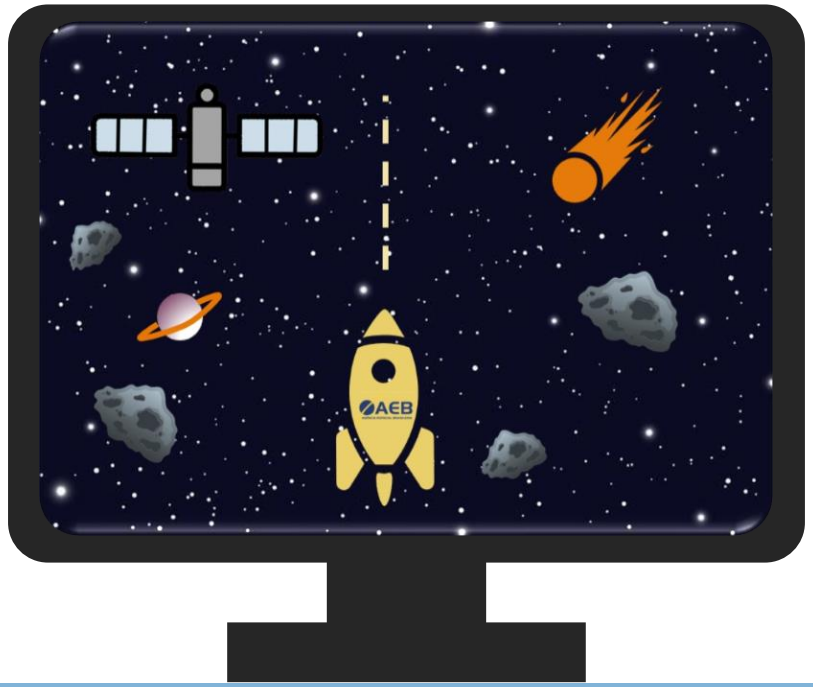


Minicurso de Programação de Jogos em Python

Aula 2
Criação de jogos em Python com Pygame

Professora: Camyla Moreno - Aluna de Pós-Graduação do ITA

Acesse

aebescolavirtual.aeb.gov.br/

PROGRAMA ESPACIAL BRASILEIRO

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Tópicos

- Conceito de Programação Orientada a Objetos – POO
- Introdução ao PyGame
- O básico do jogo
- Criando Plano de Fundo ou Background

Conceitos da Programação Orientada a Objetos

PROGRAMA ESPACIAL BRASILEIRO

AEB escola

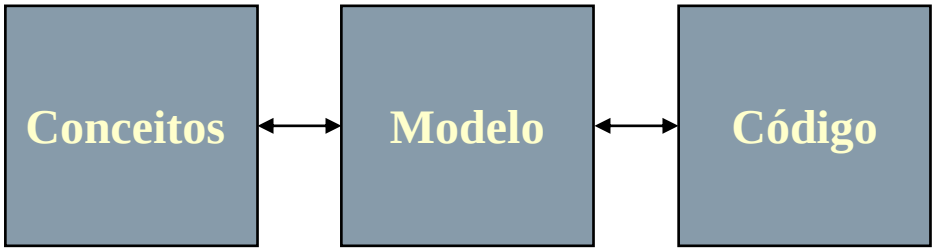
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO

Conceitos da Programação Orientada a Objetos

Batista e Moraes, 2013

Classe e objeto

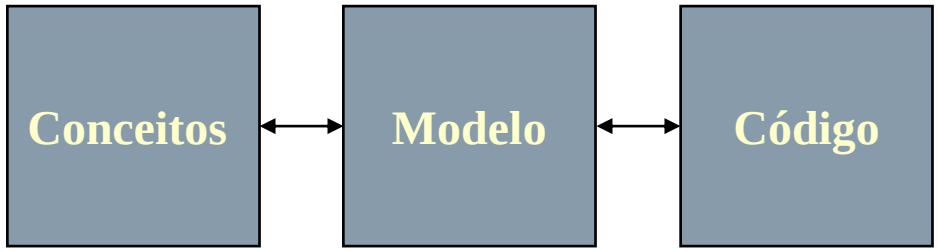


Objeto

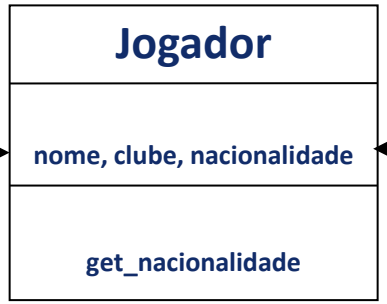
Jogador

- O objeto **Jogador** possui
  - Atributos: nome, clube, nacionalidade
  - Operações: get_nacionalidade

Classe e objeto

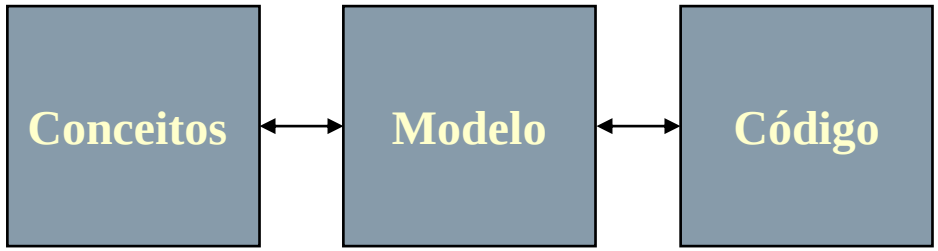


Jogador

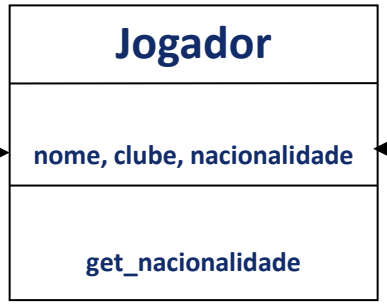



```
class Jogador:

  def __init__(self, nome, clube, nacionalidade):

    self.nome = nome
    self.clube = clube
    self.nacionalidade = nacionalidade

  def get_nacionalidade(self):
    return self.nacionalidade
```

Classe e objeto



Jogador




```
class Jogador:

  def __init__(self, nome, clube, nacionalidade):

    self.nome = nome
    self.clube = clube
    self.nacionalidade = nacionalidade

  def get_nacionalidade(self):
    return self.nacionalidade
```

Classe e objeto

Jogador

Jogador
nome, clube, nacionalidade
get_nacionalidade

Classe

nome: Cristiano Ronaldo clube: Al-Nassr nacionalidade: portuguesa
nome: Lionel Messi clube: PSG nacionalidade: argentina
nome: Marta Silva clube: Orlando Pride nacionalidade: brasileira

Objetos

Classe e objeto

```python
# Definir a classe Jogador
class Jogador:
    def __init__(self, nome, clube, nacionalidade):
        self.nome = nome
        self.clube = clube
        self.nacionalidade = nacionalidade

    def get_nacionalidade(self):
        return self.nacionalidade
```

```python
# Criar o objeto Jogador com os valores fornecidos
jogador = Jogador("Marta Silva", "Orlando Pride", "brasileira")
```

```python
print(jogador.nome)           # Saída: Marta Silva
print(jogador.clube)          # Saída: Orlando Pride
print(jogador.get_nacionalidade())  # Saída: brasileira
```

```
Marta Silva
Orlando Pride
brasileira
```

Notação para representar Classe

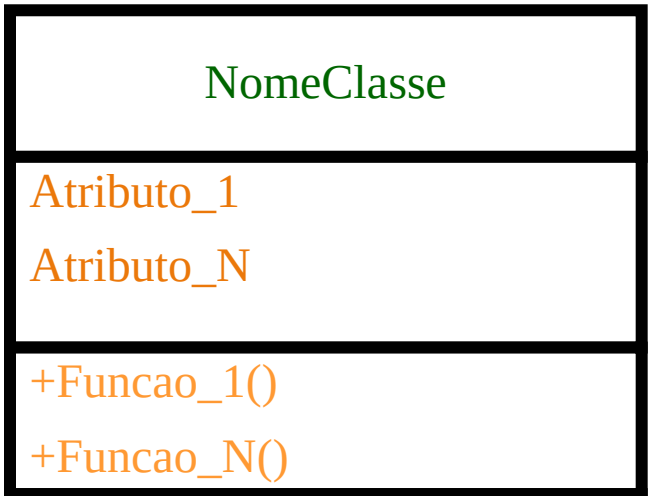


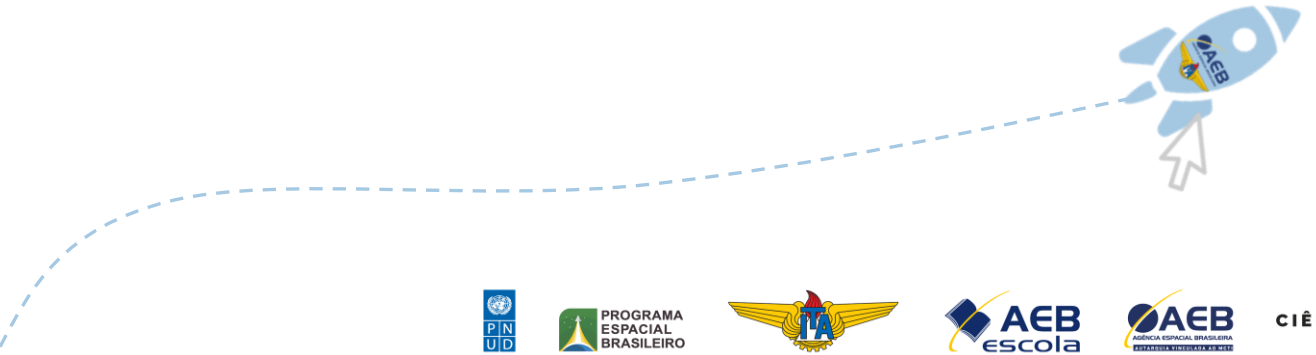
Diagrama de Classes

Classes podem ter

- Nome
- Atributos
- Funções

Apenas o nome é obrigatório

Jogos em Python com PyGame



PROGRAMA ESPACIAL BRASILEIRO

AEB escola

AEB

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL

BRASIL

UNIÃO E RECONSTRUÇÃO

Jogos em Python com PyGame

- A linguagem de programação Python é utilizada nas mais diversas áreas, como IA e Análise de Dados
- Neste minicurso utilizaremos
  - IDE PyCharm
  - O módulo PyGame para a construção de jogos digitais
- Conceitos e o uso do PyGame serão abordados no decorrer do minicurso

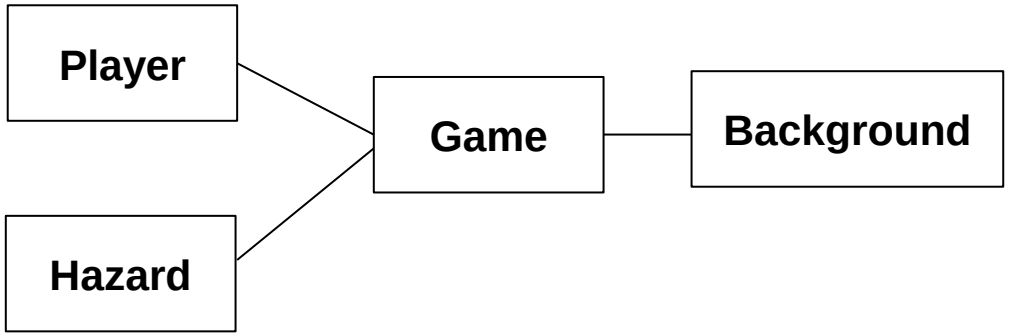
Jogos em Python com PyGame

- Em geral, um jogo em Python possui os elementos:
  - Laço principal
  - Plano de Fundo ou Background
  - Som
  - Jogador
  - Inimigos

Jogos em Python com PyGame

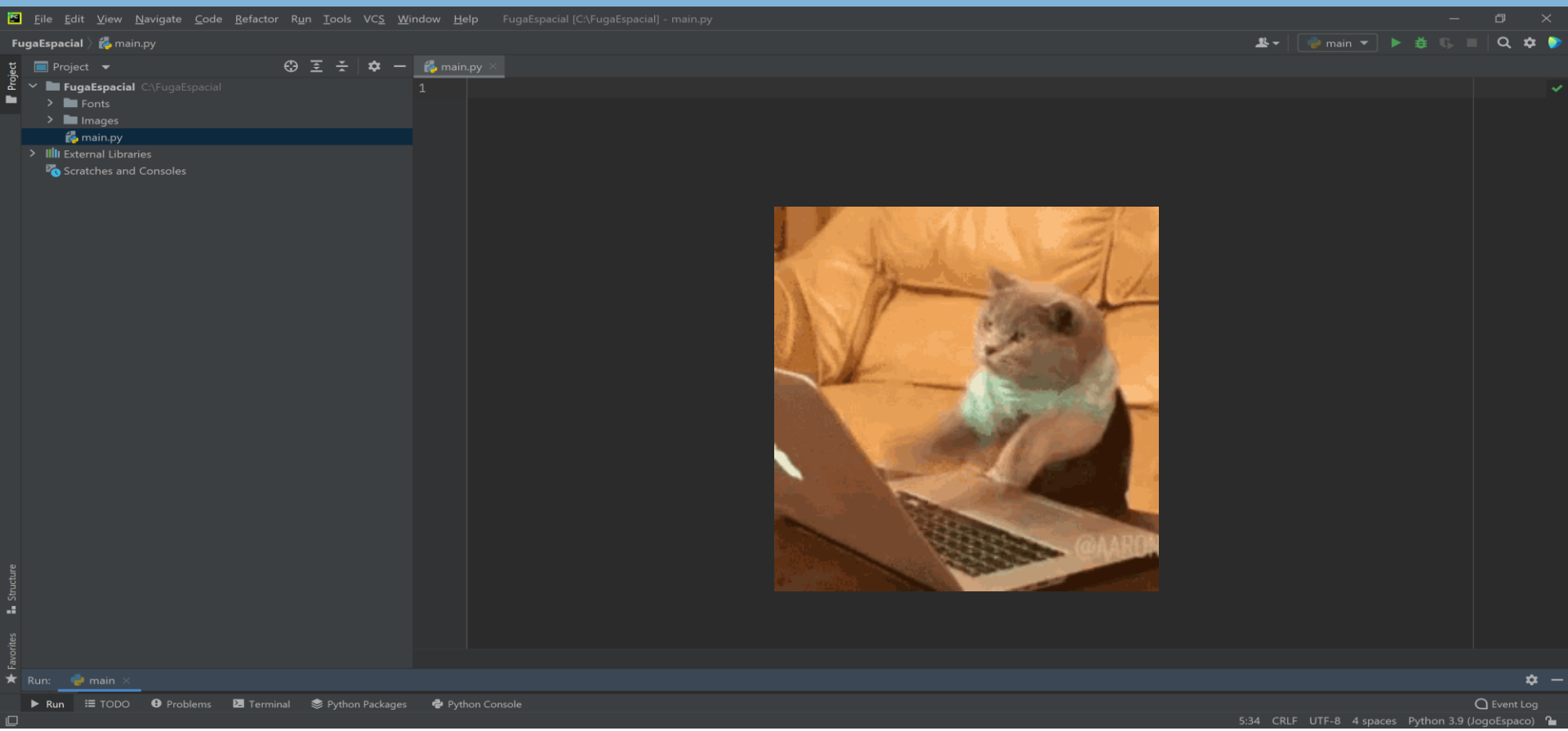
- Considerando a Programação Orientada a Objetos, vamos ter como **classes** do jogo **Fuga Espacial**
  - Game
  - Background
  - Player
  - Hazard

Diagrama de Classes



- Para fins didáticos, o Diagrama de Classes do jogo Fuga Espacial será complementado no decorrer do minicurso.

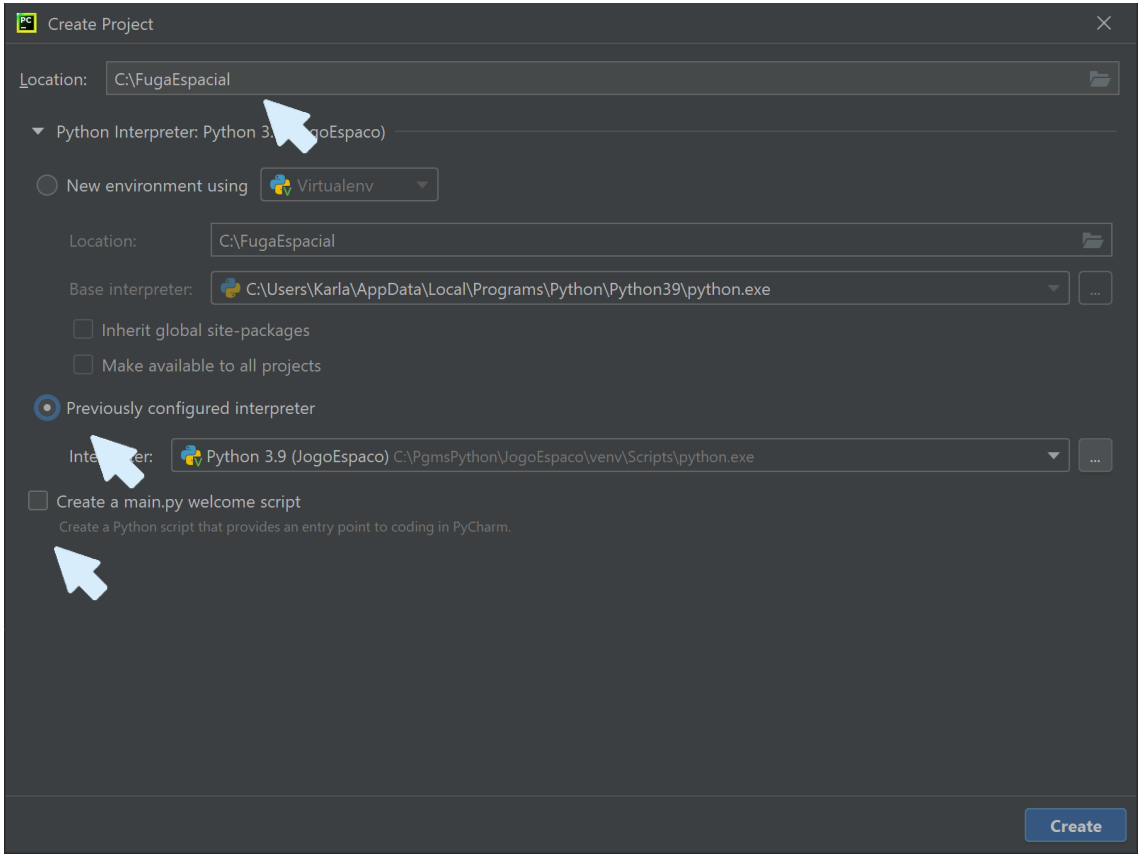
Vamos começar nosso código!



- Nesta Aula serão abordadas as classes Game e Background

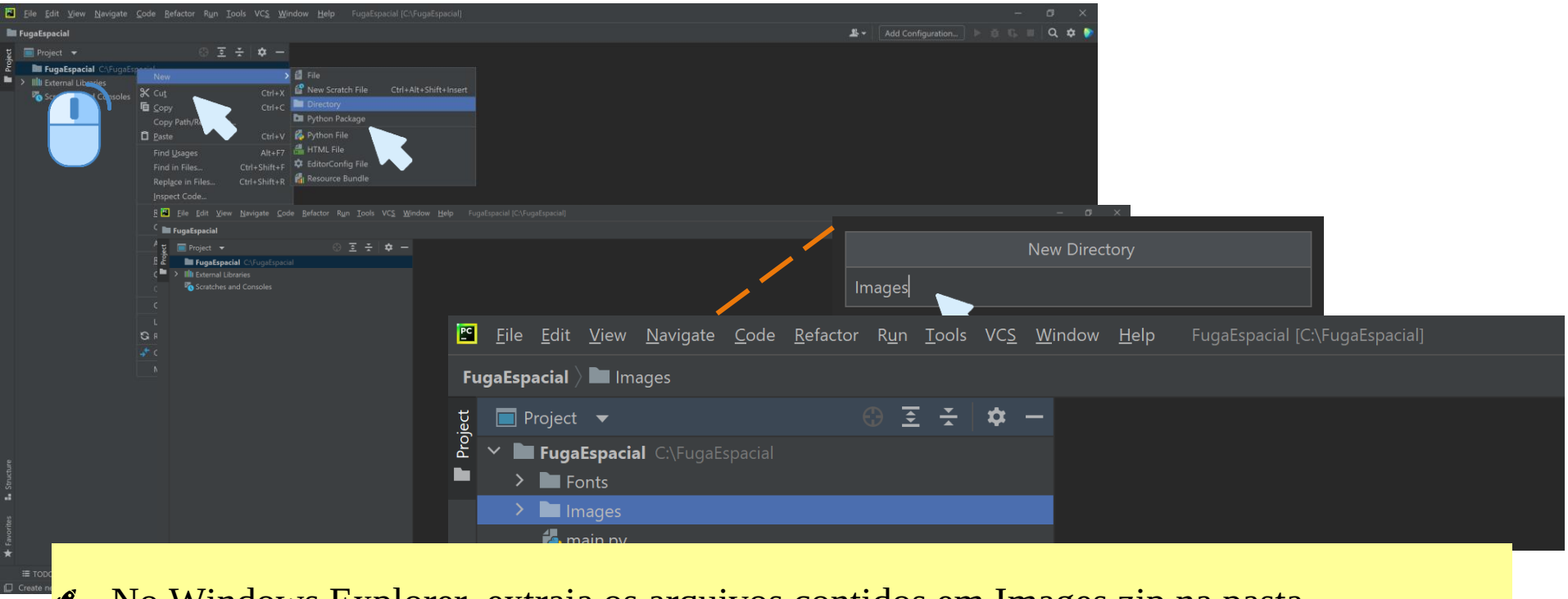
Criando o Projeto

- Criação do Projeto FugaEspacial em C:\



Criando pasta Images

- Criação da pasta (diretório) para as imagens do jogo: Images

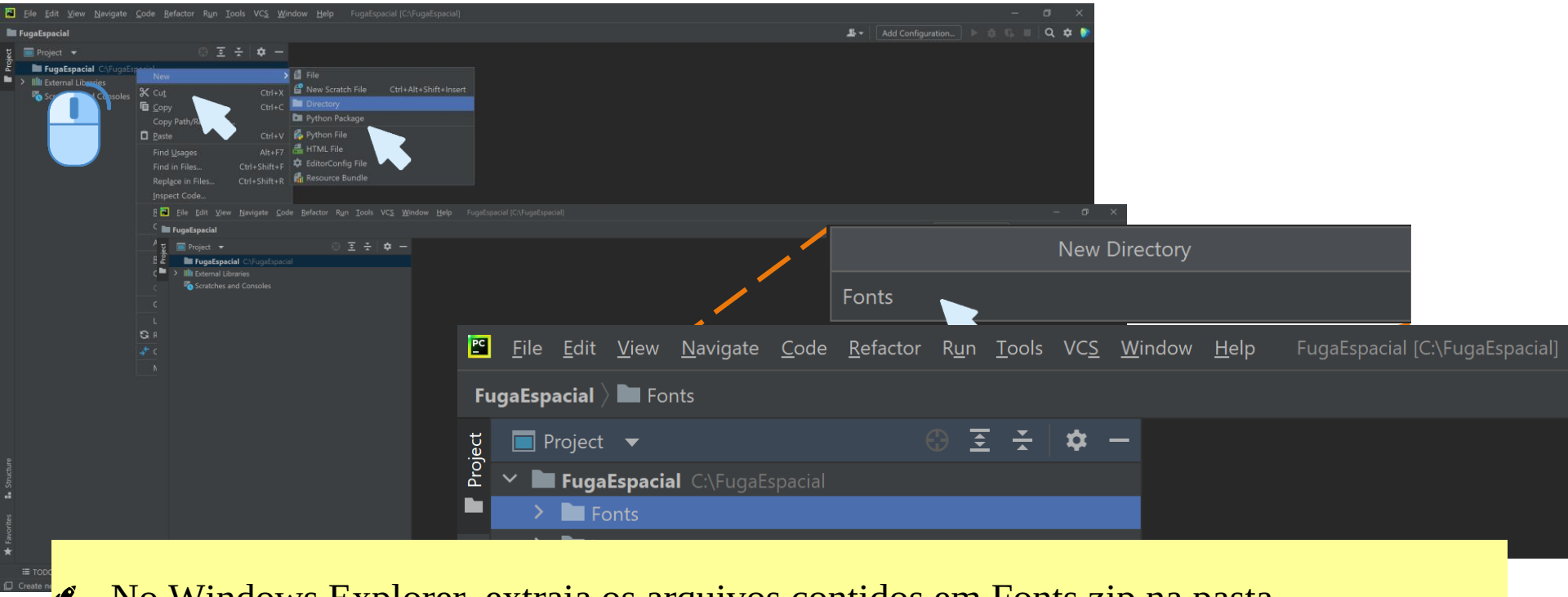


- No Windows Explorer, extraia os arquivos contidos em Images.zip na pasta

C:\FugaEspacial\Images

Criando pasta Fonts

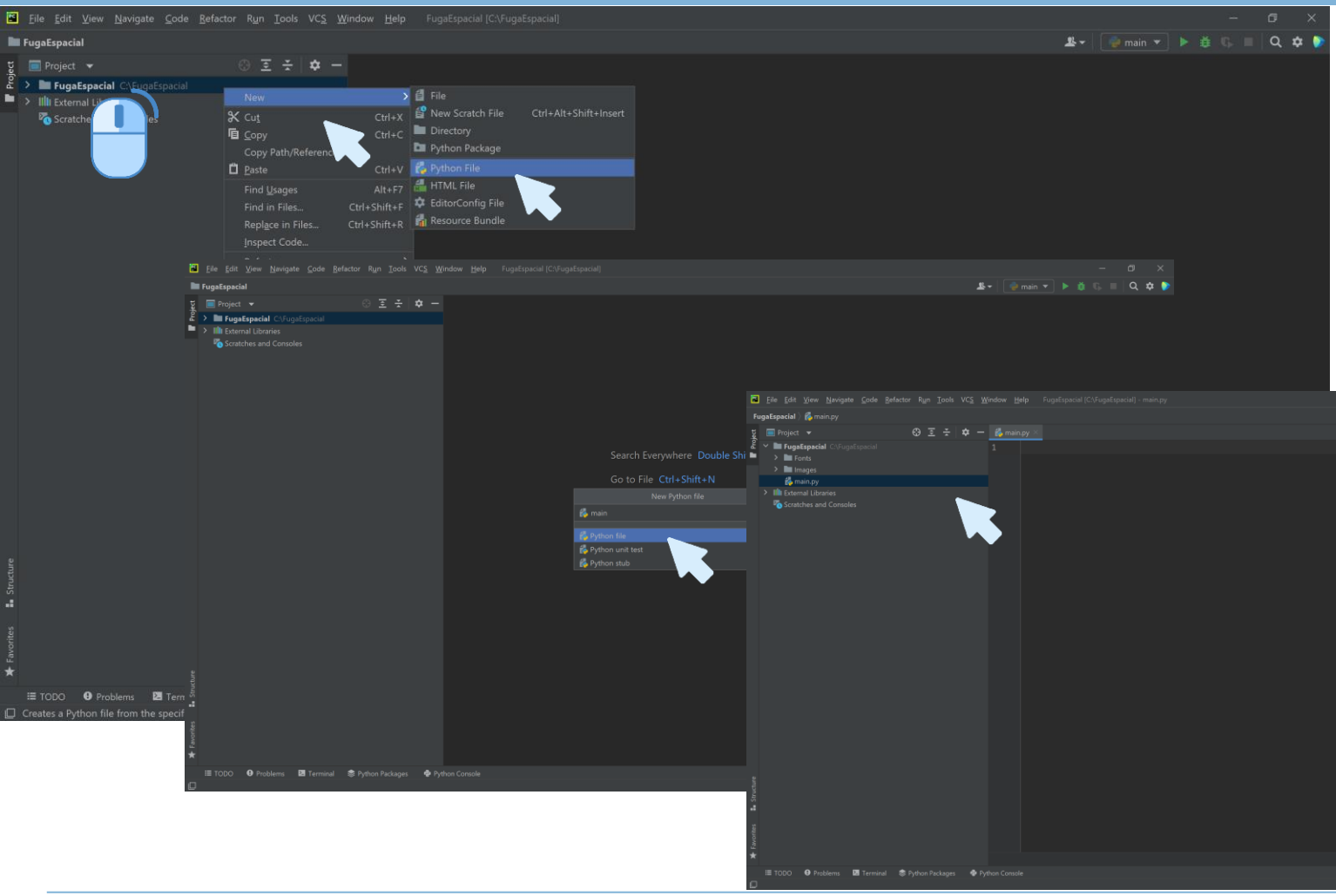
- Criação da pasta (diretório) para as fontes do jogo: Fonts



- No Windows Explorer, extraia os arquivos contidos em Fonts.zip na pasta

C:\FugaEspacial\Fonts

Criando o arquivo .py



Vamos iniciar nosso código

```
"""
Jogo: Fuga Espacial
Descrição: Um grupo de diplomatas escapam de uma fortaleza estelar a bordo de uma nave danificada.
   A nave precisa se desviar das ameaças e sobreviver até atingir a zona de segurança diplomática.
"""

import pygame

# comentar o programa

# importar o pygame
```

Codificando o início do jogo

```
9   # Inicia o jogo: Cria o objeto game e chama o loop básico
10  game = Game("resolution", "fullscreen") # instanciar o objeto jogo
11  game.loop() # iniciar o jogo
```

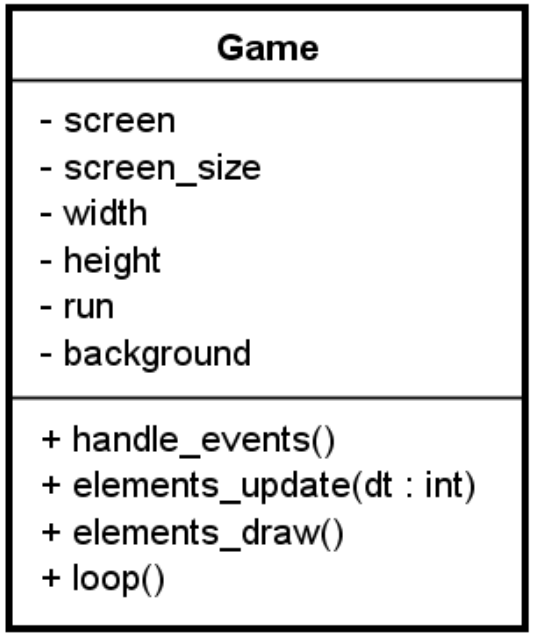
- Este código será inicialmente executado

Atualizando o Diagrama de Classes

- O objeto game foi instanciado da classe Game.

```
10      game = Game("resolution", "fullscreen")
```

- Os atributos e funções inicialmente definidas na classe são mostrados no Diagrama de Classes.
- A codificação da classe Game será feita a seguir.



Classe Game

- Vamos inserir a classe Game **antes** dos comandos iniciais do programa

Game
- screen - screen_size - width - height - run - background
+ handle_events() + elements_update(dt : int) + elements_draw() + loop()

```
9   class Game:
10      screen = None
11      screen_size = None
12      width = 800
13      height = 600
14      run = True
15      background = None
16
17      def __init__(self, size, fullscreen):...
30
31      # init()
32
33      def handle_events(self):...
40      # handle_events()
41
42      def elements_update(self, dt):
43          self.background.update(dt)
44      # elements_update()
45
46      def elements_draw(self):
47          self.background.draw(self.screen)
48      # elements_draw()
49
50      def loop(self):...
79      # loop()
80  # Game
```

inicializar atributos

operações

Classe Game

```
17      def __init__(self, size, fullscreen):
18
19          """
20          Função que inicializa o pygame, define a resolução da tela,
21          caption e desabilita o mouse.
22          """
23          pygame.init()
24
25          self.screen = pygame.display.set_mode((self.width, self.height))  # tamanho da tela
26          self.screen_size = self.screen.get_size()
27
28          pygame.mouse.set_visible(0)
29          pygame.display.set_caption('Fuga Espacial')
30
31      # init()
```

inicializar o pygame

definir o tamanho da tela do jogo

desabilitar o mouse

definir caption da janela do jogo

- Esta função é executada sempre que um objeto Game é instanciado

Classe Game

- Funções:

```
33      def handle_events(self):
34          """
35          Trata o evento e toma a ação necessária.
36          """
37          for event in pygame.event.get():
38              if event.type == pygame.QUIT:
39                  self.run = False
40      # handle_events()
41
42      def elements_update(self, dt):
43          self.background.update(dt)
44      # elements_update()
45
46      def elements_draw(self):
47          self.background.draw(self.screen)
48      # elements_draw()
```

tratar a saída do jogo

atualizar elementos

desenhar elementos

Classe Game

- Função loop()
  - Contém o laço principal, que é o código responsável em manter o jogo em execução

```
50      def loop(self):
51          """
52          Laço principal
53          """
54
55          # Criar o Plano de fundo
56          self.background = Background()
57
58          # Inicializas o relogio e o dt que vai limitar o valor de FPS (frames por segundo) do jogo
59          clock = pygame.time.Clock()
60          dt = 16
```

 # Criar objeto Background

 # inicializar relógio e dt

Classe Game

- Função loop():

```
62          # Início do loop principal do programa
63          while self.run:
64              clock.tick(1000 / dt)  # número máximo de FPS
65
66              # Handle Input Events
67              self.handle_events()   # trata eventos
68
69              # Atualiza Elementos
70              self.elements_update(dt)   # atualiza elementos
71
72              # Desenha o background buffer
73              self.elements_draw()
74                                     # desenha elementos
75              # Atualiza a tela
76              pygame.display.update()
77              clock.tick(2000)       # atualiza a tela
78          # while self.run
79      # loop()
80  # Game
```

Criando Plano de Fundo

PROGRAMA ESPACIAL BRASILEIRO

AEB escola

AEB

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

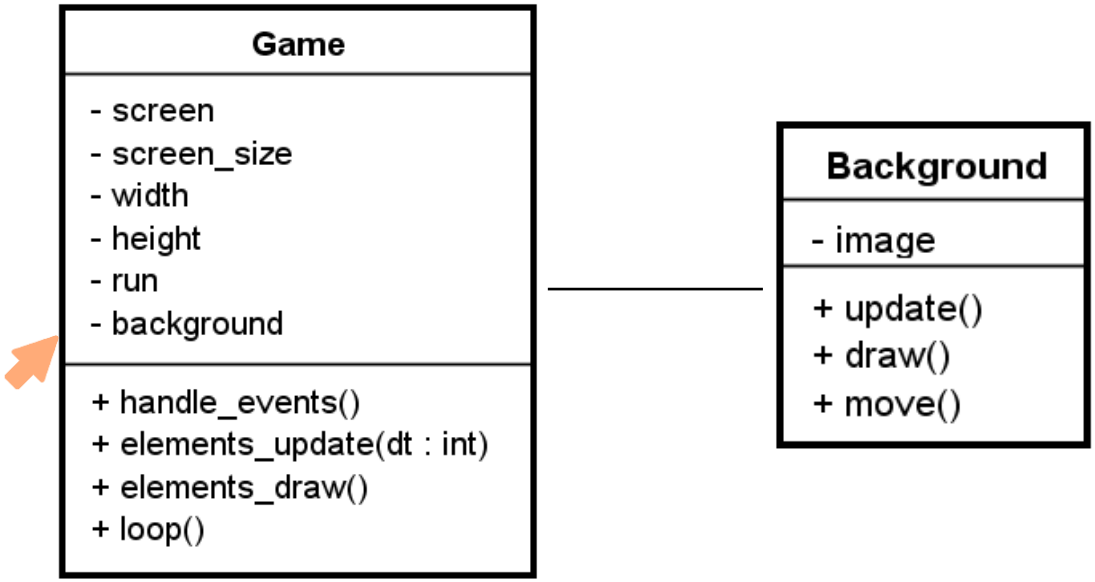
GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO

Atualizando o Diagrama de Classes

- Na função loop da classe Game o objeto background é instanciado

```
55          # Criar o Plano de fundo
56          self.background = Background()
```

- Vamos agora definir a classe Background



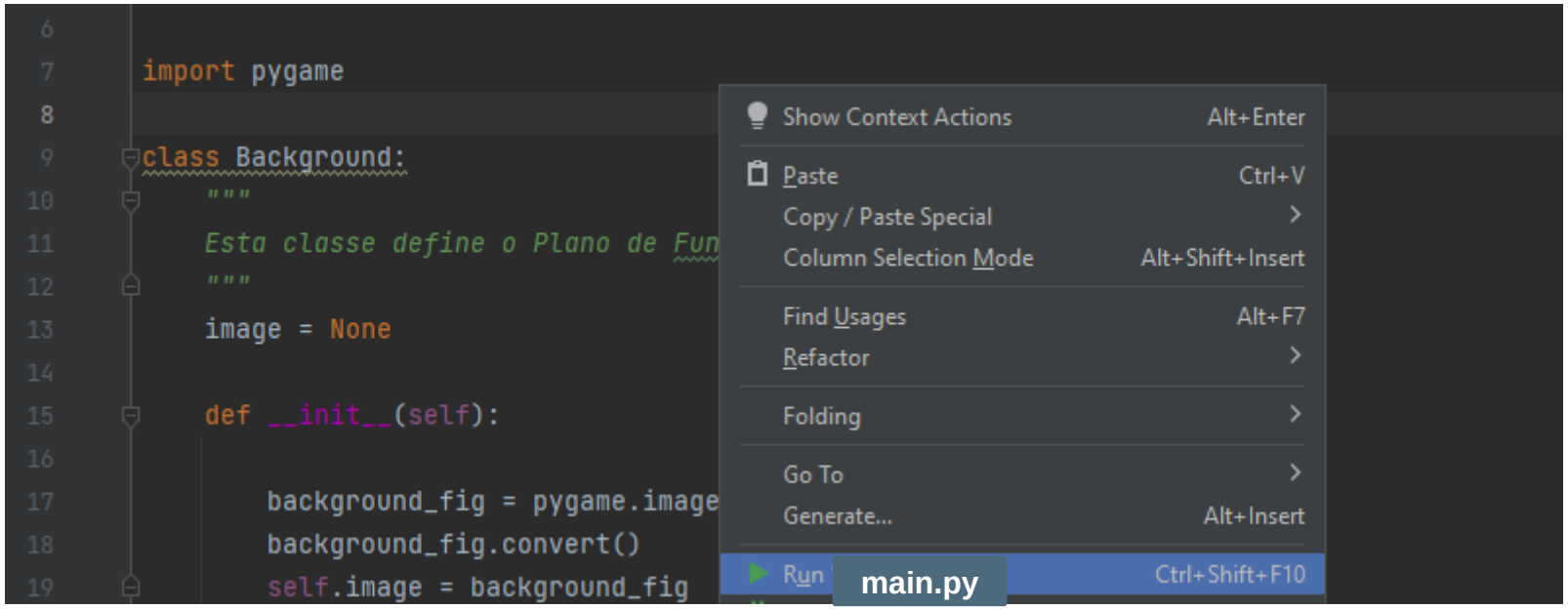
Classe Background

- Vamos inserir a classe Background **antes** da classe Game

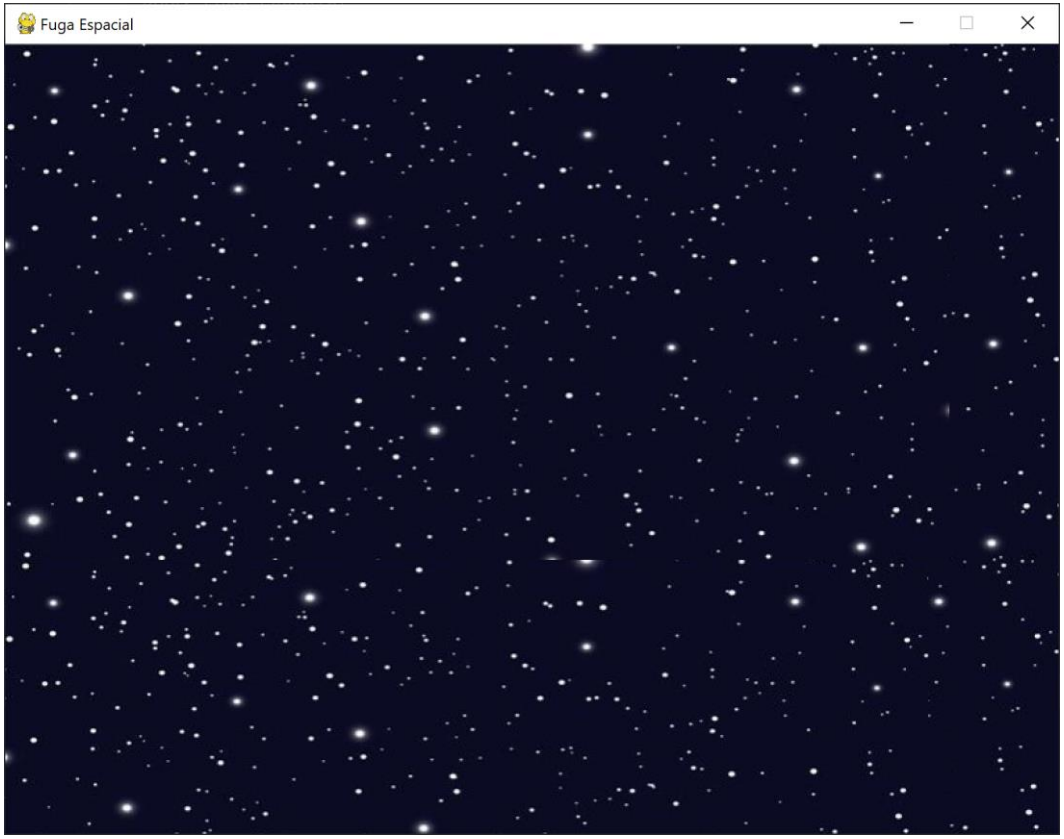
Background
- image
+ update() + draw() + move()

```python
class Background:
    """
    Esta classe define o Plano de Fundo do jogo
    """
    image = None                    # atributo

    def __init__(self):

        background_fig = pygame.image.load("Images/background.png")
        background_fig.convert()
        self.image = background_fig # atribui imagem para o background
    # __init__()

    def update(self, dt):
        pass  # Ainda não faz nada
    # update()

    def draw(self, screen):
        screen.blit(self.image, (0, 0))
    # draw()                        # imagem do background para tela

# Background
```

Executando o código



Executando o código



Tela do jogo com o background

Não esqueça de ...

- Assistir aos vídeos complementares
- Baixar os arquivos indicados na plataforma

Fonte: Imagem de catalyststuff no Freepik

Obrigada

Até a próxima Aula!

PNUD

PROGRAMA ESPACIAL BRASILEIRO

ITA

AEB escola

AEB

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Jogos em Python com PyGame

- Tela do Jogo: é importante entender como trabalhar com o monitor em um jogo

Sistema de coordenada cartesiana no monitor do computador. Fonte: Swelgart,

Eventos-padrão

Evento	Propósito	Parâmetros
QUIT	O usuário clicou no botão de fechamento.	none
ACTIVEEVENT	O Pygame foi ativado ou oculto.	gain, state
KEYDOWN	Uma tecla foi pressionada.	unicode, key, mod
KEYUP	Uma tecla foi solta.	key, mod
MOUSEMOTION	O mouse foi movido.	pos, rel, buttons
MOUSEBUTTONDOWN	Um botão do mouse foi pressionado.	pos, button
MOUSEBUTTONUP	Um botão do mouse foi solto.	pos, button
JOYAXISMOTION	O joystick ou o pad foi movido.	joy, axis, value
JOYBALLMOTION	O trackball do joystick foi movido.	joy, ball, rel
JOYHATMOTION	O joystick hat foi movido.	joy, hat, value

Kinsley e McGugan, 2019