя и выйти в меню. Если результат игрока попадает в топ-5 лучших, то он сохраняется для доски рекордов.