

SESCOMP

Introdução ao PyGAME

SESCOMP

Aula 01

Desenhando a tela basica

#Aula 01 - Desenhando a tela basica.

import pygame

import time

#iniciar o pygame

pygame.init()

#display

display_width = 640

display_height = 480

Essa resolução pode ser modificada. A unidade é em "pixels".

#setup

gameDisplay = pygame.display.set_mode((display_width,display_height))

gameDisplay.fill((255,255,255))

#atualiza tela

pygame.display.update()

pygame.display.set_caption("SESCOMP GAME")

Após um tempo, a tela desaparece.

#delay

time.sleep(2)

#fecha a tela e o jogo

pygame.quit()

quit()

Aula 02

Capturando os primeiros eventos

#Aula 02 - Capturando os primeiros eventos

#...

#atualiza tela

pygame.display.update()

pygame.display.set_caption("SESCOMP GAME")

#clock

clock = pygame.time.Clock()

#####LOGICA PRINCIPAL

fim = False

#loop do jogo

while not fim:

 #lendo uma lista de eventos [eve1,eve2,even3...]

 for evento in pygame.event.get():

 if evento.type == pygame.QUIT:

 fim = True

 #imprimindo evento

 print(evento)

 pygame.display.update()

 clock.tick(60)

#####FIM DA LOGICA PRINCIPAL

#fecha a tela e o jogo

pygame.quit()

quit()

Capturando e exibindo todos os eventos que ocorrem dentro da área da tela.

Aula 03

Desenhar o heroi na tela

#Aula 03 - Desenhar o heroi na tela

#,,,

#clock

clock = pygame.time.Clock()

#dados do heroi

heroi = pygame.image.load("emoji.png")

hx = 50

hy = 50

→ Posição em pixels do herói na tela. Eixo x e eixo y.

####LOGICA PRINCIPAL

fim = False

#loop do jogo

while not fim:

 #lendo uma lista de eventos [eve1,eve2,even3...]

 for evento in pygame.event.get():

 if evento.type == pygame.QUIT:

 fim = True

 #imprimindo evento

 print(evento)

 #apaga a tela

 gameDisplay.fill(branca)

 #desenha o heroi

 gameDisplay.blit(heroi,(hx,hy))

 #atualiza tela

 pygame.display.update()

 clock.tick(60)

→ Responsável em renderizar "emoji.png" na posição indicada por "hx" e "hy".

####FIM DA LOGICA PRINCIPAL

Aula 04

**Andando para a esquerda e para direita
(forma nao ideal)**

#Aula 04 - Andando para a esquerda e para direita (forma nao ideal)

#FUNCOES DO JOGO

```
def apaga_tela():  
    gameDisplay.fill(branca)  
  
def desenha_heroi(x,y):  
    gameDisplay.blit(heroi,(x,y))
```

#####LOGICA PRINCIPAL

```
fim = False  
#loop do jogo  
while not fim:  
    #lendo uma lista de eventos [eve1,eve2,even3...]  
    for evento in pygame.event.get():  
        if evento.type == pygame.QUIT:  
            fim = True  
        #imprimindo evento  
        print(evento)  
        #evento do teclado  
        if evento.type == pygame.KEYDOWN:  
            if evento.key == pygame.K_LEFT:  
                hx = hx - 5  
            elif evento.key == pygame.K_RIGHT:  
                hx = hx + 5  
  
        #apaga a tela  
        apaga_tela()  
        #desenha o hero  
        desenha_heroi(hx,hy)  
        #atualiza tela  
        pygame.display.update()  
        clock.tick(60)
```

#####FIM DA LOGICA PRINCIPAL

Nessa abordagem o herói apenas se move quando a tecla é pressionada. No entanto, queremos que ele se mova ENQUANTO a tecla continua pressionada.

Aula 05

**Andando para a esquerda e para direita
(forma ideal)**

```
#dados do heroi
heroi = pygame.image.load("emoji.png")
hx = 50
hy = 50
```

```
hx_change = 0
hy_change = 0
```

```
#...
```

```
#####LOGICA PRINCIPAL
```

```
fim = False
```

```
#loop do jogo
```

```
while not fim:
```

```
    #lendo uma lista de eventos [eve1,eve2,even3...]
```

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

```
        if evento.type == pygame.QUIT:
```

```
            fim = True
```

```
        #imprimindo evento
```

```
        print(evento)
```

```
        #evento do teclado
```

```
        if evento.type == pygame.KEYDOWN:
```

```
            if evento.key == pygame.K_LEFT:
```

```
                hx_change = -5
```

```
            elif evento.key == pygame.K_RIGHT:
```

```
                hx_change = 5
```

```
        if evento.type == pygame.KEYUP:
```

```
            if evento.key == pygame.K_LEFT or evento.key == pygame.K_RIGHT:
```

```
                hx_change = 0
```

Nessa abordagem o herói ENQUANTO a tecla continua pressionada. Para conseguirmos tal efeito, precisamos capturar o evento KEYDOWN.



```
#apaga a tela
```

```
apaga_tela()
```

```
#desenha o heroi
```

```
hx = hx + hx_change
```

```
hy = hy + hy_change
```

```
desenha_heroi(hx,hy)
```

```
#atualiza tela
```

```
pygame.display.update()
```

```
clock.tick(60)
```

```
#####FIM DA LOGICA PRINCIPAL
```

```
#...
```

Aula 06

Andando para a esquerda e para direita, cima e baixo (forma ideal)

```
#dados do heroi
```

```
heroi = pygame.image.load("emoji.png")
```

```
hx = 50
```

```
hy = 50
```

```
hx_change = 0
```

```
hy_change = 0
```

```
#...
```

```
#####LOGICA PRINCIPAL
```

```
fim = False
```

```
#loop do jogo
```

```
while not fim:
```

```
    #lendo uma lista de eventos [eve1,eve2,even3...]
```

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

```
        if evento.type == pygame.QUIT:
```

```
            fim = True
```

```
        #imprimindo evento
```

```
        print(evento)
```

```
        #evento do teclado
```

```
        if evento.type == pygame.KEYDOWN:
```

```
            if evento.key == pygame.K_LEFT:
```

```
                hx_change = -5
```

```
            elif evento.key == pygame.K_RIGHT:
```

```
                hx_change = 5
```

```
            elif evento.key == pygame.K_UP:
```

```
                hy_change = -5
```

```
            elif evento.key == pygame.K_DOWN:
```

```
                hy_change = 5
```

```
        if evento.type == pygame.KEYUP:
```

```
            if evento.key == pygame.K_LEFT or evento.key == pygame.K_RIGHT:
```

```
                hx_change = 0
```

```
            if evento.key == pygame.K_UP or evento.key == pygame.K_DOWN:
```

```
                hy_change = 0
```

```
#apaga a tela
```

```
apaga_tela()
```

```
#desenha o heroi
```

```
hx = hx + hx_change
```

```
hy = hy + hy_change
```

```
desenha_heroi(hx,hy)
```

```
#atualiza tela
```

```
pygame.display.update()
```

```
clock.tick(60)
```

```
#####FIM DA LOGICA PRINCIPAL
```

Mesma lógica do slide anterior mas agora para as outras duas direções restantes.

Aula 07

Incluindo limites na tela



```
#cores
```

```
branca = (255,255,255)
```

```
preto = (0,0,0)
```

```
#...
```

```
#FUNCOES DO JOGO
```

```
#...
```

```
def desenha_limites():
```

```
    pygame.draw.line(gameDisplay,preto,(20,20),(display_width-20,20),1)
```

```
    pygame.draw.line(gameDisplay,preto,(20,20),(20,display_height-20),1)
```

```
    pygame.draw.line(gameDisplay,preto,(20,display_height-20),  
                    (display_width-20,display_height-20),1)
```

```
    pygame.draw.line(gameDisplay,preto,(display_width-20,20),  
                    (display_width-20,display_height-20),1)
```

```
def ultrapassou_limiteX(meux_futuro):
```

```
    if meux_futuro<20 or meux_futuro>display_width-45:
```

```
        return True
```

```
    return False
```

```
def ultrapassou_limiteY(meuY_futuro):
```

```
    if meuY_futuro<20 or meuY_futuro>display_height-45:
```

```
        return True
```

```
    return False
```

```
#...
```

```
#apaga a tela
```

```
apaga_tela()
```

```
#desenha o heroi
```

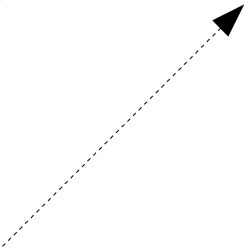
```
if(not ultrapassou_limiteX(hx+hx_change)):
```

```
    hx = hx + hx_change
```

```
if(not ultrapassou_limiteY(hy+hy_change)):
```

```
    hy = hy + hy_change
```

Os limites impedem que o herói passe da janela.



Aula 08

Desenhando os ouros


```

import random
#.....
#cores
branca = (255,255,255)
preto = (0,0,0)
amarelo = (255,255,0)
#.....
def desenha_ouros(ouros):
    for ouro in ouros:
        ouroX = ouro[0]
        ouroY = ouro[1]
        pygame.draw.circle(gameDisplay,amarelo,(ouroX,ouroY),13,0)
#.....
def criar_ouros():
    ouros = []
    for i in range(10):
        x = random.randint(50,display_width-40)
        y = random.randint(50,display_height-40)
        ouros.append([x,y])
    return ouros
#.....
####LOGICA PRINCIPAL
fim = False
ouros = criar_ouros()
#.....
#desenha ouro
desenha_ouros(ouros)
#desenha o heroi
if(not ultrapassou_limiteX(hx+hx_change)):
    hx = hx + hx_change
if(not ultrapassou_limiteY(hy+hy_change)):
    hy = hy + hy_change

```

O ouro é gerado randomicamente, colocado em uma lista e depois renderizado na tela.

Aula 09

Colidindo com o ouro

```

import pygame
import time
import random
import math
#.....
def calcula_distancia(x1,y1,x2,y2):
    dist = math.sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2))
    return dist

def colisao_ouros(meuX, meuY, ouros):
    peguei = 0
    for ouro in ouros:
        if(calcula_distancia(meuX,meuY,ouro[0]-12,ouro[1]-12)<=24):
            peguei = ouro
            break
    if(peguei!=0):
        ouros.remove(peguei)
#.....
#desenha ouro
desenha_ouros(ouros)
#desenha o heroi
if(not ultrapassou_limiteX(hx+hx_change)):
    hx = hx + hx_change
if(not ultrapassou_limiteY(hy+hy_change)):
    hy = hy + hy_change
#testa colisao
if(hx_change!=0 or hy_change!=0):
    colisao_ouros(hx,hy,ouros)

```

```

#apaga a tela
apaga_tela()
#desenha o heroi
hx = hx + hx_change
hy = hy + hy_change
desenha_heroi(hx,hy)
#atualiza tela
pygame.display.update()
clock.tick(60)

```

#####FIM DA LOGICA PRINCIPAL

*A colisão é calculada pela distância
Euclidiana entre dois pontos.*

Aula 10

Incluindo SCORE

```
#.....
def desenha_score(score):
    fonte = pygame.font.SysFont(None,25)
    texto = fonte.render("SCORE: " + str(score),True,preto)
    gameDisplay.blit(texto,(20,1))

def colisao_ouros(meuX, meuY, ouros):
    peguei = 0
    for ouro in ouros:
        if(calcula_distancia(meuX,meuY,ouro[0]-12,ouro[1]-12)<=24):
            peguei = ouro
            break
    if(peguei!=0):
        ouros.remove(peguei)
        return True
    return False
```

```
#.....
```

```
#####LOGICA PRINCIPAL
```

```
fim = False
ouros = criar_ouros()
score = 0
#loop do jogo
```

```
#.....
```

```
#desenha o heroi
if(not ultrapassou_limiteX(hx+hx_change)):
    hx = hx + hx_change
if(not ultrapassou_limiteY(hy+hy_change)):
    hy = hy + hy_change
#testa colisao
if(hx_change!=0 or hy_change!=0):
    if(colisao_ouros(hx,hy,ouros)==True):
        score = score + 10
```

*A colisão agora retorna True ou False,
valores que são testados posteriormente
com o fim de acumular pontos.*