

Minicurso de Programação de Jogos em Python

Aula 3

Movimentação background

Acesse



aebescolavirtual.aeb.gov.br/

Professor: Frederick Campbell - Aluno de Graduação do ITA



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

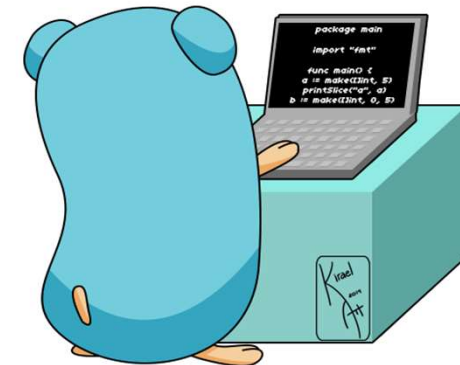




Tópicos

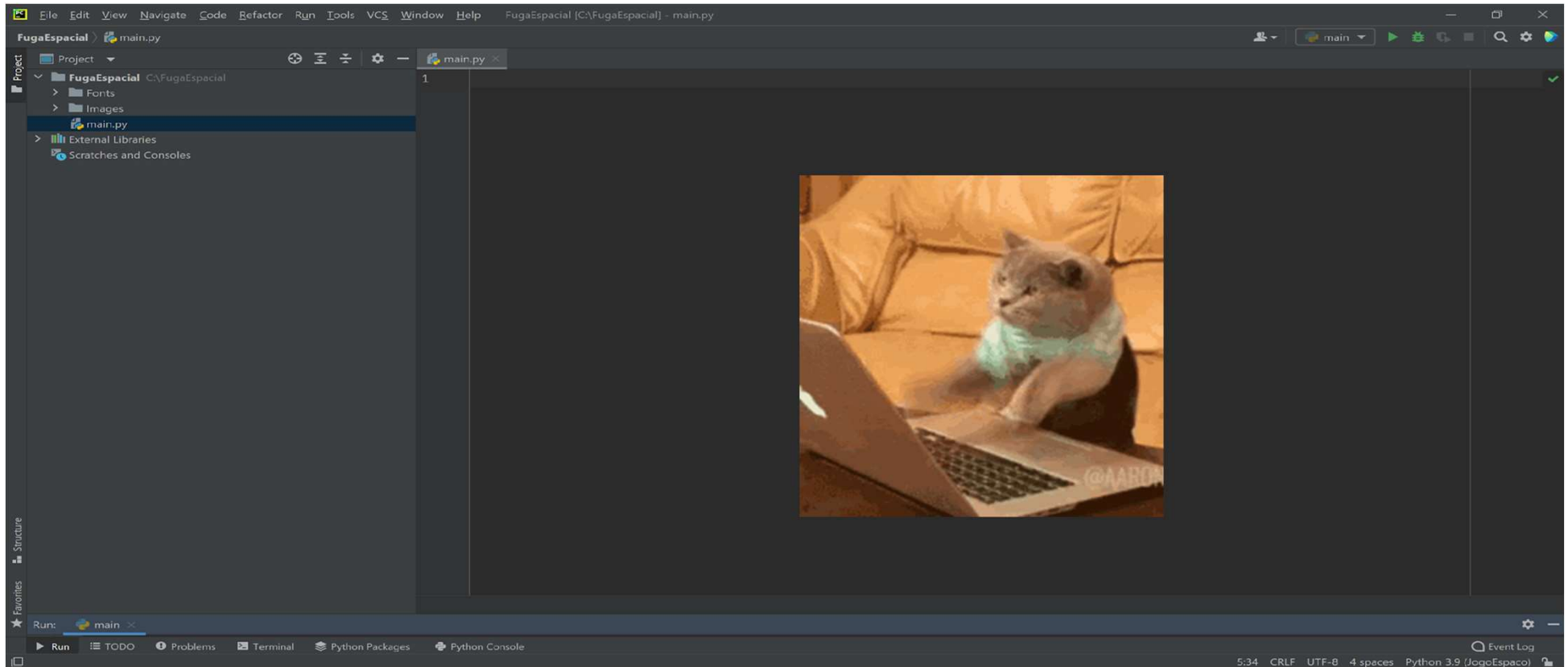


- ✈ Inserir imagens no Plano de fundo/ Background
- ✈ Mover o plano de fundo/background





Vamos continuar nosso código!



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

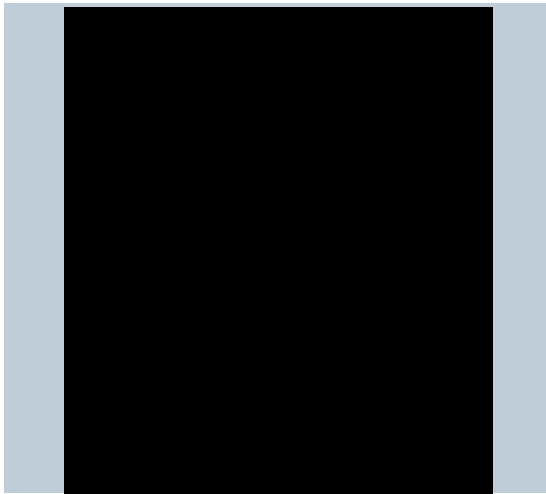




Inserindo as margens

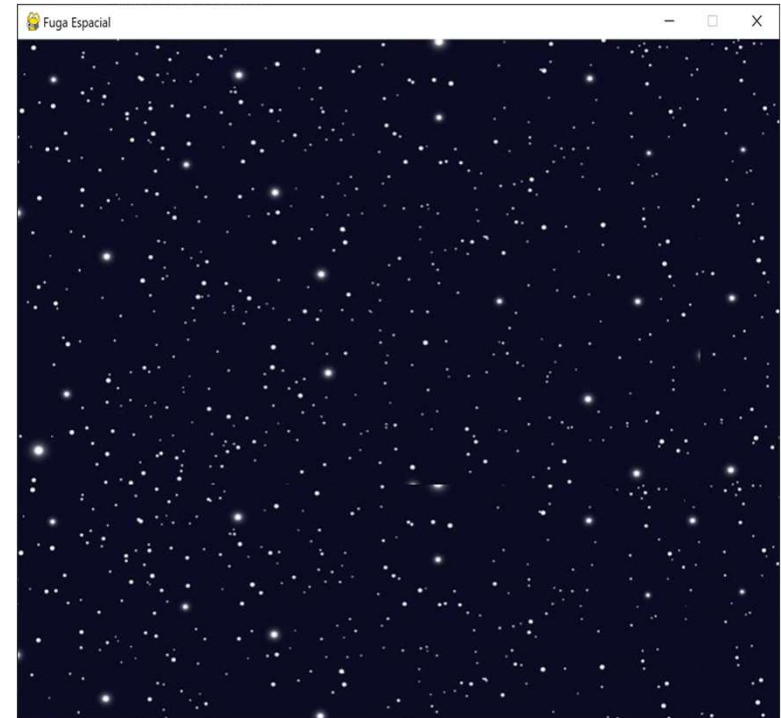


🚀 Mas o que são margens?



Exemplo de uma imagem com margem do lado direito e esquerdo.

Nosso jogo está assim, vamos avançar??

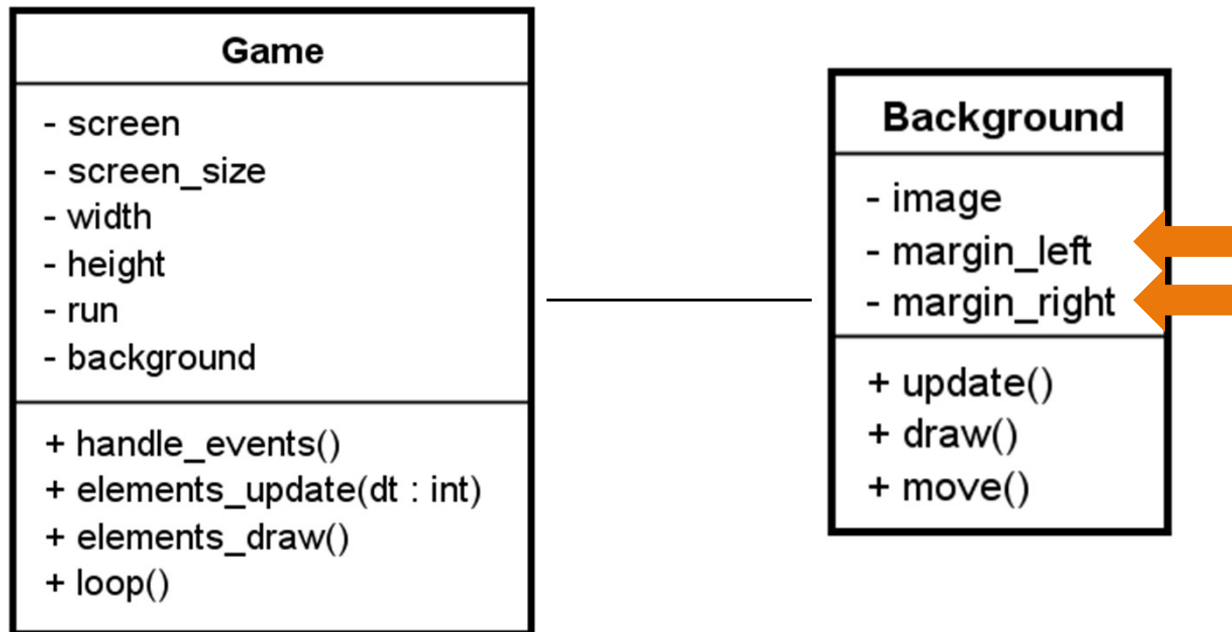




Atualizando o Diagrama de Classes



- ✿ Vamos agora inserir os atributos das margens esquerda e direita na classe **Background**





Classe Background: Margens

Definição das margens do background

Background
- image - margin_left - margin_right
+ update() + draw() + move()



```
9  class Background:
10      """
11      Esta classe define o Plano de Fundo do jogo
12      """
13      image = None
14      margin_left = None
15      margin_right = None
```



novos atributos

observe sempre a indentação do código



Classe Background: Margens

- Definição das margens do background na função `__init__()`

Background
- image - margin_left - margin_right
+ update() + draw() + move()



```
17 def __init__(self):
18
19     background_fig = pygame.image.load("Images/background.png")
20     background_fig.convert()
21     self.image = background_fig
22
23     margin_left_fig = pygame.image.load("Images/margin_1.png")
24     margin_left_fig.convert()
25     margin_left_fig = pygame.transform.scale(margin_left_fig, (60, 602))
26     self.margin_left = margin_left_fig
27
28     margin_right_fig = pygame.image.load("Images/margin_2.png")
29     margin_right_fig.convert()
30     margin_right_fig = pygame.transform.scale(margin_right_fig, (60, 602))
31     self.margin_right = margin_right_fig
32
33 # __init__()
```

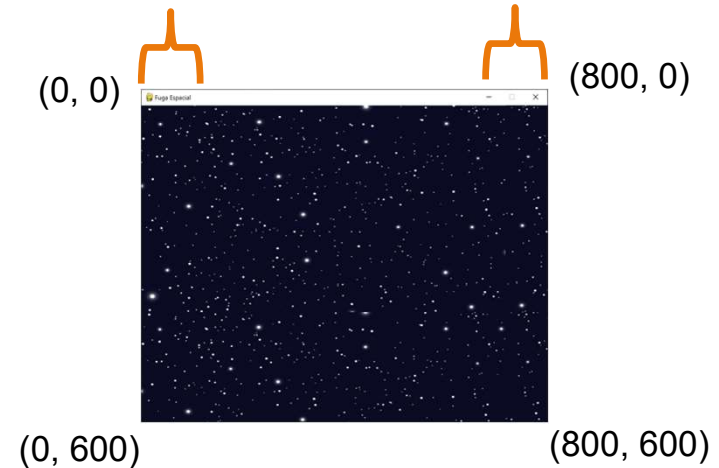
observe sempre a indentação do código



Classe Background: Margens



Definição das margens do background



Background
- image - margin_left - margin_right
+ update() + draw() + move()

```
35 def update(self, dt):
36     pass # Ainda não faz nada
37     # update()
38
39 def draw(self, screen):
40     screen.blit(self.image, (0, 0))
41     screen.blit(self.margin_left, (0, 0)) # 60 depois da primeira margem
42     screen.blit(self.margin_right, (740, 0)) # 60 antes da segunda margem
43     # draw()
44     # Background
```




Classe Background: Margens



Definição das margens do background

Background
- image - margin_left - margin_right
+ update() + draw() + move()



```
35     def update(self, dt):
36         pass # Ainda não faz nada
37         # update()
38
39     def draw(self, screen):
40         screen.blit(self.image, (0, 0))
41         screen.blit(self.margin_left, (0, 0)) # 60 depois da primeira margem
42         screen.blit(self.margin_right, (740, 0)) # 60 antes da segunda margem
43         # draw()
44     # Background
```

observe sempre a indentação do código



Executando o código



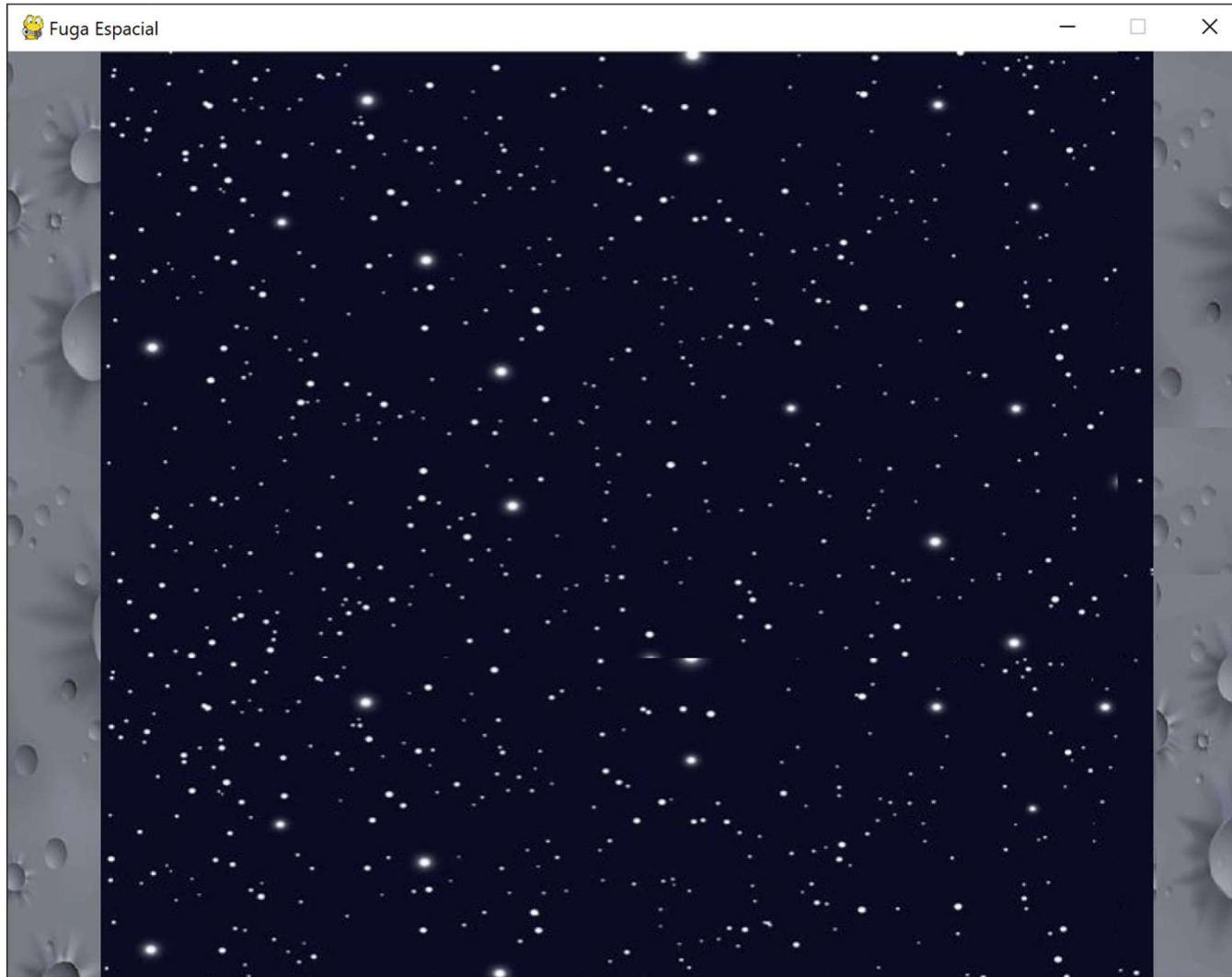
```
6
7 import pygame
8
9 class Background:
10     """
11     Esta classe define o Plano de Fundo
12     """
13     image = None
14
15     def __init__(self):
16
17         background_fig = pygame.image
18         background_fig.convert()
19         self.image = background_fig
```

Context menu for `main.py`:

- Show Context Actions (Alt+Enter)
- Paste (Ctrl+V)
- Copy / Paste Special >
- Column Selection Mode (Alt+Shift+Insert)
- Find Usages (Alt+F7)
- Refactor >
- Folding >
- Go To >
- Generate... (Alt+Insert)
- Run (Ctrl+Shift+F10)



Executando o código



**Tela do jogo com
margens no background
ou Plano de Fundo**

Movimentação do Plano de Fundo

Parte 2



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO





Movimento





Movimento





Movimento





Movimento



Temos a impressão que os **ETs** estão vindo em nossa direção, certo?

Essa movimentação é feita apenas com a **alteração das dimensões da imagem em diferentes slides!**

Guarde esta ideia, iremos usá-la logo mais.



É importante lembrar que, na programação de computadores, uma mesma atividade pode ser realizada de diferentes formas.

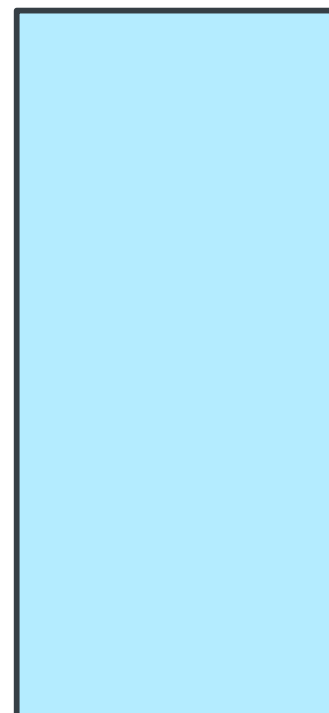
Aqui exploraremos uma delas.



Movimentação do plano de fundo



- ✈ Utiliza-se o descolamento de 'tiles' para a movimentação de objetos, como o background
- ✈ A ideia é criar a ilusão do movimento

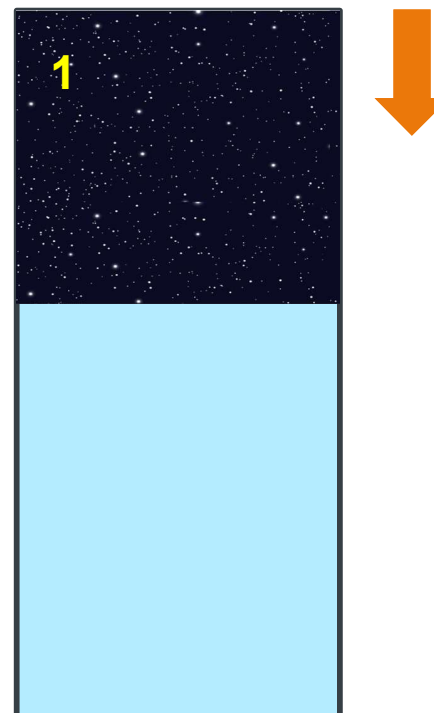




Movimentação do plano de fundo



- ✈ Utiliza-se o descolamento de 'tiles' para a movimentação de objetos, como o background
- ✈ A ideia é criar a ilusão do movimento

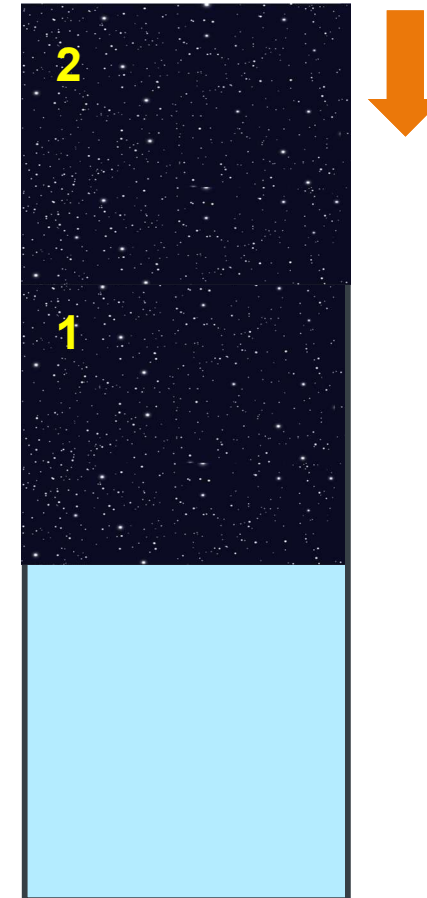




Movimentação do plano de fundo



- ✈ Utiliza-se o descolamento de 'tiles' para a movimentação de objetos, como o background
- ✈ A ideia é criar a ilusão do movimento

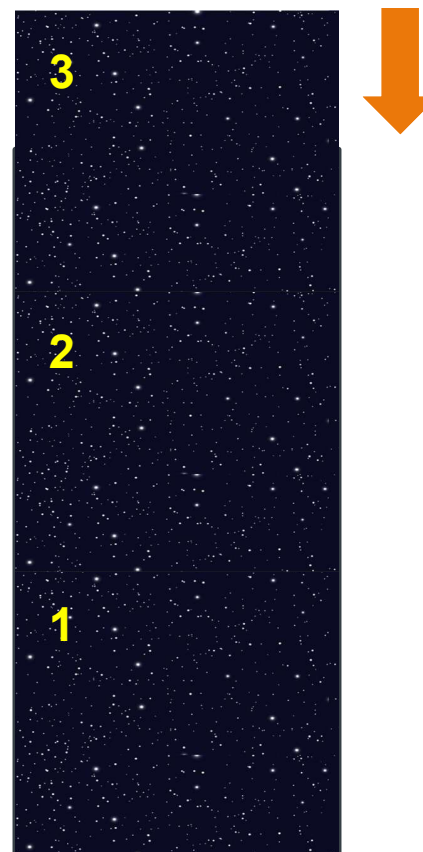




Movimentação do plano de fundo



- ✈ Utiliza-se o descolamento de 'tiles' para a movimentação de objetos, como o background
- ✈ A ideia é criar a ilusão do movimento

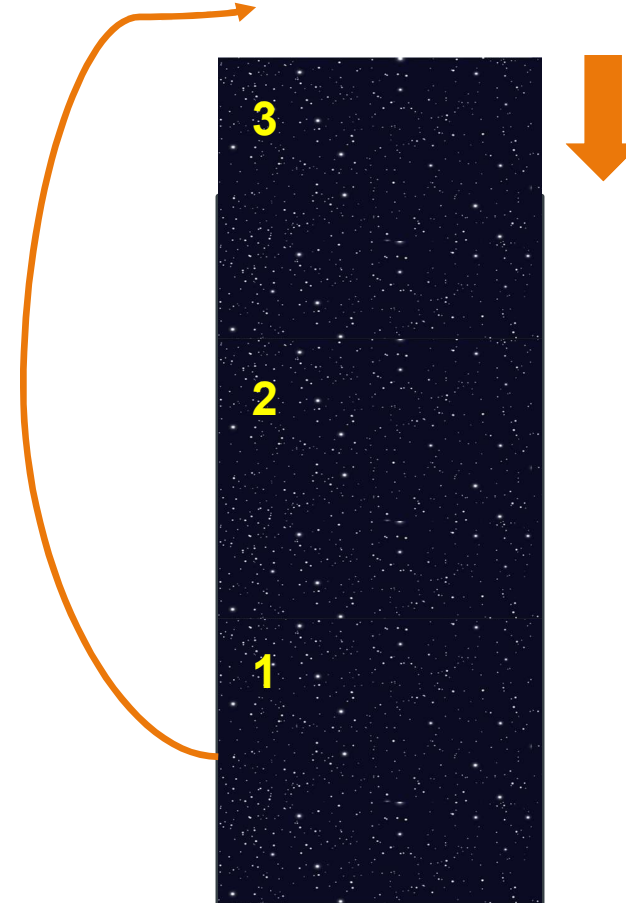




Movimentação do plano de fundo



- ✈ Utiliza-se o descolamento de 'tiles' para a movimentação de objetos, como o background
- ✈ A ideia é criar a ilusão do movimento

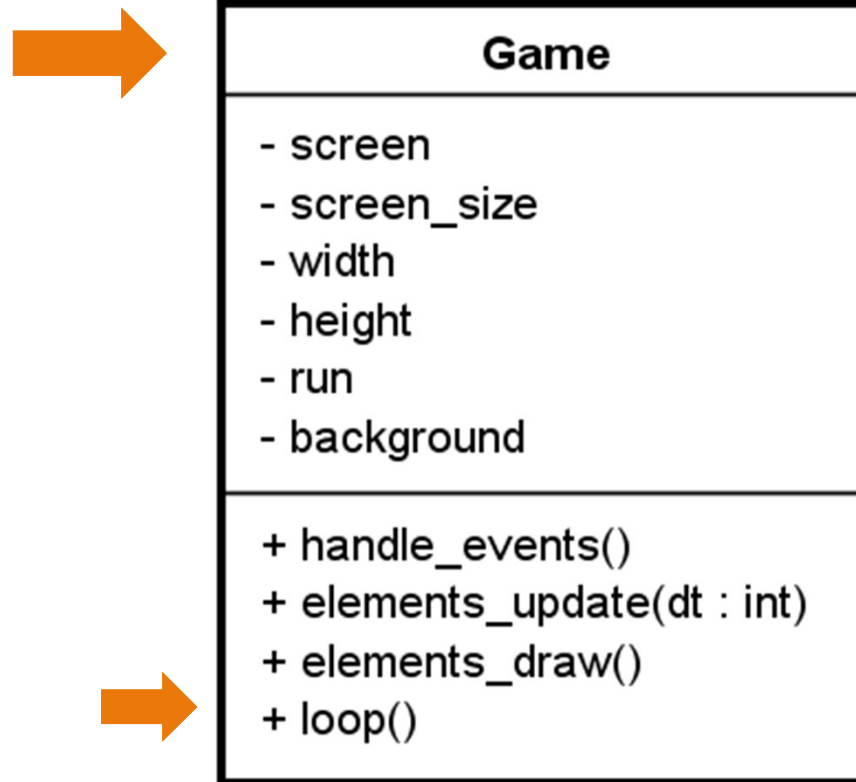




Movimentação do plano de fundo



- ✿ Na **classe Game**, iremos declarar alguns atributos na função **loop**.





Movimentação do plano de fundo



- ✈ Na **classe Game**, iremos declarar alguns atributos na função **loop**.

```
47 class Game:
48     screen = None
49     screen_size = None
50     width = 800
51     height = 600
52     run = True
53     background = None
54
55     def __init__(self, size, fullscreen: ...
56
57     # init()
58
59     def handle_events(self: ...
60     # handle_events()
61
62     def elements_update(self, dt): ...
63     # elements_update()
64
65     def elements_draw(self: ...
66     # elements_draw()
67
68     def loop(self):
69         """
70         Laço principal
71         """
```

Não digite os 3 pontinhos!

observe sempre a indentação do código



Movimentação do plano de fundo



- ✿ Declarando a variável para movimentação do plano de fundo ou background

```
121 def loop(self):  
122     """  
123     Laço principal  
124     """  
125     #variáveis para movimento de Plano de Fundo/Background  
126     velocidade_background = 10
```

observe sempre a indentação do código

nova variável



Movimentação do plano de fundo



- ✈ Ainda na classe Game, na função **loop** ...
- ✈ Vamos declarar as variáveis para o movimento da margem **esquerda (Left)**

```
121 def loop(self):  
122     """  
123     Laço principal  
124     """  
125     #variáveis para movimento de Plano de Fundo/Background  
126     velocidade_background = 10  
127  
128     # movimento da margem esquerda  
129     movL_x = 0  
130     movL_y = 0  
131
```

novas variáveis



Movimentação do plano de fundo



- ✈ Ainda na classe Game, na função **loop** ...
- ✈ Vamos declarar as variáveis para o movimento da margem **direita (Right)**

```
121     def loop(self):  
122         """  
123         Laço principal  
124         """  
125         #variáveis para movimento de Plano de Fundo/Background  
126         velocidade_background = 10  
127  
128         # movimento da margem esquerda  
129         movL_x = 0  
130         movL_y = 0  
131  
132         # movimento da margem direita  
133         movR_x = 740  
134         movR_y = 0  
135
```

novas variáveis



Movimentação do plano de fundo



- ✿ Declaradas as variáveis, iremos ajustar os tamanhos da imagem e das margens do plano de fundo na função `__init__()` da classe **Background**.

```
17 def __init__(self):
18
19     background_fig = pygame.image.load("Images/background.png")
20     background_fig.convert()
21     self.image = background_fig
22
23     margin_left_fig = pygame.image.load("Images/margin_1.png")
24     margin_left_fig.convert()
25     margin_left_fig = pygame.transform.scale(margin_left_fig, (60, 600))
26     self.margin_left = margin_left_fig
27
28     margin_right_fig = pygame.image.load("Images/margin_2.png")
29     margin_right_fig.convert()
30     margin_right_fig = pygame.transform.scale(margin_right_fig, (60, 600))
31     self.margin_right = margin_right_fig
32
33 # __init__()
```



Movimentação do plano de fundo



- ✿ Declaradas as variáveis, iremos ajustar os tamanhos da imagem e das margens do plano de fundo na função `__init__()` da classe **Background**.

```
17  def __init__(self):
18
19      background_fig = pygame.image.load("Images/background.png")
20      background_fig.convert()
21      background_fig = pygame.transform.scale(background_fig, (800, 602))
22      self.image = background_fig
```

linha inserida



Movimentação do plano de fundo



- Declaradas as variáveis, iremos ajustar os tamanhos da imagem e das margens do plano de fundo na função `__init__()` da classe **Background**.

```
17 def __init__(self):
18
19     background_fig = pygame.image.load("Images/background.png")
20     background_fig.convert()
21     background_fig = pygame.transform.scale(background_fig, (800, 602))
22     self.image = background_fig
23
24     margin_left_fig = pygame.image.load("Images/margin_1.png")
25     margin_left_fig.convert()
26     margin_left_fig = pygame.transform.scale(margin_left_fig, (60, 602))
27     self.margin_left = margin_left_fig
28
29     margin_right_fig = pygame.image.load("Images/margin_2.png")
30     margin_right_fig.convert()
31     margin_right_fig = pygame.transform.scale(margin_right_fig, (60, 602))
32     self.margin_right = margin_right_fig
```

mesmos valores para y



Movimentação do plano de fundo



- ✿ Ainda na classe **Background**, vamos definir as **posições** das imagens no plano de fundo.

Lembra das imagens apresentadas no início da aula? Faremos algo similar...

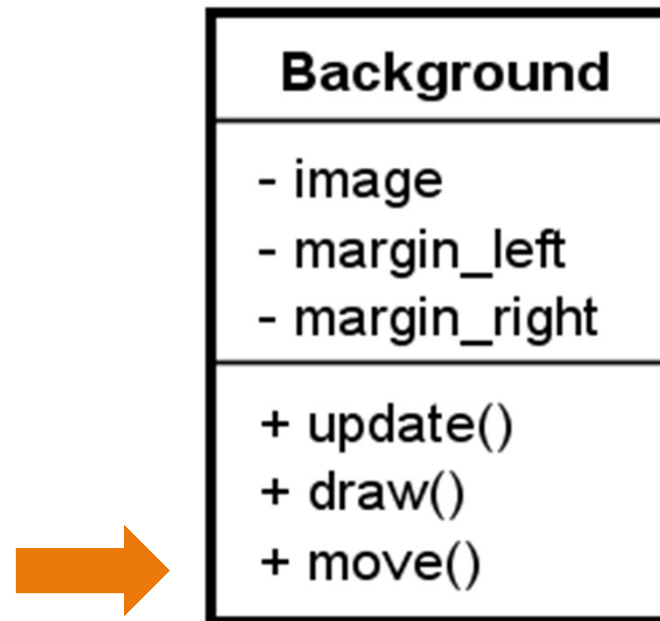
```
1  """
2  Jogo: Fuga Espacial
3  Descrição: Um grupo de diplomatas escapam de uma fortaleza estelar a bordo de uma nave danificada.
4  A nave precisa se desviar das ameaças e sobreviver até atingir a zona de segurança diplomática.
5  """
6
7  import pygame
8
9  class Background:
10     """
11     Esta classe define o Plano de Fundo do jogo
12     """
13     image = None
14     margin_left = None
15     margin_right = None
16
17     def __init__(self):
18
19         background_fig = pygame.image.load("Images/background.png")
20         background_fig.convert()
21         self.image = background_fig
```



Movimentação do plano de fundo



✈ Na classe Background, vamos inserir informações na função **move()**





Movimentação do plano de fundo



- ✿ Na classe Background, a função **move()** recebe como parâmetros
 - ✿ uma tela (screen)
 - ✿ a altura da tela (scr_height)
 - ✿ as coordenadas de movimento (movL_x, movL_y, movR_x, movR_y).

```
44  
45 # Define posições do Plano de Fundo para criar o movimento  
46 def move(self, screen, scr_height, movL_x, movL_y, movR_x, movR_y):
```



Movimentação do plano de fundo



Dentro da função **move()** criaremos o movimento para:

- 🚀 **Imagem do background;**
- 🚀 **Margem esquerda;**
- 🚀 **Margem direita.**

```
42 # Define posições das imagens do Plano de Fundo para criar o movimento
43 def move(self, screen, scr_height, movL_x, movL_y, movR_x, movR_y):
44
45     for i in range(0, 2):
46         screen.blit(self.image, (movL_x, movL_y - i * scr_height))
47         screen.blit(self.margin_left, (movL_x, movL_y - i * scr_height))
48         screen.blit(self.margin_right, (movR_x, movR_y - i * scr_height))
49
50 # move()
```




Movimentação do plano de fundo



- ❖ Obteremos a ideia de movimento do background colocando a imagem de fundo e as margens em diferentes posições na tela usando a função "blit".

```
44  
45 for i in range(0, 2):  
46     screen.blit(self.image, (movL_x, movL_y - i * scr_height))  
47     screen.blit(self.margin_left, (movL_x, movL_y - i * scr_height))  
48     screen.blit(self.margin_right, (movR_x, movR_y - i * scr_height))  
49
```

São 3 posições diferentes!

A ideia é que o background se movimente de cima para baixo.

Então, a coordenada x é constante e a coordenada y sofre variações.



Movimentação do plano de fundo



- ✿ Respira fundo que estamos quase lá.





Movimentação do plano de fundo



✈ Agora, voltaremos à Classe **Game**

```
99
100 class Game:
101     screen = None
102     screen_size = None
103     width = 800
104     height = 600
105     run = True
106     background = None
107
108     def __init__(self, size, fullscreen):...
109
110     # init()
111
112     def handle_events(self):...
113
114     # handle_events()
115
116     def elements_update(self, dt):...
117
118     # elements_update()
119
120     def elements_draw(self):...
```

Não digite os 3 pontinhos!



Movimentação do plano de fundo



✈ Na classe Game, vamos para a função **loop()**.

Game
<ul style="list-style-type: none">- screen- screen_size- width- height- run- background
<ul style="list-style-type: none">+ handle_events()+ elements_update(dt : int)+ elements_draw()+ loop()

```
121 def loop(self):
122     """
123     """
124
125     #variáveis para movimento de Plano de Fundo/Background
126     velocidade_background = 10
127
128     # movimento da margem esquerda
129     movL_x = 0
130     movL_y = 0
131
132     # movimento da margem direita
133     movR_x = 740
134     movR_y = 0
135
136     # Criar o Plano de fundo
137     self.background = Background()
138
139     ...
140
141     clock = pygame.time.Clock()
142     dt = 16
143
144     # assim iniciamos o loop principal do programa
145     while self.run:
146         clock.tick(1000 / dt)
147
148         # Handle Input Events
149         self.handle_events()
150
```

Laço principal



Movimentação do plano de fundo



- ✈ Dentro do **laço principal**, vamos chamar a função **move()** da classe **Background**.

```
118 # Laço principal do programa
119 while self.run:
120     clock.tick(1000 / dt)
121
122     """ ... """
123
124     # adiciona movimento ao background
125     self.background.move(self.screen, self.height, movL_x, movL_y, movR_x, movR_y)
126     movL_y = movL_y + velocidade_background
127     movR_y = movR_y + velocidade_background
128
129
130
131
132
133
134
```

Laço principal



Movimentação do plano de fundo



- ✈ Dentro do **laço principal**, vamos chamar a função **move()** da classe **Background**.

```
130 # adiciona movimento ao background
131 self.background.move(self.screen, self.height, movL_x, movL_y, movR_x, movR_y)
132 movL_y = movL_y + velocidade_background
133 movR_y = movR_y + velocidade_background
134
```

observe sempre a indentação do código



Movimentação do plano de fundo



- ✿ Ainda no laço principal, iremos inserir uma condição, logo abaixo do que acabamos de fazer.

```
130     # adiciona movimento ao background
131     self.background.move(self.screen, self.height, movL_x, movL_y, movR_x, movR_y)
132     movL_y = movL_y + velocidade_background
133     movR_y = movR_y + velocidade_background
134
135     #se a imagem ultrapassar a extremidade da tela, move de volta
136     if movL_y > 600 and movR_y > 600:
137         movL_y -= 600
138         movR_y -= 600
139
```





Movimentação do plano de fundo



- ✿ Ainda no loop principal, iremos inserir uma condição, logo abaixo do que acabamos de fazer.

```
135 #se a imagem ultrapassar a extremidade da tela, move de volta
136 if movL_y > 600 and movR_y > 600:
137     movL_y -= 600
138     movR_y -= 600
139
```

observe sempre a indentação do código

Esse trecho de código é uma lógica, dentro de um **loop infinito**, que garante que a imagem de fundo esteja sempre em movimento. Caso ela ultrapasse a extremidade da tela, ela é movida de volta para o topo.

Isso garante um efeito de movimento contínuo e suave do plano de fundo durante o jogo.



VAMOS RODAR O JOGO AGORA?!



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO





Executando o código



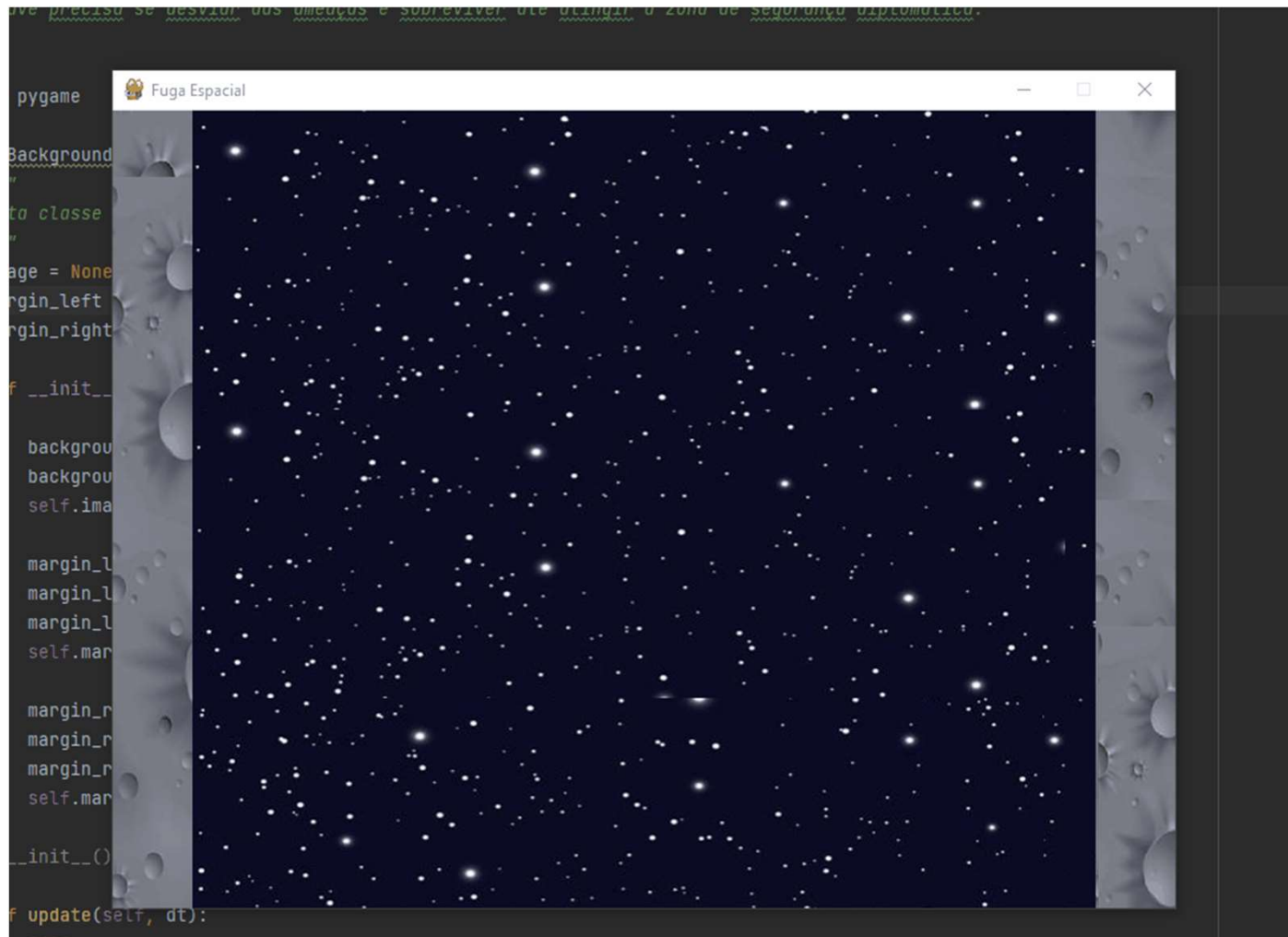
```
6
7 import pygame
8
9 class Background:
10     """
11     Esta classe define o Plano de Fundo
12     """
13     image = None
14
15     def __init__(self):
16
17         background_fig = pygame.image
18         background_fig.convert()
19         self.image = background_fig
```

Context menu options:

- Show Context Actions (Alt+Enter)
- Paste (Ctrl+V)
- Copy / Paste Special (>)
- Column Selection Mode (Alt+Shift+Insert)
- Find Usages (Alt+F7)
- Refactor (>)
- Folding (>)
- Go To (>)
- Generate... (Alt+Insert)
- Run main.py (Ctrl+Shift+F10)



Executando o código



Atividades práticas

Mude o código para melhor entendê-lo

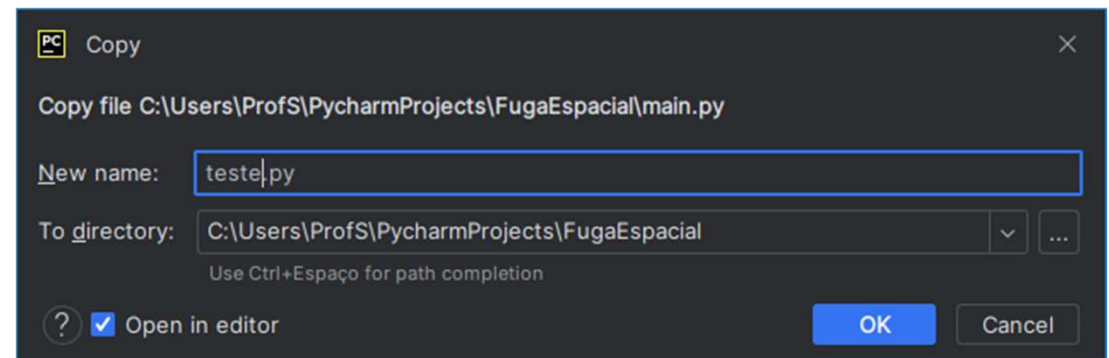
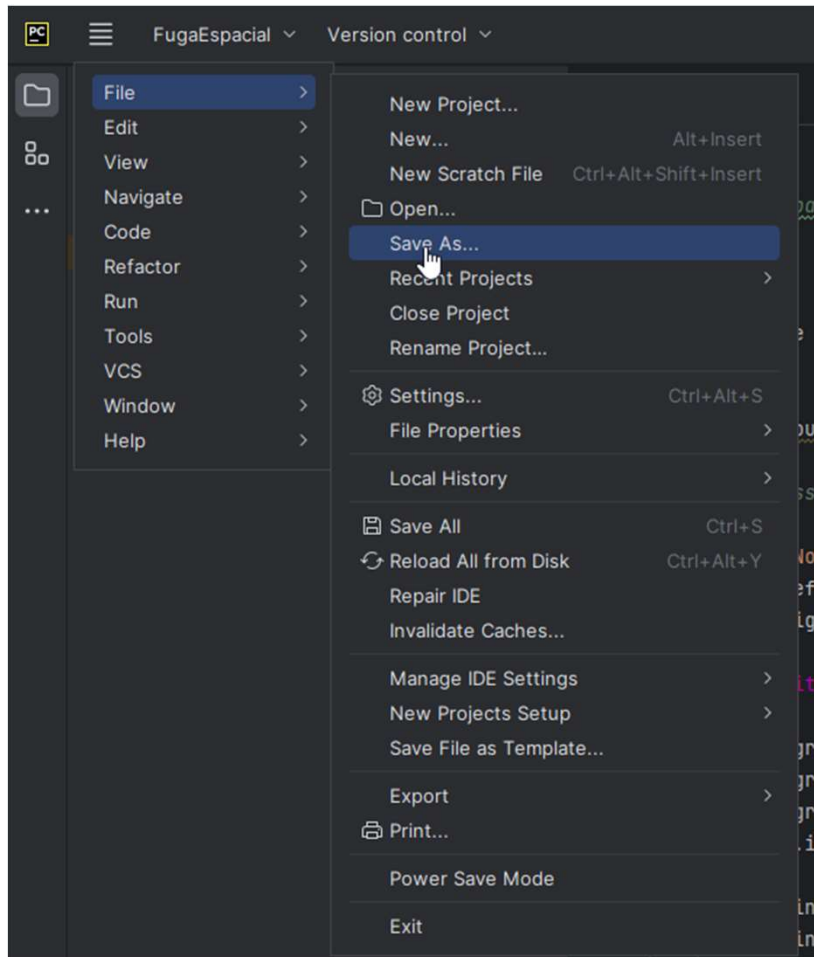




ANTES DE COMEÇAR



Salve o código como teste.py para não perder o código do jogo já desenvolvido





Entendendo o Efeito da Velocidade de Movimento



Modifique a variável ``velocidade_background`` no método ``loop`` da classe ``Game``. Experimente valores mais altos e mais baixos.

Observe e anote o que acontece com o movimento do plano de fundo quando você aumenta ou diminui a velocidade.

Pergunta: Como a mudança na ``velocidade_background`` afeta o movimento do plano de fundo? Qual é o impacto no desempenho geral do jogo quando a ``velocidade_background`` é muito alta ou muito baixa?



Entendendo a Posição Inicial da Margem



Altere os **valores iniciais** de ``movL_y`` e ``movR_y`` no método ``loop`` da classe ``Game``. Experimente valores positivos e negativos.

Observe e anote o que acontece com a posição inicial das margens quando você muda esses valores.

Pergunta: Como a mudança nos valores iniciais de ``movL_y`` e ``movR_y`` afeta a posição inicial das margens? Qual é o impacto no desempenho geral do jogo quando esses valores são muito altos ou muito baixos?



Entendendo o Efeito do Limite de Movimento



Modifique a condição dentro do `if` no método `loop` da classe `Game`, onde os valores de `movL_y` e `movR_y` são reiniciados ao atingir um determinado limite (originalmente 600). Experimente aumentar e diminuir esse limite.

Observe e anote o que acontece com o movimento do plano de fundo e as margens quando você muda esse limite.

Pergunta: Como a mudança nesse limite afeta o movimento do plano de fundo e as margens? Qual é o impacto no desempenho geral do jogo quando o limite é muito alto ou muito baixo?



Fonte: Imagem de catalyststuff no Freepik

Obrigada

Até a próxima Aula!



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

