



Rafael Vieira Coelho

rafaelvc2@gmail.com

Fazendo a Nave Atirar

- Iremos adicionar um pequeno projétil quando o usuário clicar na tecla espaço.
- Os projéteis se deslocarão para cima até sumirem da tela.
- Precisamos criar uma nova classe no arquivo `bullet.py` e adicionar as configurações dos projéteis no arquivo `settings.py`

settings.py

```
1  #!/usr/bin/env python
2  # coding: utf8
3
4  __AUTHOR__ = "Rafael Vieira Coelho"
5  __DATE__ = "19/05/2019"
6
7  class Settings():
8
9      def __init__(self):
10         self.largura = 1200
11         self.altura = 800
12         self.cor_fundo = (230, 230, 230)
13         self.nave_velocidade = 5
14         self.bala_velocidade = 15
15         self.bala_largura = 3
16         self.bala_altura = 15
17         self.bala_cor = (60, 60, 60) #cinza-escuro
```

bullet.py

```
1  #!/usr/bin/env python
2  # coding: utf8
3
4  __AUTHOR__ = "Rafael Vieira Coelho"
5  __DATE__ = "20/05/2019"
6
7  import pygame
8  from pygame.sprite import Sprite
9
10 class Bullet(Sprite): #https://github.com/myint/pygame/blob/master/lib/sprite.py
11
12     def __init__(self, config, tela, nave):
13         super(Bullet, self).__init__()
14         self.tela = tela
15         self.retangulo = pygame.Rect(0, 0, config.bala_largura, config.bala_altura)
16         self.retangulo.centerx = nave.retangulo.centerx
17         self.retangulo.top = nave.retangulo.top
18         self.y = float(self.retangulo.y)
19         self.cor = config.bala_cor
20         self.velocidade = config.bala_velocidade
21
22     def atualiza(self):
23         self.y -= self.velocidade
24         self.retangulo.y = self.y
25
26     def desenha(self):
27         pygame.draw.rect(self.tela, self.cor, self.retangulo)
```

sprite.py

- Arquivo de pygame

```
106 class Sprite(object):
107     """simple base class for visible game objects
108
109     pygame.sprite.Sprite(*groups): return Sprite
110
111     The base class for visible game objects. Derived classes will want to
112     override the Sprite.update() method and assign Sprite.image and Sprite.rect
113     attributes. The initializer can accept any number of Group instances that
114     the Sprite will become a member of.
115
116     When subclassing the Sprite class, be sure to call the base initializer
117     before adding the Sprite to Groups.
118
119     """
120
121     def __init__(self, *groups):
122         self.__g = {} # The groups the sprite is in
123         if groups:
124             self.add(*groups)
125
126     def add(self, *groups):
127         """add the sprite to groups
128
129         Sprite.add(*groups): return None
130
131         Any number of Group instances can be passed as arguments. The
132         Sprite will be added to the Groups it is not already a member of.
133
134         """
135         has = self.__g.__contains__
136         for group in groups:
137             if hasattr(group, '_spritegroup'):
138                 if not has(group):
139                     group.add_internal(self)
140                     self.add_internal(group)
141             else:
142                 self.add(*group)
```

alien_invasion.py

```
1  #!/usr/bin/env python
2  # coding: utf8
3
4  __AUTHOR__ = "Rafael Vieira Coelho"
5  __DATE__ = "19/05/2019"
6
7  import sys, pygame
8  import settings as s
9  from ship import Ship
10 import game_functions as g
11 from pygame.sprite import Group
12
13 def main():
14     pygame.init() #inicia o modo gráfico do pygame
15     config = s.Settings() #cria o objeto de configurações
16     tela = pygame.display.set_mode((config.largura, config.altura))
17     pygame.display.set_caption("Alien Invasion") #define o título
18     nave = Ship(config, tela)
19     balas = Group()
20     while True:
21         g.testa_eventos(config, tela, nave, balas)
22         g.atualiza_tela(config, tela, nave, balas)
23         nave.atualiza(config)
24         for bala in balas.sprites():
25             if bala.retangulo.bottom <= 0:
26                 balas.remove(bala)
27             else:
28                 bala.atualiza()
29         print(str(len(balas)) + ' balas')
30 main()
```

game_functions.py

```
8 import sys, pygame
9 from bullet import Bullet
10
11 #responde a eventos de pressionamento de teclas e mouse
12 def testa_eventos(config, tela, nave, balas):
13     for evento in pygame.event.get(): #verifica todos os eventos ocorridos
14         if evento.type == pygame.QUIT: #verifica se a janela foi fechada
15             sys.exit()
16         elif evento.type == pygame.KEYDOWN: #quando aperta uma tecla
17             testa_eventos_aperta_tecla(evento, config, tela, nave, balas)
18         elif evento.type == pygame.KEYUP: #quando solta a tecla
19             testa_eventos_solta_tecla(evento, nave)
20
21 #atualiza as imagens na tela e redesenha a mesma
22 def atualiza_tela(config, tela, nave, balas):
23     tela.fill(config.cor_fundo) #preenche a janela com a cor de fundo
24     nave.desenha() #desenha a nave na tela
25     #nave.atualiza(config)
26     for bala in balas.sprites():
27         bala.desenha()
28     pygame.display.flip() #redesenha a tela
29
30 def testa_eventos_aperta_tecla(evento, config, tela, nave, balas):
31     print(evento.key, chr(evento.key))
32     if evento.key == pygame.K_RIGHT: # Move a espaçonave para a direita
33         nave.move_direita = True
34     elif evento.key == pygame.K_LEFT: # Move a espaçonave para a esquerda
35         nave.move_esquerda = True
36     elif evento.key == pygame.K_UP: # Move a espaçonave para cima
37         nave.move_cima = True
38     elif evento.key == pygame.K_DOWN: # Move a espaçonave para baixo
39         nave.move_baixo = True
40     elif evento.key == pygame.K_SPACE:
41         nova_bala = Bullet(config, tela, nave)
42         balas.add(nova_bala)
43     elif evento.key == pygame.K_q: #https://www.pygame.org/docs/ref/key.html
44         sys.exit(1)
```

Tarefas

1. Escreva na tela o conteúdo `evento.key` no laço de repetição que trata os eventos. Quando executar, aperte várias teclas para ver o que é mostrado na tela.
2. Adicione um limite de balas que a nave pode disparar. Defina o mesmo em `settings.py`. Dica: modifique o método `testa_eventos_aperta_tecla()`.
3. Modifique o código criado na aula passada para que possam ser disparadas balas para a direita da tela.