Dando os primeiros passos

PyGame

Importando e iniciando o PyGame

Para trabalharmos com o PyGame primeiro precisamos importar a biblioteca pygame e realizar sua inicialização com o método init.

Veja o código a seguir:

```
import pygame

pygame.init()
```

PyGame

Desenhando a tela do jogo

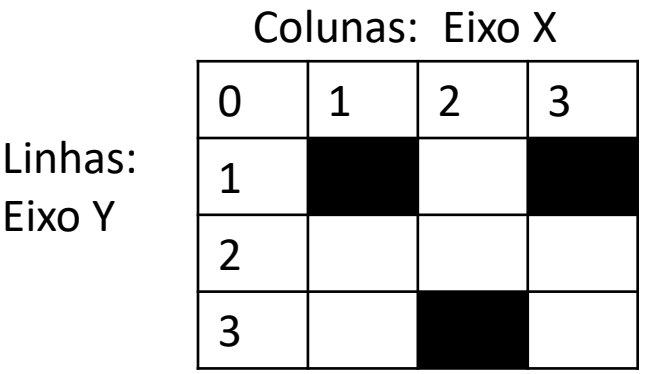
O PyGame possui um módulo para controlar a janela de exibição e telas. Este módulo é o ***pygame.display*** .

Para criar uma tela para exibir, chamamos o método ***set_mode()*** do módulo ***pygame.display*** e fornecemos uma tupla com a largura e altura da janela.

```
import pygame

pygame.init()

screen = pygame.display.set_mode((400, 300))
```



Neste exemplo estamos desenhando uma janela de 400 x 300 pixels.

PyGame

Criando o loop do jogo e capturando eventos

A tela principal do jogo terá um loop onde toda a ação acontece. Este loop é executado continuamente durante o jogo, atualizando o estado do jogo, renderizando a tela e capturando informações.

Para todo jogo é necessário uma maneira para sair do loop, e do aplicativo em si.

Todos os eventos e entradas do usuário vão para a fila de eventos do PyGame e podemos acessar essa fila de eventos chamando ***pygame.event.get() .***

Este método retorna uma lista de todos os eventos na fila, onde podemos percorrer e responder de acordo com o tipo de evento.

PyGame

Criando o loop do jogo e capturando eventos

```
import pygame

# Importação realizada para acessar as constantes de teclado
from pygame.locals import *

# Iniciando o pygame
pygame.init()

# Desenhando a tela
screen = pygame.display.set_mode((400, 300))

# Variável para manter nosso loop principal executando
running = True
...
```

PyGame

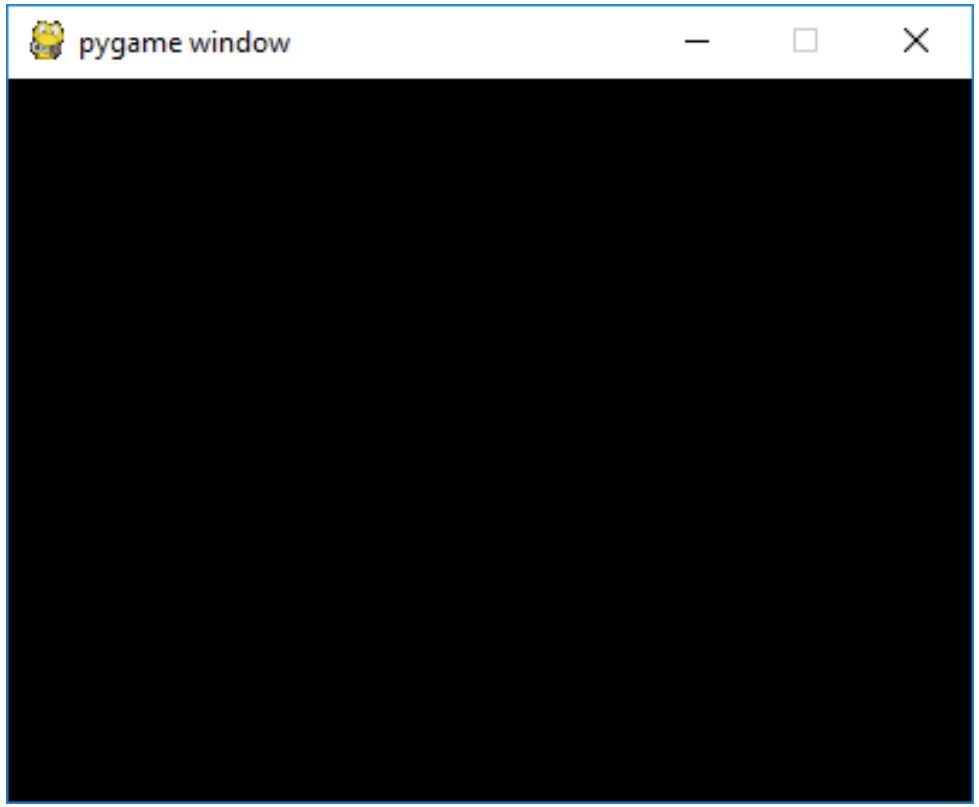
Criando o loop do jogo e capturando eventos

```
...
# Loop principal
while running:
    # Percorrendo a fila de eventos
    for event in pygame.event.get():
        # Verificando o evento KEYDOWN, q é uma constante definida em pygame.locals
        if event.type == KEYDOWN:
            # Se a tecla Esc for pressionada, setamos a variável
            # running para False para sair do loop principal
            if event.key == K_ESCAPE:
                running = False
        # Verificando se o evento foi o QUIT e setamos running para False
        elif event.type == QUIT:
            running = False
```

PyGame

Criando o loop do jogo e capturando eventos

Ao executar o exemplo anterior será exibida uma tela como a exibida abaixo:



Para fechar a tela pressione a tecla ESC.

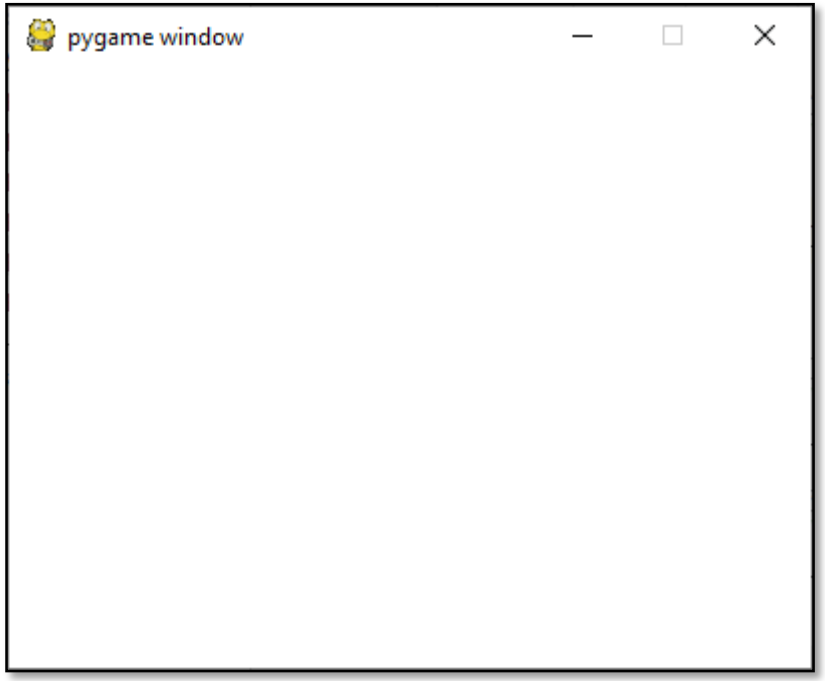
PyGame

Mudando a cor da tela

Para mudar a cor da tela, podemos utilizar o método ***fill*** do display. Este método preenche toda a superfície com a cor informada.

Adicione as linhas a seguir dentro do *While* do nosso último exemplo e veja que sua tela ficará igual a imagem ao lado.

```
…
while running:
    # Preenche a superfície com branco
    screen.fill((255, 255, 255))
    # Atualiza o conteúdo de toda a tela
    pygame.display.flip()
    # Percorrendo a fila de eventos
    for event in pygame.event.get():
…
```



O método ***flip*** do display atualiza o conteúdo da tela.

PyGame

Mudando a cor da tela

O método ***fill*** recebe uma tupla com o padrão de cores RGB (Red, green e blue). Estes valores vão de zero a 255.

Veja a definição de algumas cores no padrão RGB:

```
Preto = (0, 0, 0)
Branco = (255, 255, 255)
Azul = (0, 0, 255)
Verde = (0, 255, 0)
Vermelho = (255, 0, 0)
```

Mude os valores informados no método fill conforme cores acima (ou outro valor desejado), execute o programa e veja o resultado.

FIM