



Interface Gráfica

fmasanori@gmail.com

Interface Gráfica

- Não basta funcionar, deve ser bonito também
- Interface texto funciona, mas é pobre
- GUI – Graphical User Interface
- Show de Calouros

Show de Calouros

- Opção 1 para resposta correta e 2 para errada
- Teremos um efeito sonoro para cada resposta
- O programa deve contar quantas respostas certas e erradas houve
- Finalmente apertamos 0 para finalizar

Show de Calouros

Atenção: instalar Pygame e
baixar os arquivos de áudio
para continuar a lição



Show de Calouros

Cria um objeto mixer
Inicializa o som

```
import pygame.mixer  
[ sounds = pygame.mixer  
  sounds.init()
```

Espera até get_busy
retornar False

```
def espera_tocar (canal):  
[   while canal.get_busy():  
      pass
```



Ocupado tocando?
"pass" não faz nada

```
s = sounds.Sound("heartbeat.wav")  
espera_tocar(s.play())  
s2 = sounds.Sound("buzz.wav")  
espera_tocar(s2.play())  
s3 = sounds.Sound("ohno.wav")  
espera_tocar(s3.play())  
s4 = sounds.Sound("carhorn.wav")  
espera_tocar(s4.play())
```

Show de Calouros



Agora que o áudio está
funcionando vamos fazer o
programa do Show de Calouros!



Show de Calouros

```
import pygame.mixer
sounds = pygame.mixer
sounds.init()

def espera_tocar(canal):
    while canal.get_busy():
        pass

certos = 0
errados = 0
opção = int(input ("Aperte 1)Certo 2)Errado e 0)Finalizar "))

while opção != 0:
    if opção == 1:
        s = sounds.Sound("correct.wav")
        espera_tocar(s.play())
        certos = certos + 1
    if opção == 2:
        s2 = sounds.Sound("wrong.wav")
        espera_tocar(s2.play())
        errados = errados + 1
    opção = int(input ("Aperte 1)Certo 2)Errado e 0)Finalizar "))

print ("Certos:", certos, "Errados:", errados)
```

Show de Calouros

>>>

Aperte 1) Certo 2) Errado e 0) Finalizar 1
Aperte 1) Certo 2) Errado e 0) Finalizar 1
Aperte 1) Certo 2) Errado e 0) Finalizar 1
Aperte 1) Certo 2) Errado e 0) Finalizar 1
Aperte 1) Certo 2) Errado e 0) Finalizar 2
Aperte 1) Certo 2) Errado e 0) Finalizar 1
Aperte 1) Certo 2) Errado e 0) Finalizar 2
Aperte 1) Certo 2) Errado e 0) Finalizar 1
Aperte 1) Certo 2) Errado e 0) Finalizar 0
Certos: 6 Errados: 2

O som ficou
legal... Mal posso
esperar para o
Show!



Show de Calouros

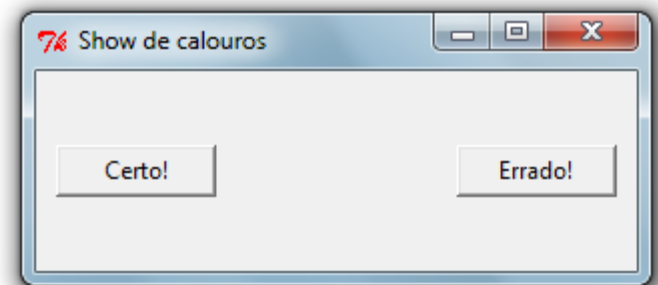


Agora vamos melhorar a usabilidade usando uma interface gráfica

Janela básica

```
from tkinter import *  
app = Tk()  
app.title("Show de calouros")  
app.geometry('300x100+200+100')  
  
b1 = Button(app, text = "Certo!", width = 10)  
b1.pack(side = "left", padx = 10, pady = 10)  
  
b2 = Button(app, text = "Errado!", width = 10)  
b2.pack (side = "right", padx = 10, pady = 10)  
  
app.mainloop()
```

Agora a extensão do seu programa é “.pyw”



Show Calouros GUI 01.pyw

Capturando um evento

```
from tkinter import *
```

```
def apertei_botão():  
    print ('Apertei o botão!')
```

```
app = Tk()  
app.title("Teste de botão")  
app.geometry('300x100+200+100')
```

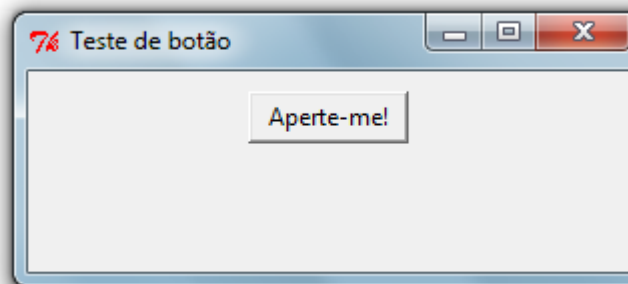
```
b = Button(app, text = "Aperte-me!", width = 10, command = apertei_botão)  
b.pack(side = "top", padx = 10, pady = 10)
```

```
app.mainloop()
```

} Função que irá rodar quando o evento ocorrer

```
>>>
```

```
Apertei o botão!  
Apertei o botão!  
Apertei o botão!
```



Show de Calouros

```
from tkinter import *
app = Tk()
app.title("Show de calouros")
app.geometry('300x100+200+100')
import pygame.mixer
sounds = pygame.mixer
sounds.init()
certos = 0
errados = 0
def espera_tocar(canal):
    while canal.get_busy():
        pass
def musica_certa():
    global certos
    s = sounds.Sound("correct.wav")
    espera_tocar(s.play())
    certos = certos + 1
def musica_errada():
    global errados
    s = sounds.Sound("wrong.wav")
    espera_tocar(s.play())
    errados = errados + 1

b1 = Button(app, text = "Certo!", width = 10, command = musica_certa)
b1.pack(side = "left", padx = 10, pady = 10)
b2 = Button(app, text = "Errado!", width = 10, command = musica_errada)
b2.pack (side = "right", padx = 10, pady = 10)

app.mainloop()
```

Show Calouros GUI 03.pyw

Show de Calouros

Estão faltando
os resultados...



Adicionando Label's

```
from tkinter import *
app = Tk()
app.title("Show de calouros")
app.geometry('300x100+200+100')
import pygame.mixer
sounds = pygame.mixer
sounds.init()
certos = IntVar()
certos.set(0)
errados = IntVar()
errados.set(0)
def espera_tocar(canal):
    while canal.get_busy():
        pass
def musica_certa():
    s = sounds.Sound("correct.wav")
    espera_tocar(s.play())
    certos.set(certos.get() + 1)
def musica_errada():
    s = sounds.Sound("wrong.wav")
    espera_tocar(s.play())
    errados.set(errados.get() + 1)
```

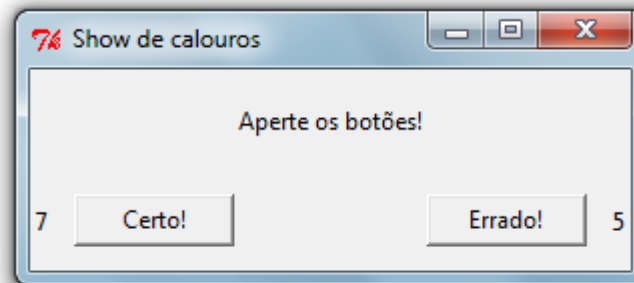
Adicionando Label's

```
lab = Label(app, text='Aperte os botões!', height = 3)
lab.pack()

lab1 = Label(app, textvariable = certos)
lab1.pack(side = 'left')
lab2 = Label(app, textvariable = errados)
lab2.pack(side = 'right')

b1 = Button(app, text = "Certo!", width = 10, command = musica_certa)
b1.pack(side = "left", padx = 10, pady = 10)
b2 = Button(app, text = "Errado!", width = 10, command = musica_errada)
b2.pack (side = "right", padx = 10, pady = 10)

app.mainloop()
```



Show Calouros GUI 04.pyw

Show de Calouros

Perfeito! E fácil de usar!
Antes que eu esqueça,
aqui vão os dois convites
para a final do Show de
Calouros!

Sorriso de vitória...



Os dois convites:
quem você vai levar?



Revisão

- Usando biblioteca de terceiros: pygame
- Loop de evento: responde apenas quando o evento ocorre
- Tratador do evento: rotina que é executada quando o evento ocorre
- pass – código python para não fazer nada
- Tkinter: biblioteca para interface gráfica