

Introdução ao Pygame

Criando o Brick Breaker I

Python para Todos

CEFET-MG

Conteúdo

Criando a tela

Criando o jogador

Criando classes e o objeto jogador

Movimentos do jogador

Criando função para desenhar objetos

Criando a bolinha

Criando a tela

Iremos criar os primeiros elementos da tela.

- ▶ Altura e largura (800 pixels cada)
- ▶ Tela do nosso jogo
- ▶ Loop Principal

Criando a tela

```
# SETUP
import pygame as pg

pg.init()

largura = 800
altura  = 800
tela = pg.display.set_mode((largura, altura))
```

Criando a tela

```
# LOOP PRINCIPAL
while True:
    for event in pg.event.get():
        if event.type == pg.QUIT:
            pg.quit()
            exit()
    pg.display.update()
```

Conteúdo

Criando a tela

Criando o jogador

Criando classes e o objeto jogador

Movimentos do jogador

Criando função para desenhar objetos

Criando a bolinha

Criando o jogador

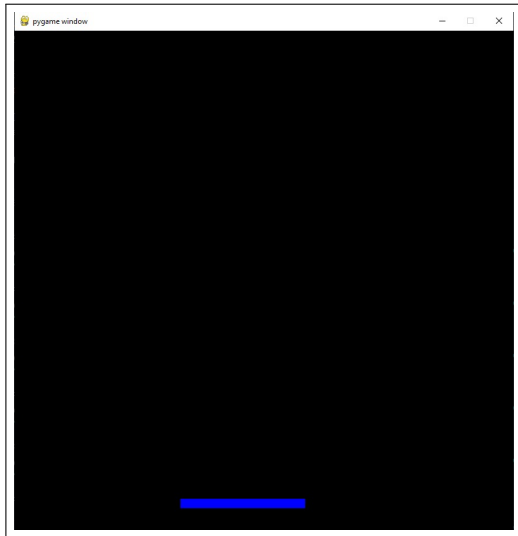
O jogador será uma barra azul que ficará na parte debaixo da tela

- ▶ Iremos criar uma reta para o jogador
- ▶ Após isso, iremos desenhar a barra azul na tela

Criando o jogador

```
# LOOP PRINCIPAL
#...
jogador = pg.Rect(
    tela.get_width()/3,
    largura-50, 200, 15
)
pg.draw.rect(tela, 'blue', jogador)
```


Criando o jogador



Conteúdo

Criando a tela

Criando o jogador

Criando classes e o objeto jogador

Movimentos do jogador

Criando função para desenhar objetos

Criando a bolinha

Criando classes e o objeto jogado

Uma classe é um conjunto de característica (variáveis) e ações (funções) que definem algum objeto. No nosso caso, iremos criar uma classe “Jogador”, que terá suas próprias funções e atributos dentro dele.

- ▶ Para isso, devemos transformar nosso **SETUP** e **LOOP PRINCIPAL** em uma função, que será a principal.
- ▶ Além disso, iremos criar uma classe para o jogador. O mesmo será feito para os próximos elementos criados no jogo.

Criando classes e o objeto jogado

```
def inicia_jogo():  
    # Coloque todo o código do jogo criado ATÉ ENTÃO aqui  
  
inicia_jogo()
```

Criando classes e o objeto jogado

Para criar uma classe, devemos

- ▶ Utilizar o comando `class Jogador`: que cria a classe “Jogador” e, dentro dela
 - ▶ `def __init__(self, ...)`: Isso é a função “Construtora” da classe, será ela que definirá os valores iniciais do nosso objeto.
- ▶ Toda vez que você quiser acessar uma variável dentro da classe, utiliza `self.variavel`. Caso seja fora da classe, utiliza `nome_objeto.variavel`

Criando classes e o objeto jogado

```
class Jogador:
    def __init__(self, cor, tamanho_barra, velocidade):
        self.cor = cor
        self.velocidade = velocidade
        self.tamanho_barra = tamanho_barra
        self.jogador = None

    def cria_jogador(self, altura):
        self.jogador = pg.Rect(350, altura-50,
                                self.tamanho_barra, 15)
```

Criando classes e o objeto jogado

```
# Na função principal
# SETUP
#...
jogador = Jogador('blue', 200, 13)
jogador.cria_jogador(altura)
```

Criando classes e o objeto jogado

```
# Na função principal  
# LOOP PRINCIPAL  
# ...  
    pg.draw.rect(tela, jogador.cor, jogador.jogador)
```


Conteúdo

Criando a tela

Criando o jogador

Criando classes e o objeto jogador

Movimentos do jogador

Criando função para desenhar objetos

Criando a bolinha

Movimentos do jogador

Agora, iremos colocar a movimentação do jogador

- ▶ Vamos criar uma função para movê-lo
- ▶ Além disso, iremos criar uma outra função dentro da classe jogador para alterar as coordenadas de movimentação dele.
 - ▶ Se a coordenada do jogador for negativa (ele está fora da tela pelo lado esquerdo), a coordenada x recebe 0
 - ▶ Se a coordenada do jogador for maior que a largura da tela (ele está fora da tela pelo lado direito), a coordenada x recebe a `largura`

Movimentos do jogador

```
def move_jogador(jogador, largura):  
    keys = pg.key.get_pressed()  
  
    if keys[pg.K_LEFT]:  
        jogador.move(-jogador.velocidade, largura)  
  
    if keys[pg.K_RIGHT]:  
        jogador.move(jogador.velocidade, largura)
```

Movimentos do jogador

```
# DENTRO DA CLASSE DO JOGADOR
```

```
def move(self, x, largura):  
    self.jogador.x += x
```

```
    if self.jogador.x < 0:  
        self.jogador.x = 0
```

```
    elif self.jogador.x > largura - self.tamanho_barra:  
        self.jogador.x = largura-self.tamanho_barra
```

Movimentos do jogador

```
# LOOP PRINCIPAL  
move_jogador(jogador, largura)  
tela.fill('black')
```

Conteúdo

Criando a tela

Criando o jogador

Criando classes e o objeto jogador

Movimentos do jogador

Criando função para desenhar objetos

Criando a bolinha

Criando função para desenhar objetos

Agora, iremos criar uma função para desenhar os objetos, pintar a tela e atualizar o jogo.

- ▶ Vamos pegar os códigos principais e transformá-lo em uma função
- ▶ Criaremos também uma variável para o clock

Criando função para desenhar objetos

```
def desenha_elementos(tela, jogador):  
    tela.fill('black')  
    pg.draw.rect(tela, jogador.cor, jogador.jogador)  
    pg.display.update()
```


Criando função para desenhar objetos

```
# SETUP  
clock = pg.time.Clock()
```

Criando função para desenhar objetos

```
# LOOP PRINCIPAL  
desenha_elementos(tela, jogador)  
move_jogador(jogador, largura)  
clock.tick(60)
```

Conteúdo

Criando a tela

Criando o jogador

Criando classes e o objeto jogador

Movimentos do jogador

Criando função para desenhar objetos

Criando a bolinha

Criando a bolinha

Por fim, iremos criar a classe da bolinha.

- ▶ O modelo da classe bolinha é semelhante a classe jogador, porém, ao invés de ser um quadrado, ela será um círculo
- ▶ Iremos criar a bolinha na função principal
- ▶ Alteraremos a função `desenha_elementos()`

Criando a bola

```
class Bola:  
  
    def __init__(self, tamanho_bola, cor):  
        self.tamanho_bola = tamanho_bola  
        self.cor = cor  
        self.bola = None  
        self.posicao = (0,0)
```

Criando a bola

```
def cria_bola(self, largura, altura):  
    self.posicao = (400, 400)  
    self.bola = pg.Rect(  
        self.posicao[0],  
        self.posicao[1],  
        self.tamanho_bola,  
        self.tamanho_bola  
    )
```

Criando a bola

```
# SETUP  
bola=Bola(15, 'white')  
bola.cria_bola(largura, altura)
```

Criando a bola

```
# LOOP PRINCIPAL  
    desenha_elementos(tela, jogador, bola)
```


Criando a bola

```
def desenha_elementos(tela, jogador, bola):  
    pg.draw.circle(tela, bola.cor, bola.bola.center,  
        bola.tamanho_bola)
```

Criando a bola

