

---

# **Invasores Documentation**

***Release 0.9.9***

**Nilo Menezes**

**abr 15, 2017**



---

## Contents:

---

<b>1</b>	<b>Universo</b>	<b>1</b>
<b>2</b>	<b>Objetos</b>	<b>3</b>
<b>3</b>	<b>Jogo</b>	<b>5</b>
<b>4</b>	<b>Nave</b>	<b>7</b>
<b>5</b>	<b>Inimigos</b>	<b>9</b>
<b>6</b>	<b>Classes de Suporte</b>	<b>11</b>
<b>7</b>	<b>Índices e tabelas</b>	<b>13</b>
	<b>Índice de Módulos Python</b>	<b>15</b>



**class** universo.**Universo** (*dimensao: typing.Tuple[int, int], quadros: int = 60*) → None  
Responsável pela manutenção do conjunto de objetos do jogo (*objetodojogo.ObjetoDoJogo*). Esta classe varre sua lista de objetos, chamando o método de respiração de cada objeto, rotina de cálculo de pontos e também gerando o fundo de estrelas.

**adicione** (*objeto: objetodojogo.ObjetoDoJogo*)  
Adiciona um objeto à lista de desenho.

**atualize** ()  
Atualiza o estado do jogo, chamando o método *ObjetoDoJogo.respire()* de todos os objetos na lista de desenho.

**desenhe** (*posicao: typing.List[int], imagem*)  
Desenha a imagem na posição x, y indicada

**desenhe\_fundo** ()  
Apaga a tela e desenha as estrelas

**desenhe\_objetos** ()  
Desenha a lista de objetos na tela

**escreva** (*posicao: typing.List[int], texto, cor, tamanho=None*)  
Escreve uma mensagem de texto na posição x, y passada. Se uma das posições for igual a -1, centraliza no eixo específico

**finalize\_sincronia** ()  
Espera o fim do frame atual.

**gere\_estrelas** () → None  
Gera aleatoriamente as estrelas do fundo. As coordenadas x e y de cada estrela são escolhidas aleatoriamente e representam a posição da estrela. z é o tamanho da estrela.

**objetos = None**  
lista de objetos do jogo

**reconfigura\_video** (*mensagem: int*)  
Usado para trocar de resolução. Atualiza os limites do universo, gera novas estrelas.

**remove** (*objeto: objetodojogo.ObjetoDoJogo*)

Retira um objeto da lista de desenho e da lista de colisões

**teste\_colisao** ()

Verifica se objetos de classes diferentes colidem uns com os outros

**class** `objetodojogo.ObjetoDoJogo` (*nome, pos, imagem=None, tipo=None*)

Implementa os objetos do universo. Um objeto do jogo representa uma imagem que será desenhada a cada frame. Para mudar o estado do objeto, o universo chama o método `respire()` a cada frame.

Quando dois objetos colidem, o valor de dano é utilizado para subtrair um valor da resistência do outro objeto.

Quando a resistência chega a zero, o objeto é removido do universo

**carregue\_imagem** (*nome: str*) → None

**colida** (*objeto*)

Chamado quando dois objetos colidem no jogo. Para evitar que os inimigos colidam entre si, apenas objetos com nomes diferentes podem colidir entre si.

Em caso de colisão, retira da resistencia do objeto atual o dano causado pelo outro objeto.

**dano = None**

dano é o valor subtraído quando algo colide com este objeto

**estado = None**

estado variável utilizada para controlar estados e principalmente troca de imagens

**imagem**

Bitmap usado para desenhar este objeto

**ix = None**

ix é o incremento x aplicado durante a respiração

**iy = None**

iy é o incremento y aplicado durante a respiração

**lx = None**

lx é a largura da imagem em pontos

**ly = None**

ly é a altura da imagem em pontos

**makeRect** () → `pygame.Rect`

Retorna um retângulo com as dimensões deste objeto

**move** (*direcao: int*)

Move o objeto, na direção indicada.

Observar que direção é um int!

- 0 - direita
- 1 - esquerda
- 2 - para baixo
- 3 - para cima

A implementação deste método é responsável por fazer os ajustes de velocidade e posição necessários.

**nome = None**

nome é utilizada para identificar um grupo de objetos

**pos = None**

pos é a posicao inicial do objeto

**resistencia = None**

resistencia é o valor que quando zerado retira o objeto do jogo

**respire** ()

Chamado a cada frame. Utilizado para modificar o estado do objeto.

**retangulo** () → typing.Tuple[int, int, int, int]

**universo = None**

universo aponta para o universo ao qual este objeto pertence seu valor é setado pela classe Universo no momento da inclusão

**visivel = None**

visivel indica se o objeto deve ou não ser desenhado

**class** objetodojogo.**Posicao2D** (x, y)

**add** (ix, iy)

**tuple** ()



```
class invasores.Invasores (tela)
    Esta classe é responsável pelo jogo em si. Toda customização deve ser feita aqui

    atira (evento=None)

    aumentamisseis ()
        Cheat para aumentar o número de mísseis do jogador em 1000

    aumentaresistencia ()
        Cheat para aumentar a resistência do jogador em 1000 pontos

    avanca_fase ()

    baixo ()

    calcula_pontos (universo, a, b)

    carrega_fase ()

    carregue_imagens ()

    cima ()

    cria_alienigena (xi, yi, c, l, script, xl=60, yl=60)

    cria_municao (carga=100)

    cria_resistencia (carga=100)

    direita ()

    esquerda ()

    fase1 ()

    fase2 ()

    faseT ()

    incrementa_tempo_script ()

    inicializa_comandos ()
```

**inicializa\_eventos** ()

**mostra\_texto** (*ttexto*)

**movejoystick** (*evento*)

**movemouse** (*evento*)

**nova\_partida** ()

**para\_tempo\_script** (*motivo*)

**repeticao\_do\_jogo** ()

Loop principal do jogo. Apaga a tela, carrega a fase e repete até o jogador morrer ou escolher para sair.

**saida** (*evento=None*)

**tela\_fim\_de\_jogo** ()

**tela\_inicial** ()

**invasores.jogo** ()

Cria o loop do jogo. Alterna entre os três estados principais do invasores:

- *Invasores.tela\_inicial* ()
- *Invasores.repeticao\_do\_jogo* ()
- *Invasores.tela\_fim\_de\_jogo* ()

O usuário pode pressionar X para sair em qualquer tela.

```
class nave.Nave (nome, pos, imagem=None, tipo='JOGADOR')
    Implementa a nave com aceleração vetorial (x,y) Faz o tratamento de colisão e evita objetos chamados “tiro”

    colida (objeto)

    move (direcao)

    respire ()

nave.sinal (x)
    Retorna o sinal de um numero

class laser.Laser (nome, pos, imagem=None, tipo='JOGADOR')
    Implementa os misseis do jogo.

    colida (objeto)

    move (direcao)

    respire ()

class objetosbonus.ObjetosBonus (nome, pos, imagem=None, tipo='INIMIGO')
    Classe utilizada para representar os objetos de recarga de munição ou resistência

    colida (objeto)

    respire ()
```



```
class alienigena.Alienigena (nome, pos, imagem, tipo='INIMIGO')  
  
    alienigenas_vivos = 0  
    move (direcao)  
    respire ()  
    set_script (script)
```



---

### Classes de Suporte

---

```
class score.Score (nome, pos=[0, 0])
    Utilizada para exibir o score do jogo. Implementada como um objeto normal, podendo inclusive suportar ani-
    mação.

    respire ()

class score.ScoreComFPS (nome, pos)

    respire ()

class score.Texto (nome, pos, texto, tamanho, tempo, universo, cor)
    Utilizada para exibir o score do jogo. Implementada como um objeto normal, podendo inclusive suportar ani-
    mação.

    respire ()
        Decrementa a resistência a cada frame. Com objetivo de fazer o texto sumir após x frames

naleatorios.faixa (inicio, fim=None)
    Gera um número aleatório entre inicio e fim

naleatorios.pega ()
    Retorna um número aleatório entre 0 e 1

som.canais (nome=None)

som.carregue (nome: str, local: str)

som.reproduza (nome: str)

class temporizador.Temporizador (nome, descricao, cabecalho='')
    Classe utilitária, usada para registrar o tempo decorrido entre dois pontos

    fim ()

    inicio (valores=[])

traducao.dicionario (lingua)

traducao.pega (chave)
```

```
class traducao.traducao (prefixo)

    carregaDicionario (nome)
    dicionario = {}
    lingua = 'pt'
    pega (chave)
class video.Imagem (nome, local)

    altura ()
    carregue_imagem (nome, local)
    largura ()
    ponto_croma (x, y)
class video.Video (dimensao, tela_cheia=False)

    adicione (funcao)
    anterior_modos ()
    atualize ()
    desenhe (imagem, posicao)
    faz_tela_cheia (sim=None)
    fonte (tamanho)
    icone (imagem)
    limpe (cor=(0, 0, 0, 0))
    modo (dimensao, tela_cheia=None)
    notifique (mensagem)
    proximo_modos ()
    texto (mensagem, cor)
    titulo (nome)
video.carregue (nome, local)
video.imagem (nome)
```



## CAPÍTULO 7

---

### Índices e tabelas

---

- genindex
- modindex
- search



### **a**

alienigena, [9](#)

### **i**

invasores, [5](#)

### **l**

laser, [7](#)

### **n**

naleatorios, [11](#)

nave, [7](#)

### **o**

objetodojogo, [3](#)

objetosbonus, [7](#)

### **s**

score, [11](#)

som, [11](#)

### **t**

temporizador, [11](#)

traducao, [11](#)

### **u**

universo, [1](#)

### **v**

video, [12](#)



## A

add() (método objetodjogo.Posicao2D), 4  
 adicione() (método universo.Universo), 1  
 adicione() (método video.Video), 12  
 Alienigena (classe em alienigena), 9  
 alienigena (módulo), 9  
 alienigenas\_vivos (atributo alienigena.Alienigena), 9  
 altura() (método video.Imagem), 12  
 anterior\_mod() (método video.Video), 12  
 atira() (método invasores.Invasores), 5  
 atualize() (método universo.Universo), 1  
 atualize() (método video.Video), 12  
 aumentamisseis() (método invasores.Invasores), 5  
 aumentaresistencia() (método invasores.Invasores), 5  
 avanca\_fase() (método invasores.Invasores), 5

## B

baixo() (método invasores.Invasores), 5

## C

calcula\_pontos() (método invasores.Invasores), 5  
 canais() (no módulo som), 11  
 carrega\_fase() (método invasores.Invasores), 5  
 carregaDicionario() (método traducao.traducao), 12  
 carregue() (no módulo som), 11  
 carregue() (no módulo video), 12  
 carregue\_imagem() (método objetodjogo.ObjetoDoJogo), 3  
 carregue\_imagem() (método video.Imagem), 12  
 carregue\_imagens() (método invasores.Invasores), 5  
 cima() (método invasores.Invasores), 5  
 colida() (método laser.Laser), 7  
 colida() (método nave.Nave), 7  
 colida() (método objetodjogo.ObjetoDoJogo), 3  
 colida() (método objetosbonus.ObjetosBonus), 7  
 cria\_alienigena() (método invasores.Invasores), 5  
 cria\_municao() (método invasores.Invasores), 5  
 cria\_resistencia() (método invasores.Invasores), 5

## D

dano (atributo objetodjogo.ObjetoDoJogo), 3  
 desenhe() (método universo.Universo), 1  
 desenhe() (método video.Video), 12  
 desenhe\_fundo() (método universo.Universo), 1  
 desenhe\_objetos() (método universo.Universo), 1  
 dicionario (atributo traducao.traducao), 12  
 dicionario() (no módulo traducao), 11  
 direita() (método invasores.Invasores), 5

## E

escreva() (método universo.Universo), 1  
 esquerda() (método invasores.Invasores), 5  
 estado (atributo objetodjogo.ObjetoDoJogo), 3

## F

faixa() (no módulo naleatorios), 11  
 fase1() (método invasores.Invasores), 5  
 fase2() (método invasores.Invasores), 5  
 faseT() (método invasores.Invasores), 5  
 faz\_tela\_cheia() (método video.Video), 12  
 fim() (método temporizador.Temporizador), 11  
 finalize\_sincronia() (método universo.Universo), 1  
 fonte() (método video.Video), 12

## G

gere\_estrelas() (método universo.Universo), 1

## I

icone() (método video.Video), 12  
 imagem (atributo objetodjogo.ObjetoDoJogo), 3  
 Imagem (classe em video), 12  
 imagem() (no módulo video), 12  
 incrementa\_tempo\_script() (método invasores.Invasores), 5  
 inicializa\_comandos() (método invasores.Invasores), 5  
 inicializa\_eventos() (método invasores.Invasores), 5  
 inicio() (método temporizador.Temporizador), 11  
 Invasores (classe em invasores), 5

invasores (módulo), 5

ix (atributo objetodojogo.ObjetoDoJogo), 3

iy (atributo objetodojogo.ObjetoDoJogo), 3

## J

jogo() (no módulo invasores), 6

## L

largura() (método video.Imagem), 12

Laser (classe em laser), 7

laser (módulo), 7

limpe() (método video.Video), 12

lingua (atributo traducao.traducao), 12

lx (atributo objetodojogo.ObjetoDoJogo), 3

ly (atributo objetodojogo.ObjetoDoJogo), 3

## M

makeRect() (método objetodojogo.ObjetoDoJogo), 3

modo() (método video.Video), 12

mostra\_texto() (método invasores.Invasores), 6

move() (método alienigena.Alienigena), 9

move() (método laser.Laser), 7

move() (método nave.Nave), 7

move() (método objetodojogo.ObjetoDoJogo), 4

movejoystick() (método invasores.Invasores), 6

movemouse() (método invasores.Invasores), 6

## N

naleatorios (módulo), 11

Nave (classe em nave), 7

nave (módulo), 7

nome (atributo objetodojogo.ObjetoDoJogo), 4

notifique() (método video.Video), 12

nova\_partida() (método invasores.Invasores), 6

## O

ObjetoDoJogo (classe em objetodojogo), 3

objetodojogo (módulo), 3

objetos (atributo universo.Universo), 1

ObjetosBonus (classe em objetosbonus), 7

objetosbonus (módulo), 7

## P

para\_tempo\_script() (método invasores.Invasores), 6

pega() (método traducao.traducao), 12

pega() (no módulo naleatorios), 11

pega() (no módulo traducao), 11

ponto\_croma() (método video.Imagem), 12

pos (atributo objetodojogo.ObjetoDoJogo), 4

Posicao2D (classe em objetodojogo), 4

proximo\_mod() (método video.Video), 12

## R

reconfigura\_video() (método universo.Universo), 1

remove() (método universo.Universo), 1

repeticao\_do\_jogo() (método invasores.Invasores), 6

reproduza() (no módulo som), 11

resistencia (atributo objetodojogo.ObjetoDoJogo), 4

respire() (método alienigena.Alienigena), 9

respire() (método laser.Laser), 7

respire() (método nave.Nave), 7

respire() (método objetodojogo.ObjetoDoJogo), 4

respire() (método objetosbonus.ObjetosBonus), 7

respire() (método score.Score), 11

respire() (método score.ScoreComFPS), 11

respire() (método score.Texto), 11

retangulo() (método objetodojogo.ObjetoDoJogo), 4

## S

saida() (método invasores.Invasores), 6

Score (classe em score), 11

score (módulo), 11

ScoreComFPS (classe em score), 11

set\_script() (método alienigena.Alienigena), 9

sinal() (no módulo nave), 7

som (módulo), 11

## T

tela\_fim\_de\_jogo() (método invasores.Invasores), 6

tela\_inicial() (método invasores.Invasores), 6

Temporizador (classe em temporizador), 11

temporizador (módulo), 11

teste\_colisao() (método universo.Universo), 2

Texto (classe em score), 11

texto() (método video.Video), 12

titulo() (método video.Video), 12

traducao (classe em traducao), 11

traducao (módulo), 11

tuple() (método objetodojogo.Posicao2D), 4

## U

universo (atributo objetodojogo.ObjetoDoJogo), 4

Universo (classe em universo), 1

universo (módulo), 1

## V

Video (classe em video), 12

video (módulo), 12

visivel (atributo objetodojogo.ObjetoDoJogo), 4