

Aula 6

Introdução

Nessa aula iremos dar continuidade na codificação do jogo “**Falling Apples**” em python utilizando a biblioteca gráfica **turtle**.

Essa aula será dividida em 3 partes: Criação da Maçã, Criação de N Maças e Criação de N Maças “Podres”.

Criação da Maçã

```
import turtle
import random

#Criação da tela do jogo
jogo = turtle.Screen()
jogo.title("Falling Apples")
jogo.bgcolor("light blue")
jogo.setup(width=800, height=600)
jogo.tracer(0)

#Criação da classe de Elementos (Pai)
class Elemento(turtle.Turtle):

    def __init__( self , formato, cor, posicaoX, posicaoY, velocidade):
        turtle.Turtle.__init__(self)
        self.speed(velocidade)
        self.penup()
        self.shape(formato)
        self.color(cor)
        self.goto(posicaoX, posicaoY)
        self.direction = "stop"

    def colidiu(personagem0, personagem1):
        if personagem1.distance(personagem0) < 40:
            return True
        else:
            return False

#Criando a classe da cesta (jogador)
class Cesta(Elemento):
    def __init__( self , formato, cor, posicaoX, posicaoY, velocidade):
        Elemento.__init__( self , formato, cor, posicaoX, posicaoY, velocidade)
        self.pontuacao = 0
        self.vidas = 0
```

```

        self.direction = "stop"

    def movimenta_para_direita(self):
        self.direction = "right"

    def movimenta_para_esquerda(self):
        self.direction = "left"

    def movimenta(self, distancia):
        x = self.xcor()
        x += distancia
        if x < 390 and x > -390:
            self.setx(x)
        self.direction = "stop"

    def movimenta_cesta(self):
        if self.direction == "right":
            self.movimenta(20)
        if self.direction == "left":
            self.movimenta(-20)

#Criando a classe da maçã
class Maca(Elemento): #6
    def __init__(self, formato, cor, posicaoX, posicaoY, velocidade, podre, som): #7
        Elemento.__init__(self, formato, cor, posicaoX, posicaoY, velocidade) #8
        self.podre = podre
        self.som = som

    def reposiciona_maca(self): #9
        x = random.randint(-380, 380) #10
        y = random.randint(200, 300) #11
        self.goto(x, y) #12

    def movimenta(self, cesta): #13
        y = self.ycor() #14
        y -= (self.speed/10) #15
        self.sety(y) #16
        if self.ycor() < -300: #17
            self.reposiciona_maca() #18

        if Elemento.colidiu(cesta, self): #19
            self.reposiciona_maca() #20

# Adicionar o jogador
cesta = Cesta("square", "blue", 0, -250, 0)

#Adicionar uma maçã
maca = Maca("circle", "red", random.randint(-380, 380), random.randint(200, 300),
random.randint(1, 3), False, "") #21

# Keyboard Binding
jogo.listen()

```

```
jogo.onkeypress(cesta.movimenta_para_esquerda, "Left")
jogo.onkeypress(cesta.movimenta_para_direita, "Right")

# Loop Principal do Jogo
while True:
    jogo.update()

    #Movimenta a maçã
    maca.movimenta(cesta) #22

    #Movimenta o jogador
    cesta.movimenta_cesta()

jogo.mainloop()
```

Código 1: Criação da maçã

Para facilitar o entendimento iremos explicar linha a linha do nosso código:

#1 : Criaremos uma função na classe Elemento que será responsável por verificar se ocorreu uma colisão.

#2 : Verificamos por meio de um if se a distância entre dois personagens/elementos é menor que 40. É possível selecionar outros valores, porém com 40 temos a melhor jogabilidade.

#3 : Retorna True, ou seja, que ocorreu uma colisão.

#4 : Senão, ou seja, se a distância entre os dois personagens/elementos é maior ou igual a 40.

#5 : Retorna False, ou seja, que não ocorreu uma colisão.

#6 : Iremos criar uma classe que será responsável pela maçã. Chamaremos ela de Maca, e a mesma é filha da classe Elemento e logo neta da classe Turtle, ou seja, a classe Maca herdará os atributos e métodos/funções da classe pai (Elemento) e da classe avô (Turtle).

#7 : Criaremos a função `__init__` essa é uma função padrão das classes do python e a mesma é a primeira função executada após a criação de um objeto dessa classe.

#8 : Chamamos o `__init__` da classe pai (Elemento).

#9 : Criaremos a função `reposiciona_maca` que será responsável pelo reposicionamento da maçã na tela do jogo.

#10 : Definimos a variável `x` com um valor randômico entre -380 e 380. Fazemos isso para que a maçã não fique sempre aparecendo na mesma posição horizontal.

#11 : Definimos a variável `y` com um valor randômico entre 200 e 300. Fazemos isso para que a maçã não fique sempre aparecendo na mesma posição vertical.

#12 : Movemos a maçã para as coordenadas `x` e `y` definidas anteriormente.

#13 : Criaremos a função movimenta que será responsável pela movimentação da maçã, se ocorreu colisão ou não com a cesta e se a maçã atingiu o final da tela do jogo.

#14 : Obtém a coordenada vertical y da maçã na tela do jogo.

#15 : Subtrai a velocidade da maçã dividida por 10 da coordenada vertical y da maçã. Isso é necessário para movimentar a maçã verticalmente. Dividimos a velocidade da maçã por 10 caso contrário a maçã irá se movimentar muito rapidamente, assim deixando o jogo muito difícil para o jogador.

#16 : Define a nova posição vertical y da maçã.

#17 : Verifica se a nova posição vertical y da maçã é menor que -300. Sendo -300 o “final da tela” do nosso jogo, ou seja, verifica se a maçã chegou no “final da tela”.

#18 : Caso a maçã tenha chegado no “final da tela” reposiciona ela chamando a função reposiciona_maca, a qual irá posicioná-la randomicamente na tela.

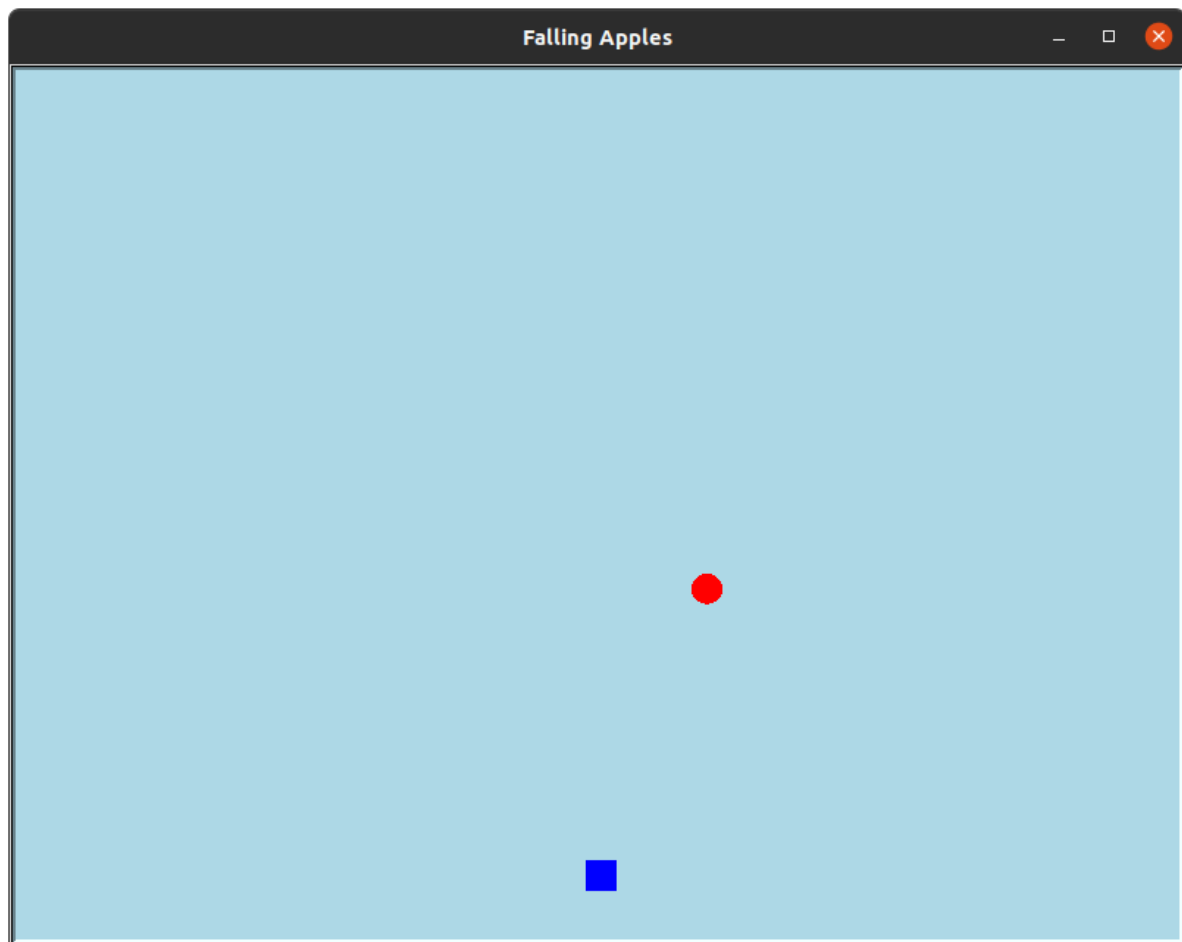
#19 : Verifica se ocorreu uma colisão entre a maçã e a cesta.

#20 : Caso ocorreu colisão, reposiciona a maçã chamando a função reposiciona_maca, a qual irá posicioná-la randomicamente na tela.

#21 : Criamos o objeto maca da classe Maca, passando como formato “circle” (círculo), a cor “red” (vermelha), a posição horizontal x como um número randômico entre -380 e 380, a posição vertical y como um número randômico entre 200 e 300, a velocidade como um número randômico entre 1 e 3, falso como podre e o nome do áudio/som como vazio. Esse objeto será a nossa maçã.

#22 : Chamamos a função movimenta da maçã, a qual realizará a movimentação da maçã pela tela do jogo.

Segue abaixo o resultado da execução do nosso código:



Tela 1: Criação da Maçã

Criação de N Maçãs

Primeiramente iremos criar uma variável que será responsável por armazenar a quantidade máxima de maçãs que serão criadas/apresentadas na tela do jogo por vez.

A variável será criada entre o import random e a criação da tela do jogo, conforme as imagens abaixo:

```
import random

#Criação da tela do jogo
jogo = turtle.Screen()
jogo.title("Falling Apples")
```

Código 2: Criação de N Maçãs - Posição para a criação da variável numero_de_macas

```
import random

numero_de_macas = 10

#Criação da tela do jogo
jogo = turtle.Screen()
jogo.title("Falling Apples")
```

Código 3: Criação de N Maças - Criação da variável numero_de_macas

Teremos 10 maçãs simultaneamente na tela do nosso jogo por vez.

Após a criação da variável, iremos adicionar as maçãs na tela, para isso é necessário alterar a criação do objeto maca para um vetor de objetos do tipo Maca.

```
#Adicionar uma maçã
maca = Maca("circle", "red", random.randint(-380, 380), random.randint(200, 300),
random.randint(1, 3), False, "")
```

Código 4: Criação de N maçãs - Criação do objeto maca, ele será substituído pelo vetor de maçãs

Substituiremos o código acima, pelo seguinte código:

```
#Cria lista de maçãs
macas = []

#Adicionar as maçãs
for _ in range(numero_de_macas):
    macas.append(Maca("circle", "red", random.randint(-380, 380), random.randint(200, 300),
random.randint(1, 3), False, ""))
```

Código 5: Criação de N Maças - Criação do vetor de maçãs de acordo com a variável de número de maçãs

Como podemos observar no código acima, primeiramente criamos um vetor macas vazio, representado pelo []. Após isso, criamos um for que para cada execução cria um objeto Maca (maçã) e adiciona o mesmo no vetor de maçãs. Esse for será executado n vezes sendo n igual ao valor definido pela variável numero_de_macas.

Após a criação do vetor de maçãs, será necessário realizar a movimentação de cada maçã na tela do jogo. Para isso será necessário substituir o trecho “Código 6” pelo trecho “Código 7”.

```
#Movimenta as maçãs  
maca.movimenta(cesta)
```

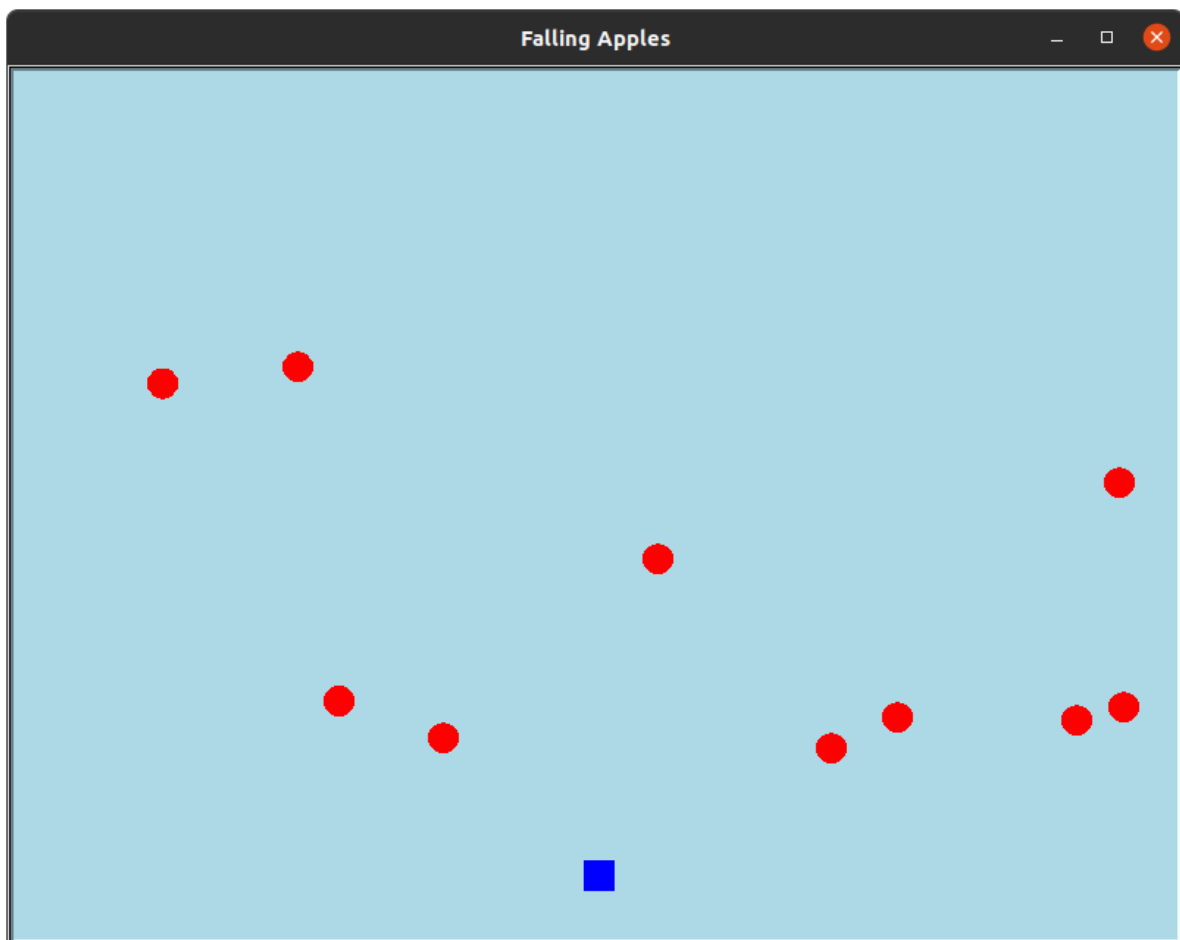
Código 6: Criação de N Maçãs - Movimenta 1 maçã pela tela do jogo

```
#Movimenta as maçãs  
for maca in macas:  
    maca.movimenta(cesta)
```

Código 7: Criação de N Maçãs - Movimenta todas as maçãs presentes no vetor de maçãs pela tela do jogo

O código acima, é um for que para cada maçã existente no vetor de maçãs (macas), chama a função movimenta da mesma, a qual realiza a movimentação da maçã pela tela do jogo.

Segue abaixo o resultado da execução do nosso código:



Tela 2: Criação de N Maçãs

Criação de N Maçãs “Podres”

Primeiramente iremos criar uma variável que será responsável por armazenar a quantidade máxima de maçãs “podres” que serão criadas/apresentadas na tela do jogo por vez.

A variável será criada abaixo da variável `numero_de_macas`, conforme as imagens abaixo:

```
import random

numero_de_macas = 10

#Criação da tela do jogo
jogo = turtle.Screen()
jogo.title("Falling Apples")
```

Código 8: Criação de N Maçãs “Podres” - Posição para a criação da variável `numero_de_macas_podres`

```
import random

numero_de_macas = 10
numero_de_macas_podres = 5

#Criação da tela do jogo
jogo = turtle.Screen()
jogo.title("Falling Apples")
```

Código 9: Criação de N Maçãs “Podres” - Criação da variável `numero_de_macas_podres`

Teremos 5 maçãs podres simultaneamente na tela do nosso jogo por vez.

Após a criação da variável, iremos adicionar as maçãs podres na tela, para isso iremos clonar o vetor de maçãs e renomeamos o mesmo como `macas_podres`.

```
#Cria lista de maçãs
macas = []

#Adicionar as maçãs
for _ in range(numero_de_macas):
    macas.append(Maca("circle", "red", random.randint(-380, 380), random.randint(200, 300),
random.randint(1, 3), False, ""))

#Cria lista de maçãs podres
macas_podres = []

#Adicionar as maçãs podres
for _ in range(numero_de_macas_podres):
    macas_podres.append(Maca("circle", "black", random.randint(-380, 380),
```



```
random.randint(200, 300), random.randint(1, 3), True, "")

# Keyboard Binding
jogo.listen()
```

Código 10: Criação de N Maças “Podres” - Criação do vetor de maçãs podres

Como podemos observar no código acima, primeiramente criamos um vetor `macas_podres` vazio, representado pelo []. Após isso, criamos um `for` que para cada execução cria um objeto `Maca` (maçã podre) e adiciona o mesmo no vetor de maçãs podres. Esse `for` será executado `n` vezes sendo `n` igual ao valor definido pela variável `numero_de_macas_podres`. O diferencial entre a maçã e a maçã podre é que a maçã é representada por um círculo vermelho, já a maçã podre é representada por um círculo preto.

Após a criação do vetor de maçãs, será necessário realizar a movimentação de cada maçã na tela do jogo. Para isso podemos clonar a movimentação das maçãs e alterar o vetor de maçãs pelo vetor de maçãs podres.

```
# Loop Principal do Jogo
while True:
    jogo.update()

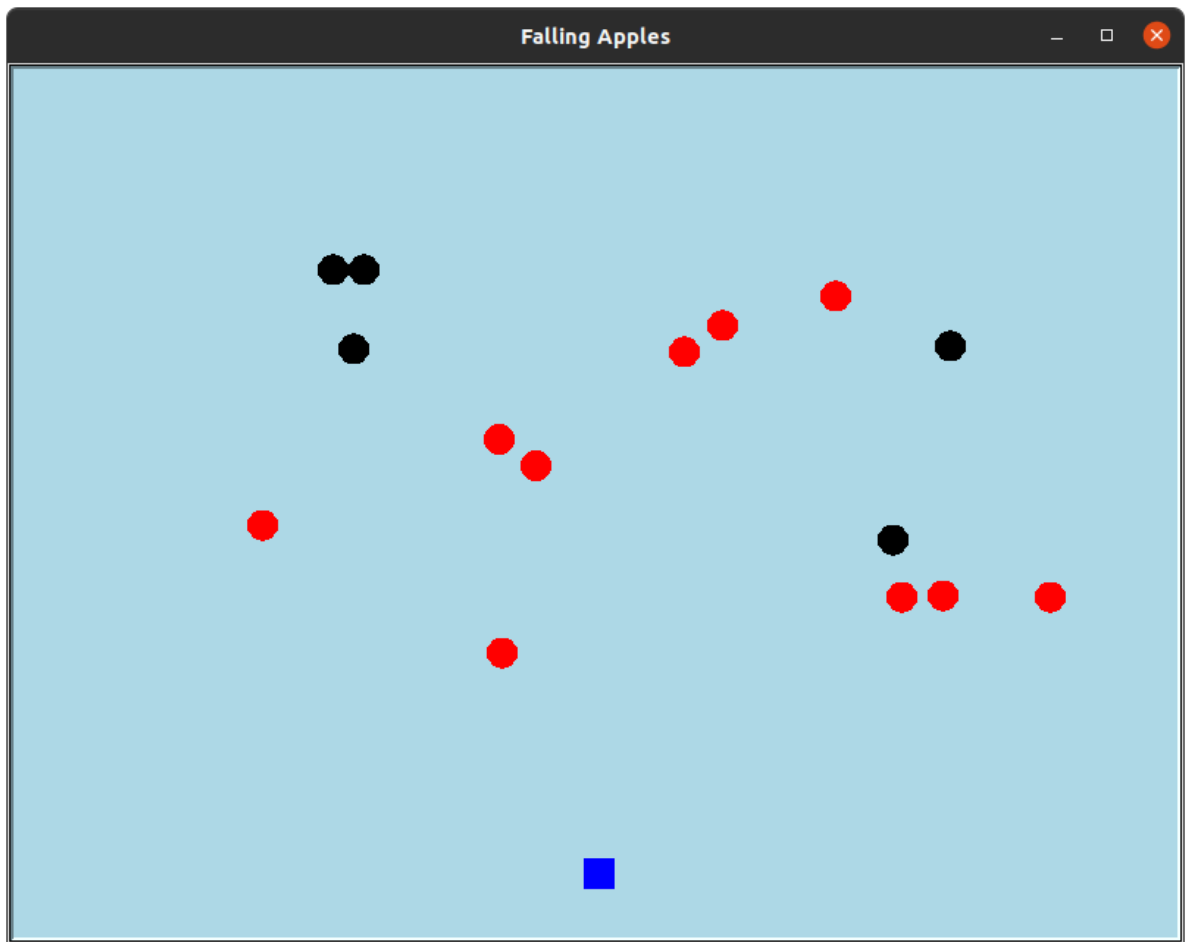
    #Movimenta as maçãs
    for maca in macas:
        maca.movimenta(cesta)

    #Movimenta as maçãs podres
    for maca_podre in macas_podres:
        maca_podre.movimenta(cesta)

    #Movimenta o jogador
    cesta.movimenta_cesta()
```

Código 11: Criação de N Maças “Podres” - Movimentação das maçãs podres na tela do jogo

O código acima, é um `for` que para cada maçã podre existente no vetor de maçãs podres (`macas_podres`), chama a função `movimenta` da mesma, a qual realiza a movimentação da maçã podre pela tela do jogo.



Tela 3: Criação de N Maças “Podres”