Space Invaders em Python

Alunos: Catherine da Silveira Fleischmann

Vitor Alves de Oliveira Tamaoki

Montagem do código

- Importar biblioteca turtle;
- Organizar a tela de jogo;
- Criação das naves;
- Criação de funções para movimentação das naves;
- Criação de funções para o laser e colisão dele com as naves inimigas;
- Definição dos comandos de teclado para movimentar e atirar;

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

Montagem do código

- Aplicação do laço while true;
- Criação de laços e condicionais para movimentação e posicionamento de naves inimigas;
- Condicional para verificar se houve colisão entre o laser e o inimigo;
- Game over;
- Condicional de movimentação do laser.

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

O código

```python
from turtle import *
from math import sqrt
import random
import winsound
import os
from tkinter import *


#Organiza a tela
tela = Screen()
tela.bgcolor("black")
tela.title("Space Invaders")
tela.setworldcoordinates(-400, -400, 400, 400) #resolução tela
tela.bgpic("space_invaders_background.gif")
#Resgitra os formato da nave e dos invaders.
register_shape("invader.gif")
register_shape("player.gif")
register_shape("laser.gif")
```

- Importação da biblioteca turtle;
- Início da organização da tela;

O código

```
21  borda = Turtle()
22  borda.speed(0)
23  borda.color("white")
24  borda.penup()
25  borda.setpos(-300,-300)
26  borda.pendown()
27  borda.pensize(3)
28  borda.hideturtle()  #esconde a caneta
29  for side in range(4):  #Desenha a borda
30      borda.fd(600)
31      borda.lt(90)
32  #Score
33  score = 0
34  #Desenha o score
35  score_pen = Turtle()
36  score_pen.speed(0)
37  score_pen.color("white")
38  score_pen.penup()
39  score_pen.setposition(-290, 270)
40  scorestring = "Score: %s" %score
41  score_pen.write(scorestring, False, align="left", font=("Arial", 10, "normal"))
```

- Organização da tela: bordas e score;

O código

```
score_pen.hideturtle()

#Desenhando a nave
nave = Turtle()
nave.hideturtle()
nave.shape("player.gif")
nave.color("green")
nave.penup() #Esconde o rastro da caneta.
nave.seth(90)
nave.speed(0)
nave.setposition(0, -280) #Posição inicial da nave
nave.showturtle()
mov_nave = 15  #Velocidade da nave

#Numero de inimigos
n_inimigos = 5
#Lista de inimigos
inimigos = []
#Adiciona os inimigos na lista.
for i in range(n_inimigos):
    #Acrescenta turtles dentro da lista
    inimigos.append(Turtle())
```

- Organização da tela: naves e primeiro laço de repetição;

O código

```
for inimigo in inimigos:

    inimigo.shape("invader.gif")
    inimigo.color("red")
    inimigo.speed(0)
    inimigo.penup()
    x = random.randint(-200, 200) #Posição aleatoria do inimigo
    y = random.randint(100, 250)  #Posição aleatoria do inimigo
    inimigo.setpos(x, y)


mov_inimigo = 2  # Velocidade do inimigo

#Laser da nave
laser = Turtle()
laser.hideturtle()
laser.shape("laser.gif")
laser.color("yellow")
laser.penup()
laser.seth(90)
laser.speed(0)
laser.shapesize(0.5, 0.5)
mov_laser = 20 #velocidade do laser
 #Estados do laser.
#ready - Preparado para atirar o laser.
#fire - Enquanto o laser é atirado.
est_laser = "ready"
```

- Organização da tela: determinação de posicionamento aleatório das naves inimigas e criação do laser;

O código

```
def esquerda():
    x = nave.xcor() #Retorna posição x da nave.
    x -= mov_nave
    if x < -280:  #Limita o movimento para a borda
        x = - 280
    nave.setx(x) #A nave fica na posição x.
def direita():
    x = nave.xcor()
    x += mov_nave
    if x > 280:   #Limita o movimento para a borda
        x = 280
    nave.setx(x)
def solta_laser():
    global est_laser #Define como uma variavel global, caso precise de mudança.
    if est_laser == "ready":
        winsound.PlaySound("laser", winsound.SND_ASYNC)
        est_laser = "fire"
    #Posição do laser em relação a nave:
        x = nave.xcor()
        y = nave.ycor()
        laser.setpos(x, y + 10)
        laser.showturtle()
```

- Funções de movimentação da nave do usuário;
- Função laser;

UFPR 105 ANOS DE ORGULHO

O código

```
def eColisao(t1, t2):
    """
    Calcula a distância entre o laser e o inimigo e determina se houve colisão.
    """
    distancia = sqrt(pow(t1.xcor()-t2.xcor(), 2) + pow(t1.ycor()-t2.ycor(), 2))
    if distancia < 15:
        return True
    else:
        return False

listen() #Espera comando do teclado.
onkey(esquerda, "Left") #Movimenta para a esquerda.
onkey(direita, "Right") #Movimenta para a direita.
onkey(solta_laser, "space") #Solta o laser.

#Parte principal do jogo:
while True:
    for inimigo in inimigos:
        #Movimenta o inimigo.
        x = inimigo.xcor()
        x += mov_inimigo
        inimigo.setx(x) #muda para a nova posição
```

- Função de colisão do laser com a nave inimiga;
- Inserção dos comando utilizados pelo usuário;
- Início dos laços e condicionais utilizados para determinar a movimentação e o posicionamento das naves inimigas;

UFPR 105 ANOS DE ORGULHO

O código

```
if inimigo.xcor() > 280:
    #Movimenta todos os inimigos para baixo
    for e in inimigos:
        y = e.ycor()
        y -= 40
        e.sety(y)
    #Muda direção
    mov_inimigo *= -1
if inimigo.xcor() < -280:
    # Movimenta todos os inimigos para baixo
    for e in inimigos:
        y = e.ycor()
        y -= 40
        e.sety(y)
    # Muda direção
    mov_inimigo *= -1
#Verifica a colisão do laser com o inimigo.
if eColisao(laser, inimigo):
    winsound.PlaySound("explosion", winsound.SND_ASYNC)
    #reseta o laser após a colisão com o alvo
    laser.hideturtle()
    est_laser = "ready" #permite o laser ser atirado novamente após a colisão com o alvo
    laser.setposition(0, -400)
    #reseta o inimigo
    x = random.randint(-200, 200)  # Posição aleatoria do inimigo
    y = random.randint(100, 250)  # Posição aleatoria do inimigo
```

- Continuação do laço while true;
- Condicional que determina o comportamento do jogo caso o laser atinja a nave inimiga;

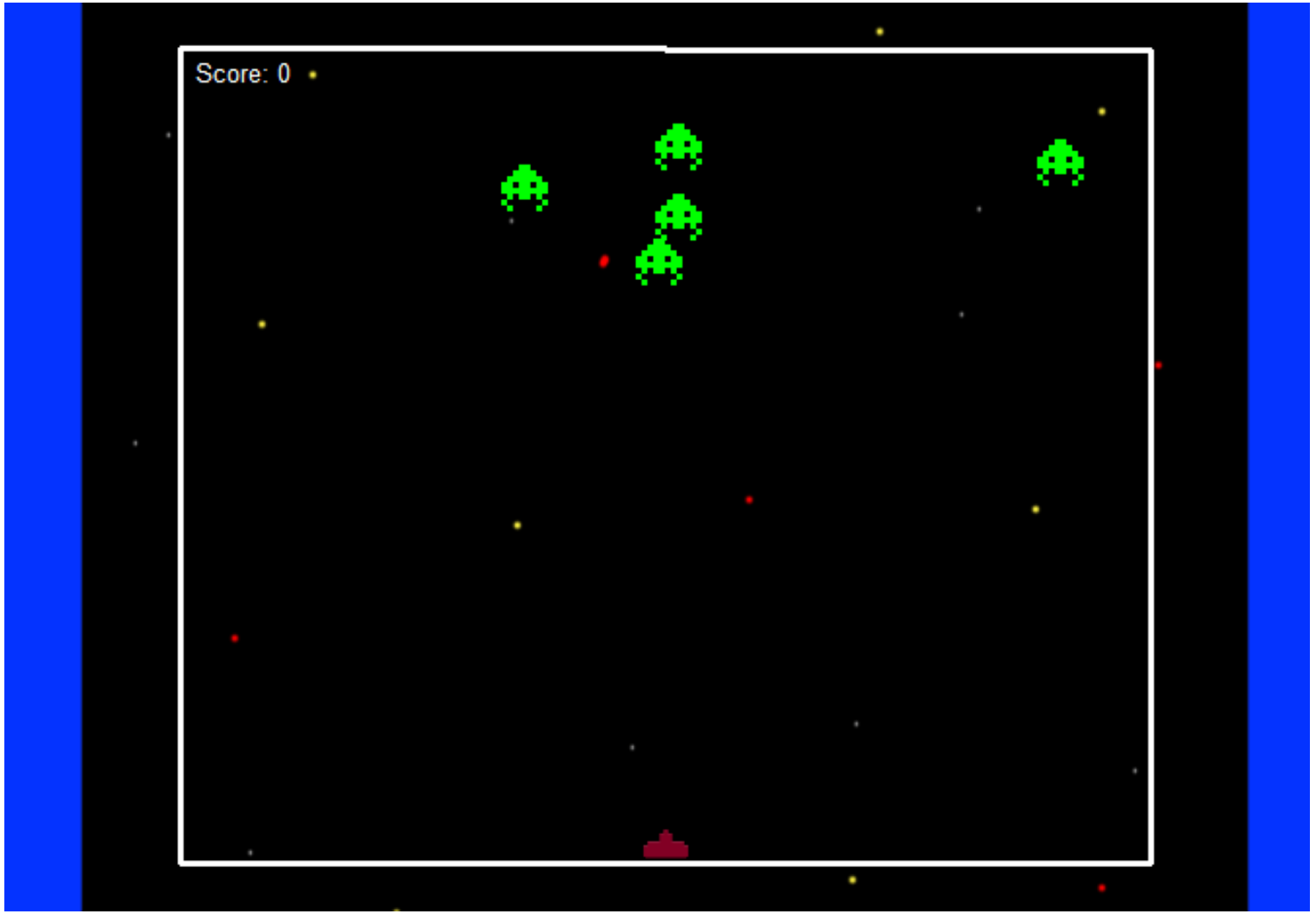
O código

```
        inimigo.setpos(x, y)
        #Atualiza o score
        score += 10
        scorestring = "Score: %s" %score
        score_pen.clear()    #Limpa o score
        score_pen.write(scorestring, False, align="left", font=("Arial", 10, "normal"))

    #Colisão entre a nave e o inimigo
    if eColisao(nave, inimigo):
        nave.hideturtle()
        inimigo.hideturtle()
        print("Game Over")
        break

#Movimento do laser:
if est_laser == "fire":
    y = laser.ycor()
    y += mov_laser
    laser.sety(y)
#Verifica se o laser atingiu a borda:
if laser.ycor() > 275:
    laser.hideturtle()
    est_laser = "ready"
```

- Continuação da condicional de colisão;
- Game over;
- Movimentação do laser.

UFPR 105 ANOS DE ORGULHO

O jogo



- Tela do jogo em funcionamento.