



STEM²

— Go girls! —

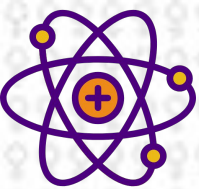


CODE

like a

girl






#girlsgonna_

Telas do Jogo



Go girls!





QUEM FOI MARIE CURIE

A

ENGENHEIRA DA NASA

B

PRIMEIRA MATEMÁTICA DO MUNDO

C

DESCOBRIDORA DO RÁDIO

D

FUNDADORA DO ITA

PULAR



PRIMEIROS PASSOS



⚛ Abrir arquivo *Tela.py*

🎮 pos[0], pos[1]

⚛ Atributos e métodos comuns a todas as telas do jogo

Tela:

imagem de fundo

comportamentoBotaoDeAudio()

comportamentoBotaoDeSair()

desenharTelaBasica()



CONSTRUTOR E ATRIBUTOS DA CLASSE



⚛ declaração de bibliotecas

```
2 from Configuracoes import *  
3 import pygame  
4 import os
```



CONSTRUTOR E ATRIBUTOS DA CLASSE



Inicialização com construtor

```
5 class Tela:  
6     def __init__(self):
```



Lembrar do prefixo "self."



Informações sobre a imagem de fundo

```
8     self.imagemDeFundo =  
pygame.image.load(os.path.join('Imagens', 'cenario.png'))  
9     self.imagemDeFundoX = 0  
10    self.imagemDeFundoX2 = self.imagemDeFundo.get_width ()
```



CONSTRUTOR E ATRIBUTOS DA CLASSE



```
# declaracao de bibliotecas
2 from Configuracoes import *
3 import pygame
4 import os
5 class Tela:
6 def __init__(self):
7 # declaracao de atributos
8 self.imagemDeFundo =
pygame.image.load(os.path.join('<Pasta>', '<Arquivo>'))
9 self.imagemDeFundoX = 0
10 self.imagemDeFundoX2 = self.imagemDeFundo.get_width ()
```




DESENHAR FUNDO NA TELA



⚛ desenha elementos comuns a toda as telas:
tela de fundo e botão de áudio

```
2 def desenharTelaBasica (self,game):  
3 game.janela.blit(self.imagemDeFundo,(self.imagemDeFundoX, 0))  
4 game.janela.blit (self.imagemDeFundo,(self.imagemDeFundoX2, 0))
```



SAIR DO JOGO



`pygame.QUIT`

ou

`pygame.KEYDOWN`

e

`pygame.key.get_pressed()[pygame.K_ESCAPE]`




SAIR DO JOGO



```
1 def comportamentoBotaoDeSair (self, game,
evento):

2 if evento.type == pygame.QUIT or
(evento.type == pygame.KEYDOWN and
pygame.key.get_pressed () [pygame.K_ESCAPE]):

3 game.usuarioSaiu = True
```

 while not game.usuarioSaiu em 'main.py'



BOTÃO DE ÁUDIO



carrega imagem do botão de audio com condições do status de áudio do jogo

```
7 if game.comAudio:
8     self.botaoSom =
pygame.image.load(os.path.join('Imagens', 'audio_ligado.png'))
9 else:
10    self.botaoSom =
pygame.image.load(os.path.join('Imagens', 'audio_desligado.png'))
11 game.janela.blit(self.botaoSom, (1200, 20))
```



COMPORTAMENTO BOTÃO DE ÁUDIO



 pressionar botão do mouse

```
pygame.MOUSEBUTTONDOWN
```

 pressionar tecla

```
pygame.KEYDOWN
```

 tecla 'm'

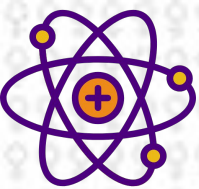


COMPORTAMENTO BOTÃO DE ÁUDIO



🎮 método para interações com o botão, em qualquer tela

```
2 def comportamentoBotaoDeAudio (self, game, evento, pos):  
  
3     if (evento.type == pygame.MOUSEBUTTONDOWN  
and pos[0]>1200 and pos[0]<1230  
and pos[1]>20 and pos[1]<45)  
or (evento.type == pygame.KEYDOWN  
and pygame.key.get_pressed () [pygame.K_m]) :  
  
4     if game.comAudio:  
5     game.administradorDeAudio.deixarSomMudo(game)  
6     else:  
7     game.administradorDeAudio.deixarSomTocar(game)
```



#girlsgonna_

Tela de Início



Go girls!





SCIENTIST GAME



INSTRUÇÕES



CONFIGURAÇÕES DE COR



 Configuracoes.py

 Padrão RGB

1 BRANCO = (255 , 255 , 255)

2 PRETO = (0 , 0 , 0)

3 VERMELHO = (255 , 0 , 0)



CONSTRUTOR E ATRIBUTOS DA CLASSE



🎮 Importar bibliotecas

```
1 # declaracao de bibliotecas
2 from Tela import *
3 import pygame
4 import os
```




CONSTRUTOR E ATRIBUTOS DA CLASSE



```
6 class TelaDeInicio(Tela):  
7     def __init__(self, game):
```

⚛ Construtor de classe-pai Tela

```
9     super().__init__()
```



CONSTRUTOR E ATRIBUTOS DA CLASSE

🎮 definindo o nome da tela

```
12 self.name = 'Tela de Inicio'
```



CONSTRUTOR E ATRIBUTOS DA CLASSE



⚛ carregando a imagem do botão de play



```
15 self.botaoPlay = pygame.image.load  
(os.path.join('Imagens', 'play_1.png'))
```



CONSTRUTOR E ATRIBUTOS DA CLASSE



⚛ carregando a fonte para o título do jogo

```
18 self.fonte1 = pygame.font.Font(os.path.join  
( 'Fontes', 'T00NISH .ttf' ), 95)
```




CONSTRUTOR E ATRIBUTOS DA CLASSE



⚛ carregando o texto do título do jogo

```
23 self.titulo = self.fonte1.render(TITULO, True, AZULBB)
```



SCIENTIST GAME



CONSTRUTOR E ATRIBUTOS DA CLASSE



⚛ carregando a fonte para o botão de instruções

```
20 self.fonte2 = pygame.font.Font(os.path.join  
( 'Fontes', 'T00NISH .ttf ' ), 70)
```



CONSTRUTOR E ATRIBUTOS DA CLASSE



⚛ carregando o texto do botão de instruções

```
26 self.inst = self.fonte2.render( 'INSTRUÇÕES', True, AZULBB)
```

INSTRUÇÕES



CONSTRUTOR E ATRIBUTOS DA CLASSE

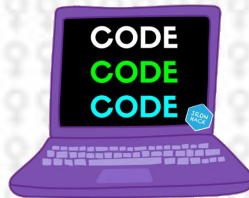


```
11 # definindo o nome da tela
12 self.name = 'Tela de Inicio '
13
14 # carregando a imagem do botao de play
15 self.botaoPlay = pygame.image.load (os.path.join('Imagens', 'play_1.png'))
16
17 # carregando a fonte para o titulo do jogo
18 self.fonte1 = pygame.font.Font(os.path.join('Fontes', 'TOONISH.ttf'), 95)
19 # carregando a fonte para o 'botao ' de instrucoes
20 self.fonte2 = pygame.font.Font(os.path.join('Fontes', 'TOONISH.ttf'), 70)
21
22 # carregando o texto do titulo do jogo
23 self.titulo = self.fonte1.render(TITULO, True, AZULBB)
24
25 # carregando o texto do botao de instrucoes
26 self.inst = self.fonte2.render('INSTRUCOES', True, AZULBB)
```

EXERCÍCIO

- Implemente o método **desenharTela(self, game)**, tendo como base o método **desenharTelaBasica(self, game)**, descrito no Código 100.

Dica: Subseção 13.2 que para desenhar algo na tela do jogo, devemos utilizar a instrução `game.janela.blit(,)..`





POR ÚLTIMO NA TELA DE INÍCIO



```
1 def desenhar(self, game):  
2     self.desenharTelaBasica (game)  
3     self.desenharTela(game)  
4     pygame.display.flip()
```




BOTÃO DE INSTRUÇÕES



🌀 método para o botão que direciona para a tela de instruções

```
2 def comportamentoBotaoDeInstrucoes(self, game, evento, pos):
```

🎮 se o cursor estiver posicionado sobre a palavra 'INSTRUÇOES '

```
4 if pos[0]>445 and pos[0]<805 and pos[1]>540 and pos[1]<600:
```

🌀 se o usuario nao tiver clicado nesta regioao

```
6 if evento . type != pygame . MOUSEBUTTONDOWN :
```

🎮 alterar a cor da palavra 'INSTRUÇOES '

```
8 self.inst = self.fonte2.render('INSTRUÇOES', True, AMARELO )
```

🌀 do contrário , mudar de tela

```
10 else :
```

```
11     game.ultimaTela = 'Tela de Inicio'
```

```
12     game.telaAtual = 'Tela de Instrucoes'
```

🎮 do contrário

```
14 else :
```

🌀 retornar a cor da palavra 'INSTRUÇOES' para a cor original

```
16 self.inst = self.fonte2.render ('INSTRUÇOES', True, AZULBB)
```



Método *run* em *AdministradorDeJogo*

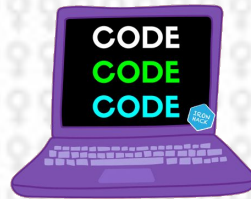
```
2 def run(self):
3     while not self.usuarioSaiu:
4         if self.telaAtual == 'Tela de Inicio':
5             self.tela = TelaDeInicio(self)
6         elif self.telaAtual == 'Tela de Instrucoes':
7             self.tela = TelaDeInstrucoes()
8         elif self.telaAtual == 'Tela de Jogo':
9             self.tela = TelaDeJogo (self)
10        elif self.telaAtual == 'Tela de Perguntas':
11            self.tela = TelaDePerguntas(self)
12        elif self.telaAtual == 'Tela Resultado da Pergunta':
13            self.tela = TelaResultadoDaPergunta()
14        else:
15            self.tela = TelaDeFim ( self )
16
17        self.tela.run ( self )
18
19    pygame.quit ()
```

EXERCÍCIO

Comportamento do Botão de Jogar

- Implemente o método `comportamentoBotaoDeJogar(self,game,evento,pos)`

Dica: tenha como base o método `comportamentoBotaoDeInstrucoes(self,game,evento,pos)` descrito no Código 106.





Interpretação de Eventos



```
1 def interpretarEventos (self, game):
```

🎮 executar o jogo com uma taxa maior ou igual a `game.fps`

```
2     game.clock.tick (game.fps)
```

```
5     for evento in pygame.event.get() :
```

```
7         pos = pygame.mouse.get_pos()
```

```
10        self.comportamentoBotaoDeSair(game, evento)
```

```
13        self.comportamentoBotaoDeAudio(game, evento, pos)
```

```
16        self.comportamentoBotaoDeInstrucoes(game, evento, pos)
```

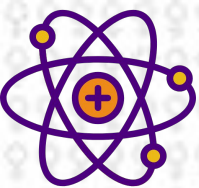
```
19        self.comportamentoBotaoDeJogar(game, evento, pos)
```



Executar *TelaDeInicio.py*



```
1 def run (self, game):  
2     while game.telaAtual == self.name and  
   not game.usuarioSaiu:  
3         self.interpretarEvento  
4         self.desenhar(game)
```

#girlsgonna_

Tela de Fim



Go girls!





FIM DE JOGO!



SCORE FINAL: 0

MELHOR SCORE: 0



```
2 from Configuracoes import *
3 from TelaDeJogo import *
4 from Tela import *
5 import pygame
6 import os
7 class TelaDeFim (Tela) :
8     def __init__(self, game) :
```


```
10     super().__init__()

13     self.name = 'Tela de Fim'

16     self.fonte1 =
pygame.font.Font(os.path.join ('Fontes',
'TOONISH.ttf'), 95)
17     self.fonte2 =
pygame.font.Font(os.path.join ('Fontes',
'TOONISH.ttf'), 65)
```



```
20     self.botaoReplay =  
pygame.image.load(os.path.join( 'Imagens',  
'replay_2.png' ))  
  
23     self.fimJogo = self.fonte1.render  
( 'FIM DE JOGO!', True, AMARELO)
```

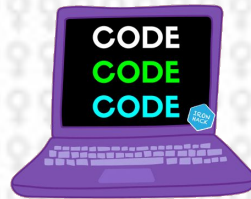
```
26         if game.pontuacao >
game.melhorPontuacao:
27             game.melhorPontuacao =
game.pontuacao

31         self.pontuacaoFinalNum =
self.fonte1.render(str(game.pontuacao), True,
AMARELO)
32         self.melhorPontuacaoNum =
self.fonte1.render(str(game.melhorPontuacao),
True, AMARELO)
```

EXERCÍCIO

- Logo abaixo da definição do texto de '**FIM DE JOGO**', da classe **TelaDeFim**, inicialize as variáveis que guardarão outros textos a serem exibidos nesta tela: "**SCORE FINAL**" e "**MELHOR SCORE**", com a cor de sua preferência, tendo como referência o Código 110.

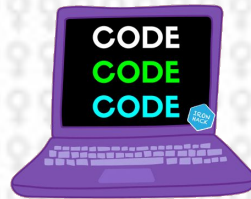
Dica: Lembre-se também de utilizar nomes facilmente legíveis para as variáveis; podemos convencionar em chamá-las de *fimJogo*, *scoreFinal* e *melhorScore*.



EXERCÍCIO

- Implemente o método **desenharTela(self, game)**, e desenhe todos os elementos desta tela, isto é: os textos '**FIM DE JOGO**', '**SCORE FINAL**' e '**MELHOR SCORE**', assim como os textos das pontuações e o botão de *replay*.

Dica: tenha como base o método **desenharTelaBasica(self, game)**, descrito no Código 100.



EXERCÍCIO

Comportamento do Botão de *Replay*

- Implemente o método

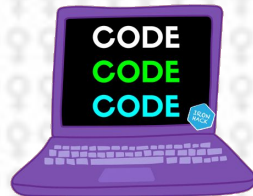
`comportamentoBotaoDeReplay(self,game,evento,pos)`

Dica: tenha como base o método

`comportamentoBotaoDeInstrucoes(self,game,evento,pos)`

descrito no Código 106.

Dica: trocar a imagem do botão sem brilho para um botão com brilho no evento do mouse estar sobre a imagem





STEM²

Go girls!

Muito obrigada!