

| PYGAME.

ЦЕЛЬ: познакомиться с встроенными классами `Sprite` и `Group`

ПРИМЕЧАНИЕ:

ПЛАНИРОВАНИЕ

1. Встроенный класс `Sprite`

В библиотеке Pygame есть свои встроенные классы, то есть они уже прописаны там разработчиками, которые написали библиотеку pygame. И один из них — это класс `Sprite`. В нем есть несколько полезных нам инструментов. Поэтому сделаем так, чтобы наш класс `Sprite` был наследником встроенного класса `Sprite`.

! Проблемная задача: что изменить в коде программы, чтобы наш уже существующий класс `Sprite` стал наследником встроенного класса?

Ответ: нужно изменить одну строчку кода:

```
class Sprite(pygame.sprite.Sprite):
```

Теперь наш класс `Sprite` наследник встроенного класса `Sprite`. Ну и, наверное, нужно в нашем конструкторе вызвать конструктор класса — родителя, то есть встроенного класса `Sprite`:

```
class Sprite(pygame.sprite.Sprite):  
  
    def __init__(self, x, y, speed, img):  
  
        pygame.sprite.Sprite.__init__(self)
```

Возможно, вы зададитесь вопросом для чего это все нужно? Далее будет понятно.

2. Встроенный класс `Group`

В pygame есть очень полезный класс, который описывает такой объект, как группа спрайтов. Полезность этого класса в том, что можно одним методом — **`update()`** объекта класса `Group` выполнить этот метод у всех спрайтов, состоящих в этой группе. Например, в игре, где несколько спрайтов-врагов, можно все эти спрайты объединить в группу, чтобы одной строчкой в программе запускать выполнение какого-нибудь метода у всех спрайтов-врагов.

Создадим в нашей программе такой объект встроенного класса `Group`, пускай это будет группа спрайтов, в которой будут состоять все спрайты — платформы:

```
GroupPlatform = pygame.sprite.Group()
```

Чтобы добавить спрайт в группу используется метод — **`add`**

```
GroupPlatform.add(Ground)
```

`Ground` — объект встроенного класса `Sprite` (наши платформы в игре)

Метод – **add()** есть у объекта **Sprite**

Ground.add(GroupPlatform) - добавляет спрайт в группу.

Теперь рассмотрим, как можно вызвать метод у группы, и он вызовет этот же метод у всех спрайтов в этой группе.

У объекта класса Group есть метод **update()** – он вызывает такой же метод **update()** у каждого спрайта в группе.

В методе update() можно прописать, например код, отвечающий за местоположение спрайта.

!Проблемная задача: Создать в классе спрайт метод – update(). Записать в него код, который изменяет координаты спрайта.

Решение:

```
class Sprite(pygame.sprite.Sprite):
    def __init__(self, x, y, speed, img):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load(img)
        self.rect = self.image.get_rect(topleft = (x, y))
        self.speed = speed
        self.speedx = 0
        self.speedy = 0
        self.jumppower = -10
    def update(self, keys, g, GrPlatf):
        if keys[pygame.K_RIGHT]:
            self.rect.x += self.speedx
            self.image = self.animGoR[self.cadr // 2 % 5]
            self.dir = True
        elif keys[pygame.K_LEFT]:
            self.rect.x -= self.speedx
            self.image = self.animGoL[self.cadr // 2 % 5]
            self.dir = False
        if keys[pygame.K_SPACE]:
            if self.speedy == 0:
                self.speedy = self.jumppower
        if not collideG(self, GrPlatf):
            self.speedy += g.value # gravitation
        elif self.speedy > 0:
```

```
self.speedy = 0  
self.rect.y += self.speedy
```

keys - это список нажатых клавиш в данный момент

g – это объект класса Const, константы g – ускорение свободного падения

GrPlatf – объект встроенного класса Group

collideG – функция, которая определяет пересекается ли спрайт с каким-либо спрайтом из группы.

! Проблемная задача: написать функцию collideG (Sprite, Group)

Подсказка: у класса Group есть метод – sprites(), который возвращает список, элементами которого являются сами спрайты из этой группы.

Решение:

```
def collideG(Sprite, Group):  
    Iscollide = False  
    for Spr in Group.sprites():  
        if collide(Sprite.rect, Spr.rect):  
            Iscollide = True  
    return Iscollide
```

Еще у спрайта Group есть метод – draw()

Он выполняет команду – blit() для всех свойств – image у спрайтов находящихся в группе.

То есть у каждого спрайта должно быть свойство с именем image.

!Проблемная задача: реализовать в программе метод draw у группы спрайтов платформ.

Решение:

Игровой цикл программы на данном этапе должен выглядеть так:

```
game = True

while game:
    clock.tick(24)

    for ev in pygame.event.get():
        if ev.type == pygame.QUIT:
            game = False

    keys = pygame.key.get_pressed()
    Player.update(keys, g, GroupPlatform)

    w.fill((0, 0, 0))

    GroupPlatform.draw(w)
    w.blit(Player.image, Player.rect)

    pygame.display.update()

    Ncadr += 1

pygame.quit ()
```

3. Рефлексия

- Сегодня мы познакомились со встроенными классами Sprite и Group
- Модифицировали программу