

LINGUAGEM DE PROGRAMAÇÃO I

AULA 01
CONCEITOS INTRODUTÓRIOS

LUIS GUSTAVO ARAUJO



01

INTRODUÇÃO AO
PPLAY

PPLAY?

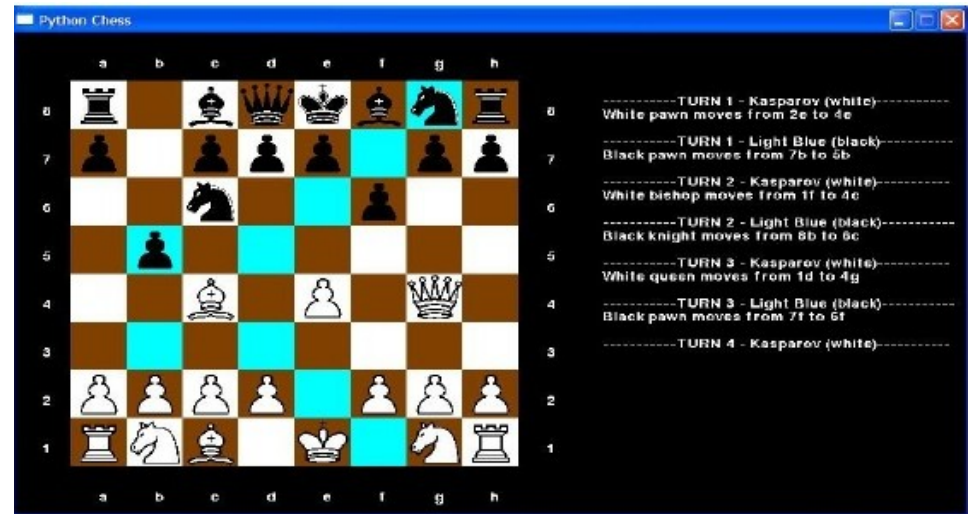
PPlay é uma biblioteca em Python que abstrai (facilita) os comandos utilizados no Pygame (outra biblioteca) para a criação de jogos.

DEPENDÊNCIAS:

Para usarmos o PPlay é preciso instalar o Python e também o Pygame.

01

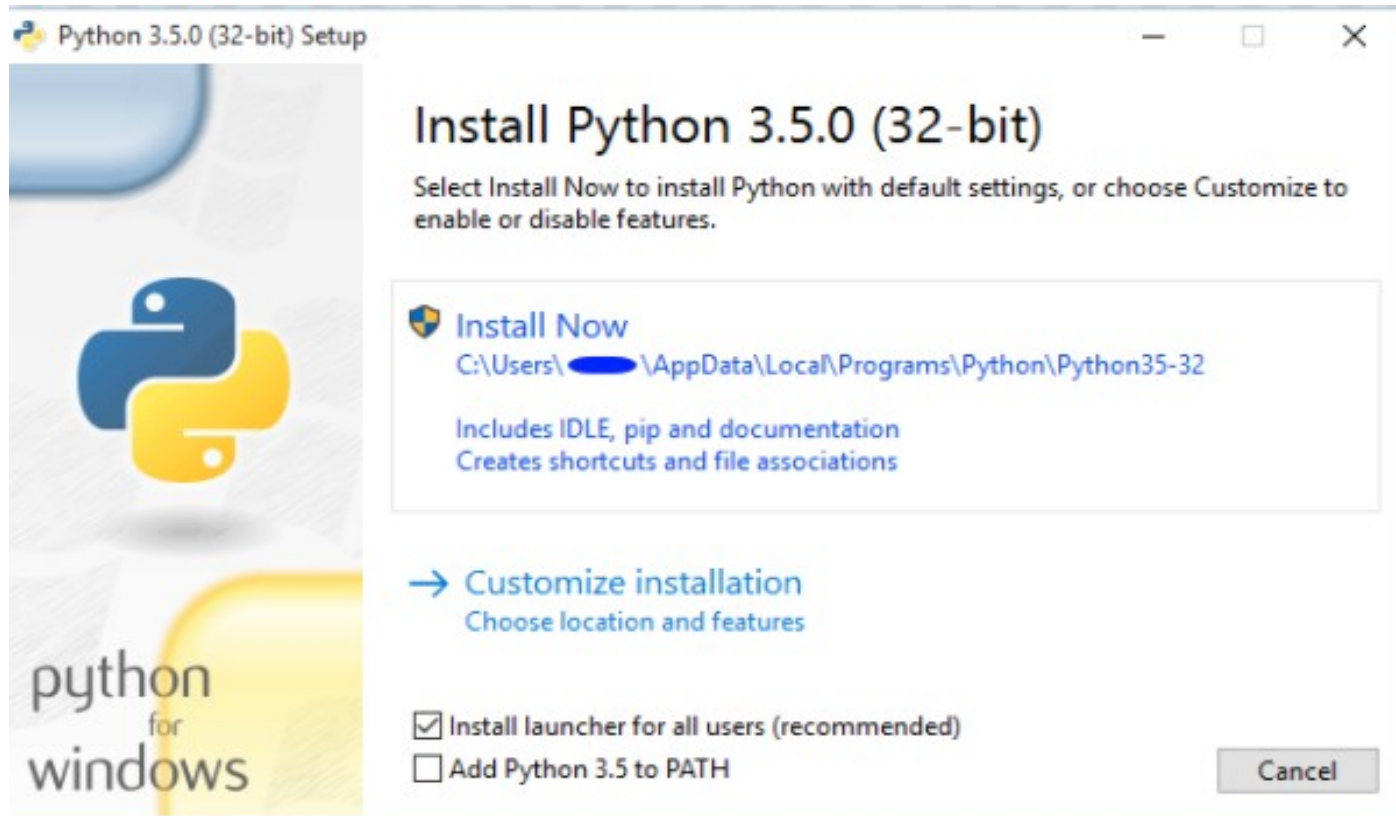
INTRODUÇÃO AO PPLAY

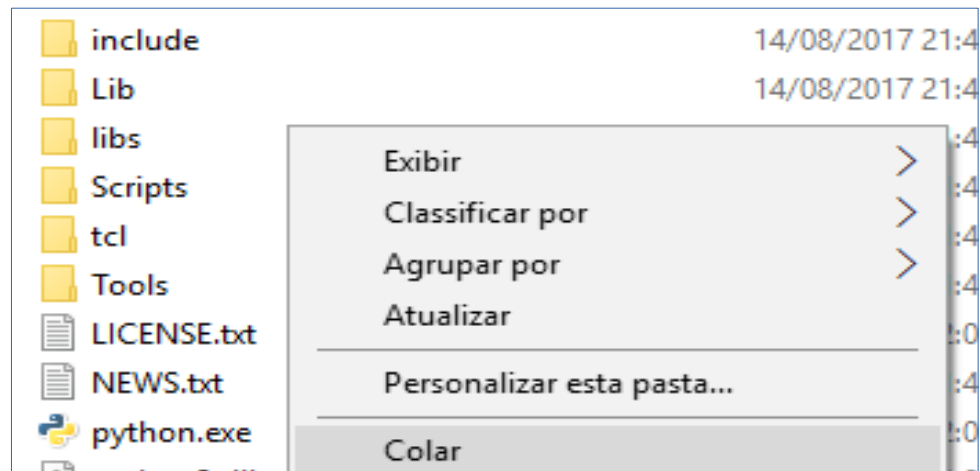
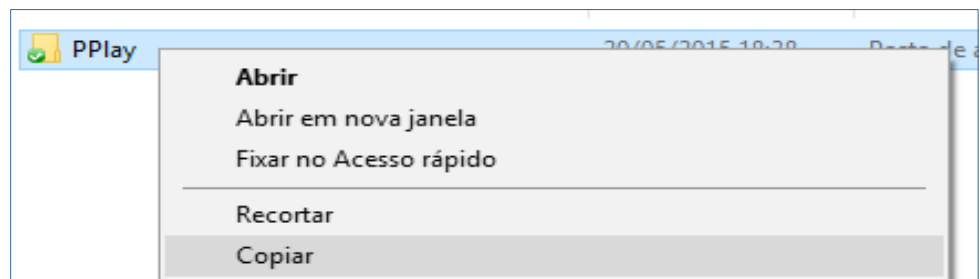
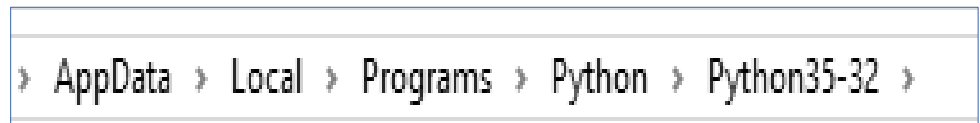
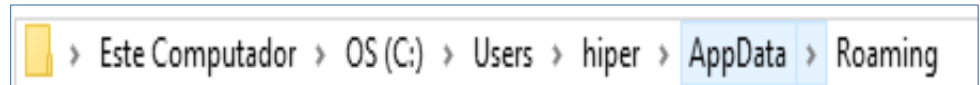
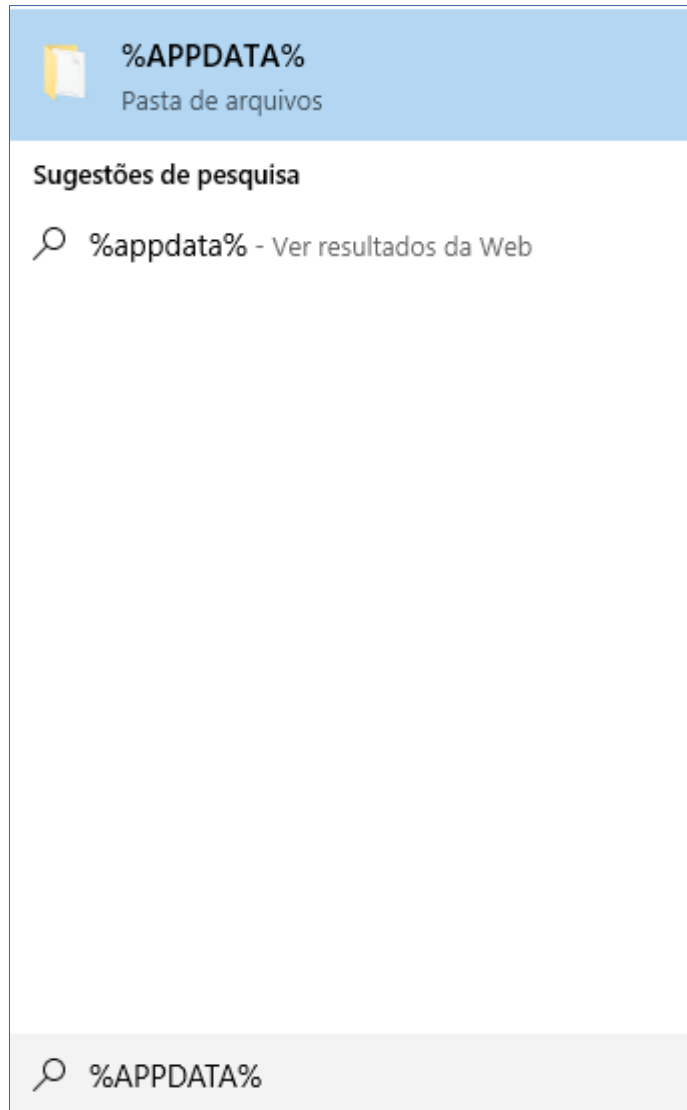




02

INSTALANDO O
PPLAY





03

O CLÁSSICO
HELLO WORLD

teste.py - C:\Users\hiper\Desktop\teste.py (3.5.2)

File Edit Format Run Options Window Help

```
#importando o módulo WINDOW
from PPlay.window import *

# Como sempre, criamos primeiro a janela (Largura, Altura)
janela = Window(400,300)
#Titulo da Janela
janela.set_title("Jogo")
```

04

INSERINDO
IMAGENS

 *teste.py - C:\Users\hiper\Desktop\teste.py (3.5.2)*

File Edit Format Run Options Window Help

```
from PPlay.window import *
#importando o módulo SPRITE
from PPlay.sprite import *

janela = Window(400,300)
janela.set_title("Jogo")
janela.set_background_color((0,0,0))

#Criando uma Image
ball = Sprite("quadrado.png")
#Desenhando a Image
ball.draw()
#Atualizando a tela
janela.update()
```

 *teste.py - C:\Users\hiper\Desktop\teste.py (3.5.2)*

File Edit Format Run Options Window Help

```
from PPlay.window import *
from PPlay.sprite import *

janela = Window(400,300)
janela.set_title("Jogo")
janela.set_background_color((0,0,0))

ball = Sprite("quadrado.png")
#Configurando a posição da Imagem
ball.set_position(200,150)

ball.draw()
janela.update()
```



03

LOOPS
GAME

GAME LOOP

É um *loop* especial utilizado em TODOS os jogos. Sem ele, um jogo não poderia ter animações, ações, movimentos, interação como teclado, mouse e etc.

USAMOS UM FOR?

Sabemos quando um jogo começa, mas nunca temos certeza quando ele termina, **assim não podemos usar o for, pois ele espera uma quantidade definida de repetições. Lembra?**

LOOP WHILE

While é também uma estrutura de repetição, ou seja, um loop. While significa “Enquanto”, nele NÃO precisamos definir a quantidade de repetições, basta colocarmos uma CONDIÇÃO, como os IF (seleção), para que ENQUANTO seja verdade execute.

While (**CONDIÇÃO**):

```
from PPlay.window import *
from PPlay.sprite import *

janela = Window(400,300)
janela.set_title("Jogo")
janela.set_background_color((0,0,0))

ball = Sprite("quadrado.png")
ball.set_position(200,150)

#Loop sempre!
while(True):
    janela.set_background_color((0,0,0))
    #movendo a Imagem na eixo x
    ball.move_x(0.1)
    ball.draw()
    janela.update()
```



```
#Loop sempre!  
while(True):  
    janela.set_background_color((0,0,0))  
    ball.move_x(0.1)  
    #movendo a Imagem na eixo y  
    ball.move_y(0.1)  
    ball.draw()  
    janela.update()
```

05

USANDO CONDICIONAIS PARA O TECLADO

CONDICIONAIS

Já vimos condicionais, também conhecidos com estruturas de seleção. Elas são: **if**, **else** e **elif**. Essas condicionais retornam *True* ou *False* para qualquer sentença. Aqui vamos usar para saber se a tecla foi pressionada (*True*) ou não (*False*).

```
if ( teclado.key_pressed("A" )):
```

```
#Loop sempre!  
while(True):  
    janela.set_background_color((0,0,0))  
    #SE o tecla for pressionada  
    if(teclado.key_pressed("RIGHT")):  
        ball.move_x(0.1)  
  
    ball.draw()  
    janela.update()
```

PODEMOS USAR VÁRIAS TECLAS:

`if(teclado.key_pressed("RIGHT")): (DIREITA)`

`if(teclado.key_pressed("LEFT")): (ESQUERDA)`

`if(teclado.key_pressed("UP")): (CIMA)`

`if(teclado.key_pressed("DOWN")): (BAIXO)`

`if(teclado.key_pressed("A")): (TECLA A)`

`if(teclado.key_pressed("B")): (TECLA B)`

06

USANDO CONDICIONAIS
PARA O COLISÃO

CONDICIONAIS

Aqui vamos usar para saber se um objeto colidiu com outro (*True*) ou não (*False*).

```
if ( ball.collided( ball2) ):
```

```
#Outra Imagem
ball2 = Sprite("quadrado.png")
ball2.set_position(0,150)
#Flag
colidiu = False;

#Loop sempre!
while(True):
    janela.set_background_color((0,0,0))
    #Se a Image 1 colidier com a outra
    if(ball.collided(ball2)):
        colidiu = True
    #Se não colidiu ande no eixo x
    if(colidiu == False):
        ball.move_x(-0.1)
        ball2.move_x(0.1)
    #se colidiu ande no eixo y
    else:
        ball.move_y(-0.1)
        ball2.move_y(0.1)

    ball.draw()
    ball2.draw()
```


VARIÁVEIS

Variáveis servem para GUARDAR informações. No código anterior usamos uma variável para saber se movemos no eixo x ou no eixo y, esse tipo de uso é chamado de flag (bandeira), uma espécie de SINALIZADOR.

colidiu = True

07

CRIANDO O NOSSO
PONG

LINGUAGEM DE PROGRAMAÇÃO I

AULA 01
CONCEITOS INTRODUTÓRIOS

LUIS GUSTAVO ARAUJO