

# Introdução ao Pygame

Criando o jogo da Cobrinha II

Python para Todos

CEFET-MG

# Conteúdo

Sistema de pontuação

Modificar a movimentação da cobra

Colocar som no jogo

Colocar mais cor

Configuração da tela inicial

Reiniciar o jogo

# Sistema de pontuação

Precisamos criar um sistema de pontuação para toda vez que a cobra comer a maçã

- ▶ Criar uma fonte para a pontuação do jogo
- ▶ Toda vez que a cobra colidir com a maçã (já montado na aula anterior), um ponto é adicionado

# Sistema de pontuação

```
# SETUP  
# ...  
pontos = 0  
font = pg.font.SysFont("Arial", 40, True)
```

# Sistema de pontuação

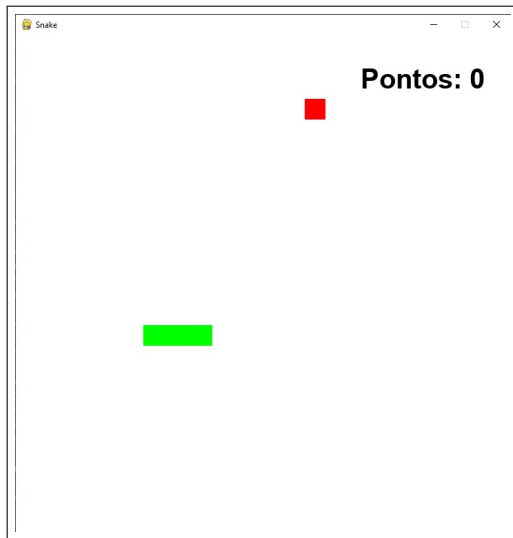
```
# LOOP PRINCIPAL
```

```
    texto=font.render(  
        f'Pontos: {pontos}',  
        True,  
        "Black"  
    )  
    tela.blit(texto, (500,40))
```

# Sistema de pontuação

```
# LOOP PRINCIPAL
    if maca.colliderect(cobra):
        pontos +=1
```

# Sistema de pontuação



# Conteúdo

Sistema de pontuação

Modificar a movimentação da cobra

Colocar som no jogo

Colocar mais cor

Configuração da tela inicial

Reiniciar o jogo



# Modificar a movimentação da cobra

Queremos limitar a movimentação da cobra

- ▶ Quando você está movimentando para a **direita**, você **não pode andar para a esquerda**
- ▶ Quando você está movimentando para a **cima**, você **não pode andar para baixo**
- ▶ Além disso, toda vez que a cobra comer a maçã, ele irá aumentar de tamanho

# Modificar a movimentação da cobra

```
# LOOP PRINCIPAL
key = pg.key.get_pressed()
if key[pg.K_w]:
    if control_pos.y != velocidade:
        control_pos.y = -velocidade
        control_pos.x = 0
elif key[pg.K_s]:
    if control_pos.y != -velocidade:
        control_pos.y = velocidade
        control_pos.x = 0
# ...
```

# Modificar a movimentação da cobra

```
# ...  
elif key[pg.K_a]:  
    if control_pos.x != velocidade:  
        control_pos.y = 0  
        control_pos.x = -velocidade  
elif key[pg.K_d]:  
    if control_pos.x != -velocidade:  
        control_pos.y = 0  
        control_pos.x = velocidade
```

# Modificar a movimentação da cobra

```
# ...  
if maca.colliderect(cobra):  
    tamanho_cobra+=5
```

# Conteúdo

Sistema de pontuação

Modificar a movimentação da cobra

**Colocar som no jogo**

Colocar mais cor

Configuração da tela inicial

Reiniciar o jogo

# Colocar som no jogo

Queremos que, ao encostar em uma maçã, o jogo coloque um som para tocar

- ▶ Iremos usar um novo comando:

```
pg.mixer.Sound('smw\_coin.wav')
```

- ▶ Iremos também baixar o som (que será o som da moedinha do Mario)
- ▶ Acesse [esse link](#) para pegar o som da moedinha.

# Colocar som no jogo

Após baixar o arquivo e colocar na pasta do jogo...

```
# SETUP
```

```
#...
```

```
colisao = pg.mixer.Sound('smw_coin.wav')
```

# Colocar som no jogo

```
# LOOP PRINCIPAL
# if maca.colliderect(cobra):
    colisao.play()
```



# Conteúdo

Sistema de pontuação

Modificar a movimentação da cobra

Colocar som no jogo

**Colocar mais cor**

Configuração da tela inicial

Reiniciar o jogo

# Colocar mais cor

Queremos que o cenário tenha mais cor

- ▶ Colocaremos uma cor especial para a cobra e para o cenário, mas você pode colocar qualquer cor!
- ▶ Acesse [esse link](#) para pegar o código hexadecimal de qualquer cor.

# Colocar mais cor

```
# SETUP
```

```
#...
```

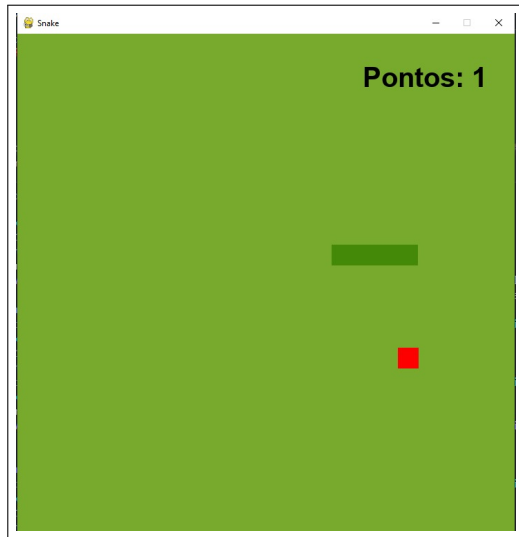
```
verde = "0x78AB2D"      # cor do fundo
```

```
verde_cobra = "0x448908" # cor da cobra
```

# Colocar mais cor

```
# LOOP PRINCIPAL
#...
tela.fill(verde)
cobra = pg.draw.rect(
    tela,
    verde_cobra,
    (cobra_pos.x, cobra_pos.y, tam, tam)
)
```

# Colocar mais cor



# Implementação de colisão

Agora, precisamos criar um sistema de colisão para a cobra, que será responsável por detectar quando o jogador perde o jogo. Esse sistema deve identificar duas situações principais:

- ▶ Colisão da cobra com seu próprio corpo.
- ▶ Colisão com as paredes do cenário.

# Implementação de colisão

Quando uma dessas colisões acontecer:

- ▶ A variável de controle será atualizada para interromper o jogo.
- ▶ Uma mensagem de “Game Over” será exibida, com a opção de reiniciar pressionando uma tecla específica.

# Delimitação das paredes e Game over

```
# SETUP
```

```
#...
```

```
morte = False
```



# Delimitação das paredes e Game over

```
# LOOP PRINCIPAL
#...
parede1 = pg.draw.line(
    tela,
    "black",
    (0,0),
    (0,altura),
    5
)
#...
```

# Delimitação das paredes e Game over

```
# ...  
parede2 = pg.draw.line(  
    tela,  
    "black",  
    (0,0),  
    (largura,0),  
    5  
)  
# ...
```

# Delimitação das paredes e Game over

```
# ...  
parede3 = pg.draw.line(  
    tela,  
    "black",  
    (largura, altura),  
    (0, altura),  
    5  
)  
# ...
```

# Delimitação das paredes e Game over

```
# ...  
parede4 = pg.draw.line(  
    tela,  
    "black",  
    (largura, altura),  
    (largura, 0),  
    5  
)  
# ...
```

# Delimitação das paredes e Game over

```
#...
if ( lista_cobra.count(lista_cabeca) > 1 or
    parede1.colliderect(cobra) or
    parede2.colliderect(cobra) or
    parede3.colliderect(cobra) or
    parede4.colliderect(cobra) ):
    morte = True
while morte:
    texto=font.render('Game over!', True, "Black")
    tela.blit(texto, (250,290))
    pg.display.update()
#...
```

# Delimitação das paredes e Game over

```
#...  
for event in pg.event.get():  
    if event.type == pg.QUIT:  
        pg.quit()
```

# Delimitação das paredes e Game over



# Conteúdo

Sistema de pontuação

Modificar a movimentação da cobra

Colocar som no jogo

Colocar mais cor

Configuração da tela inicial

Reiniciar o jogo



# Configuração da tela inicial

Vamos montar uma tela inicial que, quando o usuário apertar uma tecla, o jogo inicia

- ▶ Além disso, vamos apresentar o nome do programa

# Configuração da tela inicial

```
# SETUP  
# ...  
pg.display.set_caption("Snake")  
inicio = 1
```

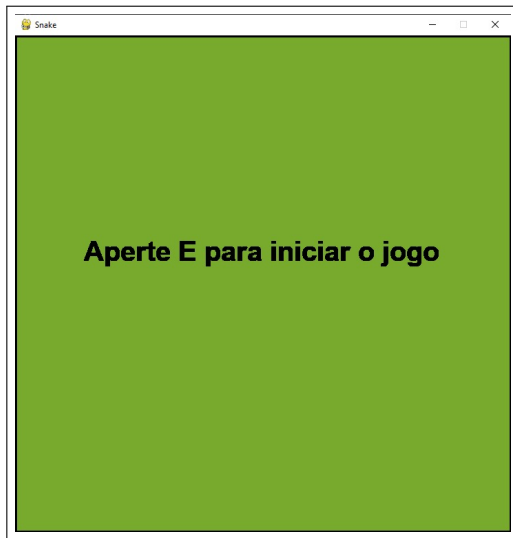
# Configuração da tela inicial

```
# LOOP PRINCIPAL
#...
while inicio:
    texto=font.render(
        'Aperte E para iniciar o jogo',
        True,
        "Black"
    )
    tela.blit(texto, (100,290))
#...
```

# Configuração da tela inicial

```
#...  
for event in pg.event.get():  
    if event.type == pg.QUIT:  
        pg.quit()  
  
    key = pg.key.get_pressed()  
    if key[pg.K_e]:  
        inicio = 0
```

# Configuração da tela inicial



# Conteúdo

Sistema de pontuação

Modificar a movimentação da cobra

Colocar som no jogo

Colocar mais cor

Configuração da tela inicial

Reiniciar o jogo

# Reiniciar o jogo

Por fim, vamos colocar, dentro do sistema de “Game Over” um algoritmo que, quando o usuário apertar a letra R no teclado, o jogo reiniciará.

- ▶ Para isso, devemos iniciar **todos os valores do jogo** (inclusive os vetores) para os valores iniciais

# Reiniciar o jogo

```
# Dentro da função de game over...
#...
for event in pg.event.get():
    if event.type == pg.QUIT:
        pg.quit()
key = pg.key.get_pressed()
if key[pg.K_r]:
    pontos = 0
    tamanho_cobra = 15
    lista_cobra = []
    lista_cabeca = []
#...
```



# Reiniciar o jogo

```
# ...
cobra_pos = pg.Vector2(
    tela.get_width()/2,
    tela.get_height()/2
)
control_pos = initial_control
morte = False
maca_pos.x = rd.randint(tam, largura-tam)
maca_pos.y = rd.randint(tam, altura-tam)
```