ДЕТСКАЯ ОНЛАЙН-ШКОЛА ПРОГРАММИРОВАНИЯ
HELLO WORLD

#ШПАРГАЛОЧКИ

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

Базовый уровень

Материалы подготовлены отделом методической разработки

Больше полезных материалов и общения в нашем комьюнити в Telegram: https://t.me/hw_school

Пинг-понг часть 1

Arcade - библиотека, с помощью которой мы можем создать окно для нашей игры, а также игровые объекты и персонажей - **спрайты**. Перед началом работы её нужно установить на компьютере, введя в консоль специальную команду:

pip install arcade

(для MacOS старше 2019 года **pip install arcade==2.5.6**, причем нужно использовать версию **Python 3.9.7**)

А потом импортировать в файле с кодом:

import arcade

arcade.Window - класс для создания окна. От него нужно унаследовать главный игровой класс:

```
class Game(arcade.Window):
```

Конструктор класса принимает 3 основных параметра - ширину, высоту и заголовок окна:

```
    def __init__(self, width, height, title):
        super().__init__(width, height, title)
```

on_draw - метод класса **arcade.Window**, позволяющий отрисовывать что-либо в окне. В первую очередь в нем нужно вызвать метод **clear**, очищающий окно. А если в **clear** передать цвет (например, в формате **RGB**), то так можно задать цвет фона:

```
def on_draw(self):
    self.clear((255, 0, 0))
```

Для версии arcade 2.5.6 код немного другой:

```
def on_draw(self):
    self.background_color = (255,0,0)
    self.clear()
```

RGB (Red Green Blue) - формат, в котором любой цвет можно указать как сочетание красного, зеленого и синего разной интенсивности (от 0 до 255). Например, **(255, 0, 0)** - красный цвет, а **(0, 255, 0)** - зелёный.

update - еще один метод класса **arcade.Window**, отвечает за обновление окна и игровую логику. Принимает параметр **delta_time** - время между сменой кадров:

def update(self, delta_time):

Чтобы программа заработала, нужно создать объект класса, унаследованного от **arcade.Window** и “запустить” **arcade**:

window = Game(600, 600, “Ping Pong”)

arcade.run()

arcade.Sprite - класс **arcade**, позволяющий создавать спрайты. При создании объекта (в **__init__** класса окна) нужно указать путь к картинке спрайта и его масштаб:

```
class Ball(arcade.Sprite):
    pass
class Game(arcade.Window):
    def __init__(self, width, height, title):
        …
        self.ball = Ball('ball.png', 0.8)
```

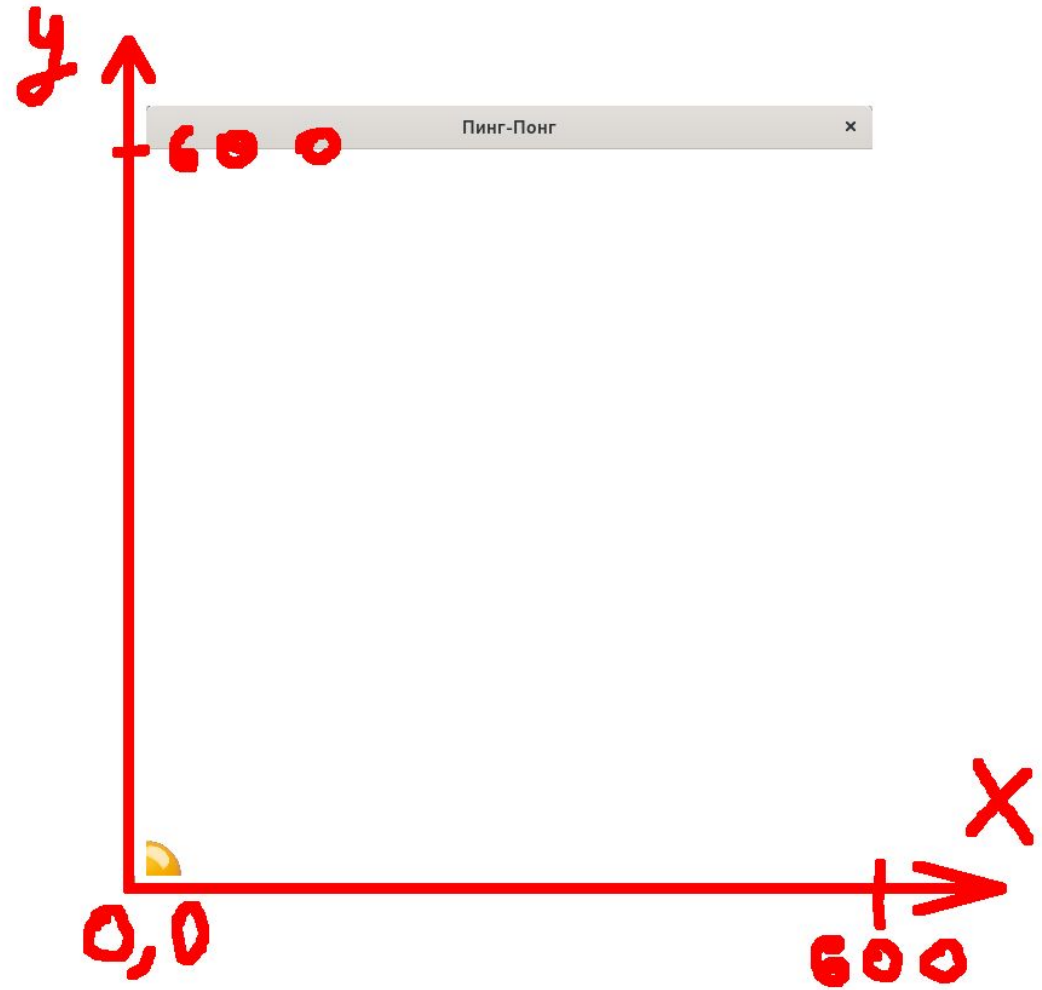
Чтобы увидеть созданный спрайт, его нужно отрисовать. Для этого в методе **on_draw** окна нужно вызвать метод **draw** спрайта:

```
def on_draw(self):
    ...
    self.ball.draw()
```

Управлять расположением спрайта на окне можно изменяя его свойства **center_x** (координаты центра спрайта по оси x) **center_y** (координаты центра спрайта по оси x). Важно! X увеличивается слева направо, а Y - снизу вверх:

...

```
self.ball = Ball('ball.png', 0.8)
self.ball.center_x = 300
self.ball.center_y = 300
```


y
600
Пинг-Понг
X
0,0
600

Чтобы спрайт двигался, нужно:

1.Задать значения его атрибутам **change_x** и **change_y** (скорость):

```
class Game(arcade.Window):

    def __init__(self, width, height, title):

        ...

        self.ball.change_x = 5

        self.ball.change_y = 3
```

2. Затем прописать у спрайта метод **update**, в котором его координаты будут изменяться на скорость:

```
class Ball(arcade.Sprite):

    def update(self):

        self.center_x += self.change_x
```

3. Вызвать обновление спрайта в **update** окна:

```
class Game(arcade.Window):

    def update(self, delta_time):

        self.ball.update()
```

Чтобы спрайт не уходил за границы окна, нужно менять его скорость на противоположную, когда он доходит до определенных координат. Это удобнее проверять через атрибуты **top**, **left**, **right** и **bottom** - координаты верхнего, левого, правого и нижнего краев спрайта:

```
def update(self):
    ...
    if self.top > 600 or self.bottom < 0:
        self.change_y = -self.change_y
```

Константы - это специальные переменные, которые нельзя изменять, значение задается им один раз (при создании). В **Python** нет констант как таковых, но существует соглашение между программистами: если имя переменной написано большими буквами, то это константа.


```
SCREEN_WIDTH = 600
SCREEN_HEIGHT = 600
```