

# | PYGAME.

**ЦЕЛЬ:** создать стартовое меню в игре. Изучить технологию создания класса Button – кнопка.

**ПРИМЕЧАНИЕ:**

## ПЛАНИРОВАНИЕ

### 1. Меню

Что собой представляет меню? Упрощенно, это заставка перед игрой, где расположена кнопка – **Старт**, **Выход** и др.

А также, для работы меню, нам необходимо реализовать выполнение разных режимов в нашей программе:

- 1) Мы находимся в меню игры, отображается фон меню и кнопки
- 2) Мы находимся на первом уровне игры, отображаются спрайты игры, и, по сути, идет сама игра.
- 3) Мы находимся на 2 уровне игры (на него мы попадаем, если проходим первый)
- 4) И т.д.

Для начала реализуем режимы игры. Для этого можно создать глобальную переменную, значение которой будет обозначать номер уровня: 0 – стартовое меню игры, 1 – первый уровень, 2- второй уровень, и т.д. А в игровом цикле с помощью конструкции:

```
if  
elif  
elif  
.....  
elif
```

Реализуем выполнение разных режимов.

Переменную, отвечающую за номер уровня, назовем – **Stage**:

```

while game:

    clock.tick(24)

    for ev in pygame.event.get ():

        if ev.type == pygame.QUIT:

            game = False

    keys = pygame.key.get_pressed()

    # start menu

    if Stage == 0:

        .....

    # first stage

    elif Stage == 1:

        Player.update(keys, g, GroupPlatform)

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

        world.fill((0, 0, 0))

        world.blit(Player.image, Player.rect)

        GroupPlatform.draw(world)

        GroupFireBall.update(g)

        cam.update(Player)

        w.blit(world, (cam.x, cam.y))

        GroupFireBall.draw(w)

    pygame.display.update()

```

## 2. Class Button

В pygame нет встроенного класса, который бы реализовывал функции кнопки. Поэтому напомним его сами =)

Для начала давайте подумаем, что собой представляет кнопка, это изображение, которое отображается на объекте – поверхность, имеет определенные координаты, а также ширину и высоту. Как спрайт. Поэтому, будет удобно сделать этот класс наследником класса Sprite. Но у кнопки должно обрабатываться событие нажатия на нее. Поэтому создадим в классе Button метод – **onclick()**. Сделаем так, чтобы этот метод вызывался каждый раз как происходит нажатие на нашу кнопку – спрайт. Для этого можно поступить так: создать в классе метод – **update()**, который будет вызываться постоянно в игровом цикле, как и у других спрайтов, а в этом методе прописать код, который будет проверять, находится ли указатель мыши на спрайте-кнопке и не нажата ли в этот момент левая кнопка мыши, и при выполнении этих двух условий – вызывать метод **onclick()**.

А внутри метода **onclick()** мы пишем код, который должен выполняться при нажатии кнопки.

Схематически это выглядит так:

Класс Кнопка(Спрайт):

Метод `__init__()`:

**Спрайт.\_\_init\_\_()**

Метод `update()`:

Если указатель мыши над мной

И нажата левая кнопка:

`onclick()`

Метод `onclick()`:

**Уровень = 1**

пока Игра:

Если **Уровень == 0:**

**КнопкаСтарт.update**

**class Button(Sprite):**

**def \_\_init\_\_(self):**

**Sprite.\_\_init\_\_(self)**

**def update(self):**

**if (self.rect.x < pygame.mouse.get\_pos()[0]**

**< self.rect.x + self.rect.width**

**and self.rect.y < pygame.mouse.get\_pos()[1]**

**< self.rect.y + self.rect.height):**

**if pygame.mouse.get\_pressed()[0]:**

**self.onclick()**

**def onclick(self):**

**global Stage**

**Stage = 1**

While game:

if Stage == 0:

**buttonstart.update()**

**! Проблемная задача:**

Реализовать этот класс в своей игре. Создать меню с работающей кнопкой старт. Сделать кнопке эффект наведения – менять ей свойство – `image` при наведении курсора. (подсказка: создать в классе `Button` метод – `hover()`)

### 3. Рефлексия

- Сегодня мы сделали нашу игру многоуровневой
- Создали класс `Button`, который реализует собой настоящую кнопку