CRIANDO SEU PRIMEIRO JOGO COM PYTHON E PYGAME

João Paulo Carvalho

PYTHON BRASIL 2020

QUEM SOU EU

PYTHON BRASIL 2020

Estudou Informática no IFPI Campus Parnaíba. Entusiasta do mundo Open-Source. Programador Python e JavaScript por amor. Trabalha na maior parte do tempo com desenvolvimento full stack de aplicações web, hoje usando principalmente **Python** no back-end. Apaixonado pela arte dos jogos.

@jjpaulo2

TUTORIAL ALPHA-0.0.1

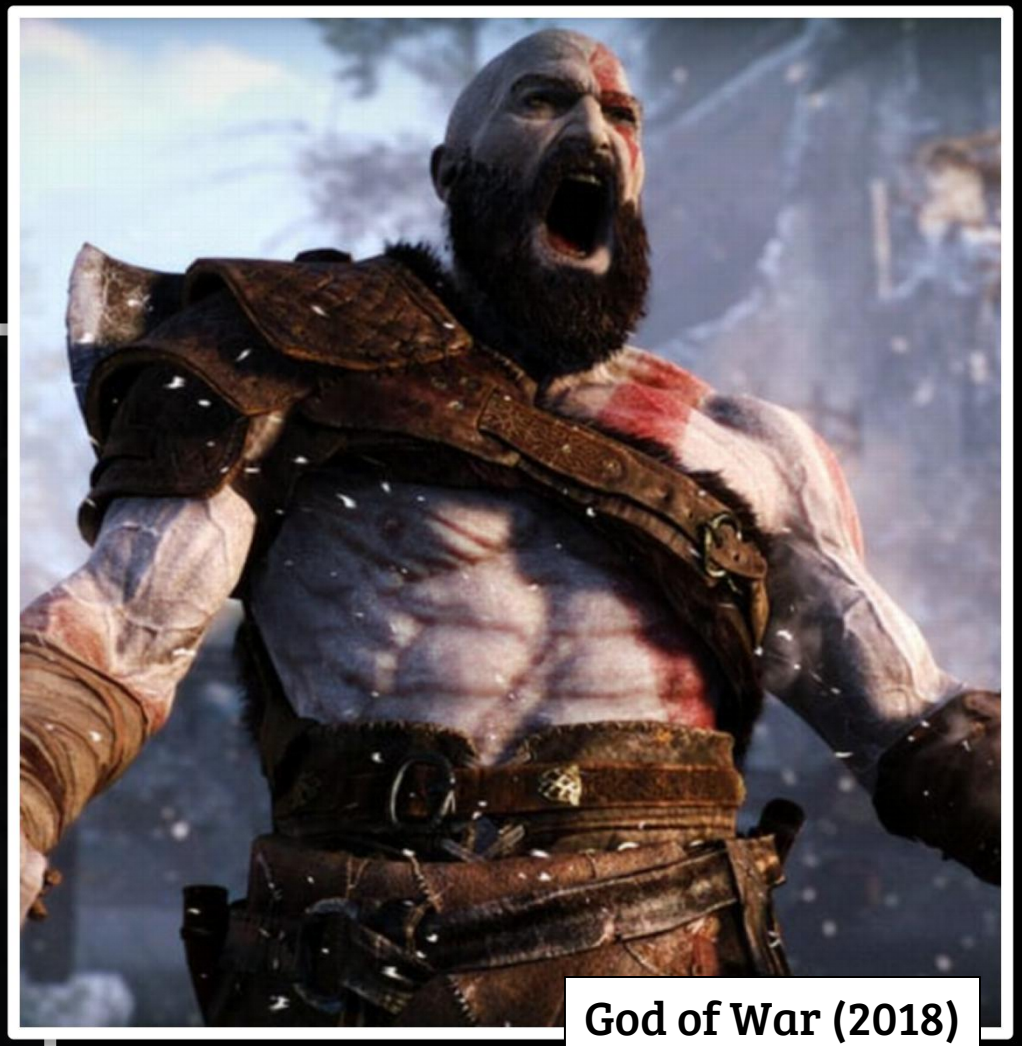
PYTHON
BRASIL
2020

Minicurso de PyGame ministrado na VII SEIFPI em 2018

MOTIVAÇÃO

PYTHON BRASIL 2020

The Last of Us Part II (2020)



God of War (2018)

POR QUE PYGAME?

PYTHON
BRASIL
2020

- Feito em Python.
- Simples de aprender.
- Gratuito, distribuido sob a LGPL.
- Processamento Multi-Core nativo.
- Compatível com qualquer sistema operacional.
- Construa seus jogos da forma que quiser.

PROJETOS DA COMUNIDADE

PYTHON BRASIL 2020

https://www.pygame.org

Barbie Seahorse Adventures, por **philhassey, trick, pekuja, tim, DrPetter**

Doom, por **StanislavPetrovV**

INSTALAÇÃO

PYTHON BRASIL 2020

Assumindo que você já preparou um ambiente virtual do Python, apenas execute o seguinte comando.

```
$ pip install pygame
```

Ou se preferir usar o **Pipenv**.

```
$ pipenv install pygame
```

TESTANDO A BIBLIOTECA

PYTHON BRASIL 2020

```
$ python -m pygame.examples.aliens
```

TESTANDO A BIBLIOTECA

PYTHON BRASIL 2020

```
$ python -m pygame.examples.chimp
```

INSTALAÇÃO (3D)

PYTHON BRASIL 2020

Assumindo que você está dentro do ambiente virtual do Python utilizado previamente, apenas execute o seguinte comando.

```
$ pip install pyopengl numpy
```

Ou se preferir usar o **Pipenv**.

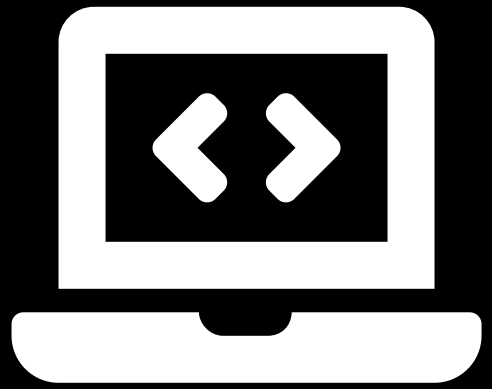
```
$ pipenv install pyopengl numpy
```

TESTANDO A BIBLIOTECA (3D)

PYTHON BRASIL 2020

```
$ python -m pygame.examples.glcube
```

PYTHON
BRASIL
2020



PRIMEIRO JOGO

CÓDIGO FUNCIONAL

PYTHON BRASIL 2020

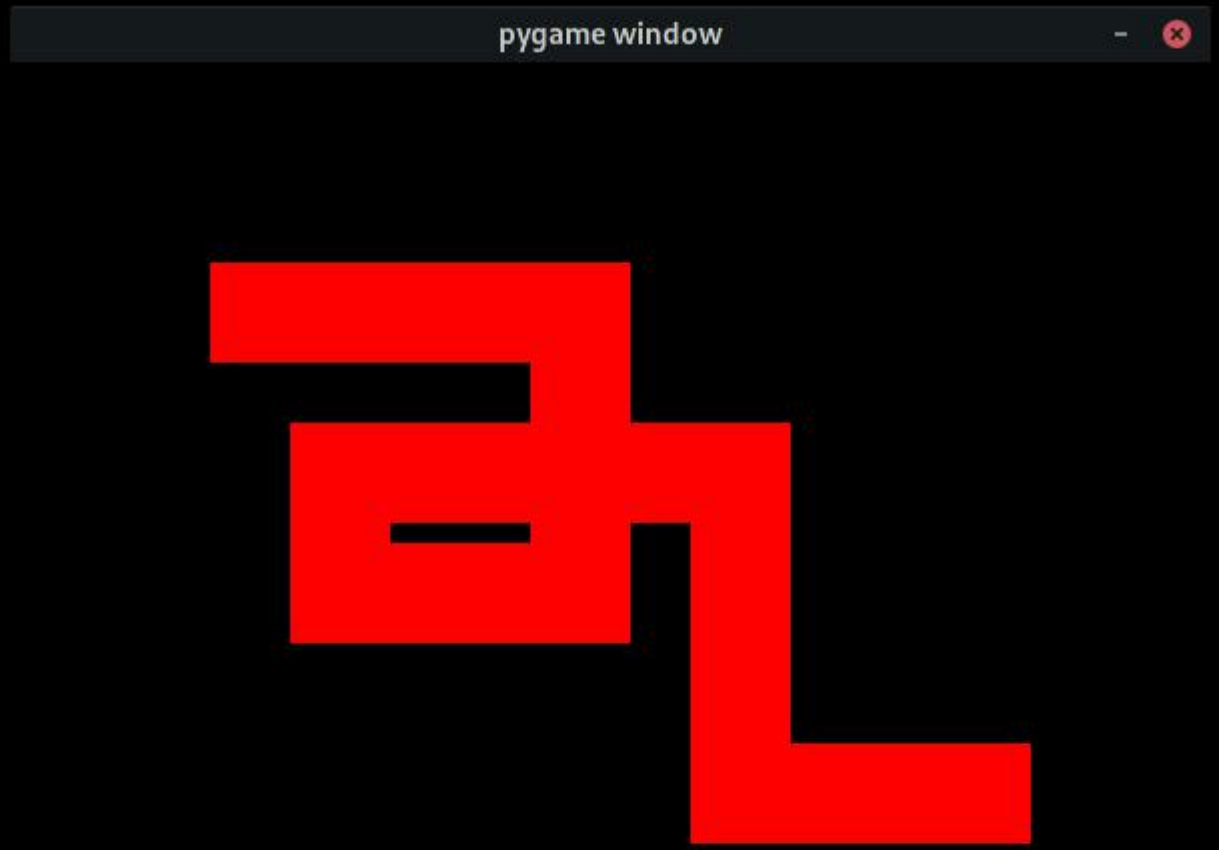
```python
# IMPORTANDO A BIBLIOTECA
import pygame

# INICIALIZANDO O PYGAME
pygame.init()

# CRIANDO NOSSA JANELA
tela = pygame.display.set_mode([600, 800])

# DEFININDO O RELÓGIO QUE VAI CONTAR OS FRAMES POR SEGUNDO
clock = pygame.time.Clock()

# CRIANDO UM QUADRADO QUE SERÁ EXIBIDO NA TELA
quadrado = pygame.Rect(100,  100,  50,  50)
```

```python
executando = True
while executando:

        # VERIFICANDO EVENTOS
        for evento in pygame.event.get():

                # EVENTO DE FECHAR A TELA
                if evento.type == pygame.QUIT:
                        executando = False

                # EVENTOS DE TECLA PRESSIONADA
                if evento.type == pygame.KEYDOWN:
                        if evento.key == pygame.K_DOWN:
                                quadrado.move_ip([0, 20])
                        if evento.key == pygame.K_UP:
                                quadrado.move_ip([0, -20])
                        if evento.key == pygame.K_LEFT:
                                quadrado.move_ip([-20, 0])
                        if evento.key == pygame.K_RIGHT:
                                quadrado.move_ip([20, 0])

        # ELEMENTOS DA TELA
        pygame.draw.rect(tela, (255, 0, 0), quadrado)

        # CONFIGURAÇÃO DE QUADROS
        clock.tick(27)
        pygame.display.update()

pygame.quit()
```

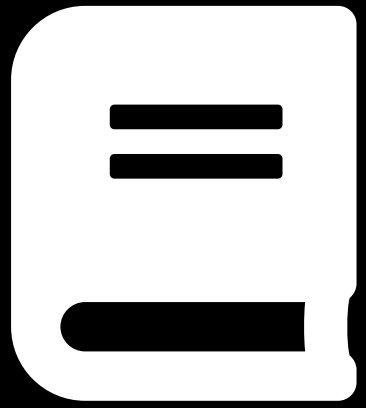
PYTHON BRASIL 2020

RESULTADO

PYTHON
BRASIL
2020



PYTHON
BRASIL
2020



COMPREENDENDO CONCEITOS

JANELAS

PYTHON BRASIL 2020

Onde todo o conteúdo que você vai criar será exibido.

```
# DEFININDO DIMENSÃO DA TELA
# É IMPORTANTE GUARDAR A INSTÂNCIA DA TELA PARA PODER FAZER MANIPULAÇÕES
tela = pygame.display.set_mode([300, 300])

# DEFININDO O TÍTULO DA JANELA
pygame.display.set_caption("Título da minha janela")
```

LOOP PRINCIPAL

PYTHON BRASIL 2020

O loop que acontece durante a execução do jogo. É onde toda a lógica irá ser moldada.

```
1  # DEFININDO UMA FLAG PRA VERIFICAR A EXECUÇÃO
2  executando = True
3  while executando:
4
5          # OBTENDO OS EVENTOS E ITERANDO SOBRE ELES
6          for evento in pygame.event.get():
7
8                  # VERIFICANDO SE HOUVE SOLICITAÇÃO PRA FECHAR A JANELA
9                  if evento.type == pygame.QUIT:
10                         executando = False
11
12 # INDICANDO O FIM DO SCRIPT DO PYGAME
13 pygame.quit()
```

QUADROS / CLOCK

PYTHON BRASIL 2020

O relógio que marca e define a quantidade de loops que acontecerão em 1 segundo.

```
# CRIANDO O OBJETO CLOCK
clock = pygame.time.Clock()
...

while executando:
        ...

        # MANDA O RELÓGIO RODAR PARA SER CAPAZ DE
        # EXIBIR 30 QUADROS POR SEGUNDO
        clock.tick(30)

        # MÉTODO QUE ATUALIZA A TELA
        pygame.display.update()
```

CORES

PYTHON BRASIL 2020

Cores no PyGame são representadas por tuplas do tipo `(red: int, green: int, blue: int)`.

```
1  VERMELHO = (255, 0, 0)
2  AZUL = (0, 0, 255)
3  VERDE = (0, 255, 0)
4  BRANCO = (255, 255, 255)
5  PRETO = (0, 0, 0)
6  CINZA = (150, 150, 150)
```

EVENTOS

PYTHON BRASIL 2020

Ações que podem ser tratadas durante o loop principal do seu jogo. Elas podem ser de vários tipos e possuem parâmetros que podem ser lidos para tratá-los.

```
while executando:
        ...

        # EVENTO DE SAIR DO JOGO
        if evento.type == pygame.QUIT:
                ...

        # EVENTOS DE TECLA PRESSIONADA
        if evento.type == pygame.KEYDOWN:
                if evento.key == pygame.K_DOWN:
                        ...
```

EVENTOS (TIPOS)

PYTHON BRASIL 2020

TIPO DE EVENTO	PARÂMETROS
QUIT	none
ACTIVEEVENT	gain, state
KEYDOWN	key, mod, unicode, scancode
KEYUP	key, mod
MOUSEMOTION	pos, rel, buttons
MOUSEBUTTONUP	pos, button
MOUSEBUTTONDOWN	pos, button
VIDEORESIZE	size, w, h
USEREVENT	code

RETÂNGULOS

PYTHON BRASIL 2020

Retângulos no PyGame são declarados por objetos da forma `pygame.Rect(left: float, top: float, width: float, height: float)`.

```
1  # DECLARANDO RETÂNGULO
2  retangulo = pygame.Rect(10, 10, 30, 30)
3  ...
4
5  while executando:
6          ...
7
8          # MÉTODO QUE DESENHA O RETÂNGULO NA TELA
9          pygame.draw.rect(tela, cor, retangulo)
```

COLISÕES

PYTHON BRASIL 2020

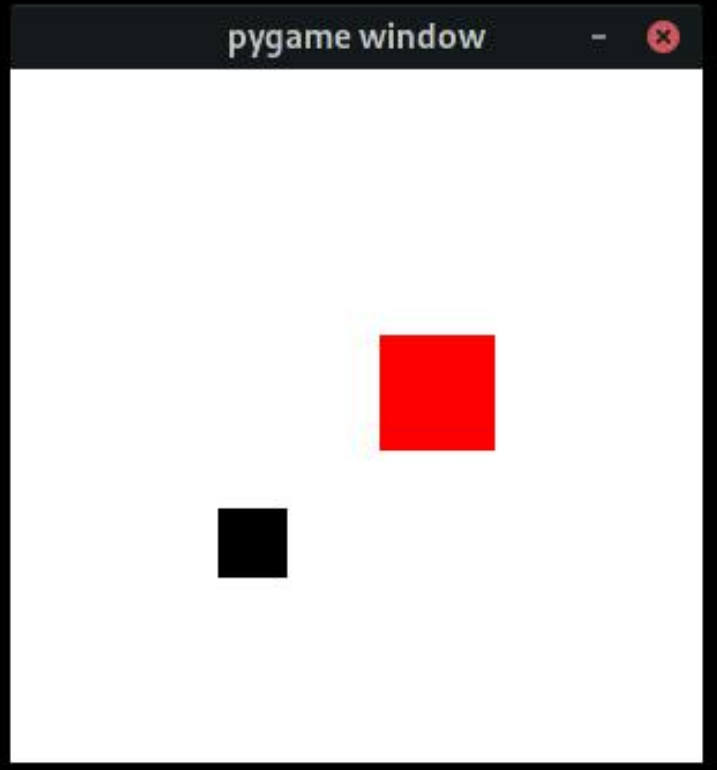
A classe Rect possui métodos capazes de verificar colisões.

```
1  # VERIFICA SE O RETÂNGULO CONTÉM O OUTRO PASSADO COMO PARÂMETRO
2  pygame.Rect.contains(Rect) -> bool
3
4  # VERIFICA SE O RETÂNGULO CONTÉM O PONTO PASSADO COMO PARÂMETRO
5  pygame.Rect.collidepoint(x, y) -> bool
6
7  # VERIFICA SE O RETÂNGULO SE COLIDIU COM O OUTRO
8  pygame.Rect.colliderect(Rect) -> bool
9
10 # VERIFICA SE ALGUM RETÂNGULO DA LISTA COLIDIU COM O OBJETO
11 pygame.Rect.collidelist(list[Rect]) -> bool
12
13 # VERIFICA SE TODOS OS RETÂNGULO DA LISTA COLIDIRAM COM O OBJETO
14 pygame.Rect.collidelistall(list[Rect]) -> bool
```

PYTHON
BRASIL
2020

DESAFIO 1

Construa um jogo com **dois retângulos** na tela. Você deverá controlar um deles e assim que ele colidir com o segundo, este deverá **mudar sua posição** na tela.



SUPERFÍCIES

PYTHON BRASIL 2020

Elas nos permitem ir além de apenas retângulos coloridos e nos dá o poder de começar a manipular imagens.

```
# CRIANDO A SUPERFÍCIE DE DIMENSÕES (largura, altura)
superficie = pygame.Surface((200, 200))
...

while executando:
        ...

        # MÉTODO QUE DESENHA A SUPERFÍCIE NA TELA
        # NA POSIÇÃO (x, y)
        tela.blit(superficie, (0, 0))
```

SUPERFÍCIES

PYTHON BRASIL 2020

Para usar imagens, basta usar o seguinte método.

```
# A VARIÁVEL “imagem” É UM OBJETO DA CLASSE pygame.Surface
imagem = pygame.image.load(r“C:/minha_imagem.png”)

# MÉTODO QUE RETORNA O RETÂNGULO DA SUPERFÍCIE
imagem_rect = imagem.get_rect()
```

PYTHON
BRASIL
2020

DESAFIO 2

Construa um jogo com uma **imagem de cenário** e uma **imagem de personagem** que consegue se mover.

MÚSICA

PYTHON BRASIL 2020

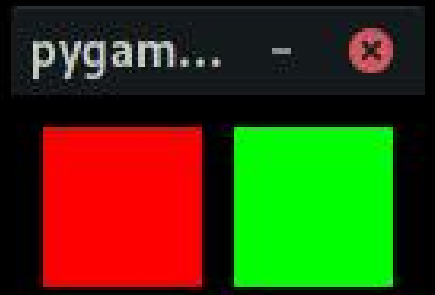
Podemos tocar músicas no nosso jogo com a ajuda do pacote `pygame.mixer`.

```
1   # INICIALIZE O MIXER ANTES DE INICIALIZAR O PYGAME
2   # OS PARÂMETROS SÃO (frequencia, tamanho, canal, buffer)
3   pygame.mixer.pre_init(44100, 16, 2, 1024)
4   pygame.init()
5   ...
6
7   while executando:
8           ...
9
10          # CARREGANDO E TOCANDO A MÚSICA
11          pygame.mixer.music.load(r"C:\minha_musica.wav")
12          pygame.mixer.music.set_volume(0.5)
13          pygame.mixer.music.play()
```

PYTHON
BRASIL
2020

DESAFIO 3

Construa um jogo com **dois retângulos** na tela. Cada um deve tocar uma música específica ao ser clicado.



SPRITES

PYTHON BRASIL 2020

São sequências de imagens que geram uma animação. Iremos implementa-los como classes filhas de `pygame.sprite.Sprite`.

```
1   class MeuSprite(pygame.sprite.Sprite):
2
3           def __init__(self, x, y):
4                   ...
5
6                   self.x = x
7                   self.y = y
8                   self.images = [...]
9                   self.index = ...
10                  self.image = ...
11                  self.rect = ...
12
13          def update(self):
14                  ...
```

ONDE CONSEGUIR SPRITES?

PYTHON BRASIL 2020

https://www.spriters-resource.com

FOLHA DE SPRITE

PYTHON
BRASIL
2020

PYTHON
BRASIL
2020

DESAFIO 4

Adicione um pouco de movimento ao **desafio 2**. Implemente uma classe de sprites e dê vida ao seu personagem.

PYTHON
BRASIL
2020

PARABÉNS POR TER CHEGADO ATÉ AQUI!

LEITURA COMPLEMENTAR

PYTHON BRASIL 2020

Introdução ao Desenvolvimento de Jogos em Python com PyGame, de Harrison Kinsley e Will McGugan.

OBRIGADO PELA ATENÇÃO!

@jjpaulo2

João Paulo Carvalho

PYTHON BRASIL 2020