

The background features several pixelated, 3D blocky characters in various colors (green, orange, yellow, pink, cyan) arranged in a line, receding into the distance. The characters are composed of simple rectangular blocks, giving them a retro, pixelated appearance.

Desenvolvido Jogos com PyGame

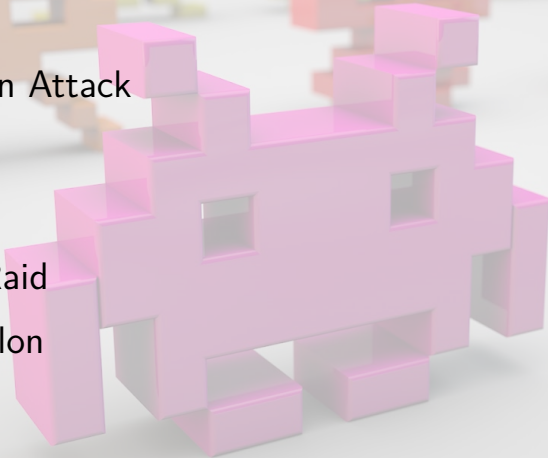
Rafael Guterres Jeffman

Tchelinux

2018

Por que Jogos?

- Daemon Attack
- Pitfall
- Enduro
- River Raid
- Decathlon



Por que Python?

- Expressividade
- Simplicidade
- Multi-paradigma

Por que PyGame?

- Abstrai boa parte das "parada chata".
- *Cross-platform* (SDL)
- Modelo de aplicação simples.

O que é e não é PyGame?

- É uma biblioteca que auxilia no desenvolvimento de jogos.
- Não é um *engine* de jogos.
- É uma coleção de métodos e ferramentas de *baixo nível*.
- Não é para desenvolver o novo FPS 3D a 240fps em 4K.
- Mas é bacana para desenvolver jogos 2D...

Modelo de Aplicação

```
import sys, pygame
pygame.init()

# initialize stuff
while True:
    # handle events
    # update game objects
    # redraw screen stuff
    pygame.display.flip()
```

O Projeto

- Um diretório **media** onde serão armazenados os arquivos de mídia do jogo, com subdiretórios para sons, imagens, vídeos.
- Um diretório **features**, afinal, você vai usar o [behave](#)
- O seu código, bem organizado...

Side Scrolling Shot'em Up

Sua nave foi transportada para o quadrante *gamma*, no meio das hordas inimigas, durante a *Guerra do Infinito*.

Sua missão, aceite ou não, é destruir todos os inimigos.

A Tela da Aplicação

- PyGame utiliza SDL para o gerenciamento da janela.
- Deve ser definido o tamanho da janela na sua criação.
- *Flags* podem ser utilizados para configurar a tela.

```
size = width, height = (800, 600)  
flags = pygame.FULLSCREEN  
screen = pygame.display.set_mode(size, flags)
```

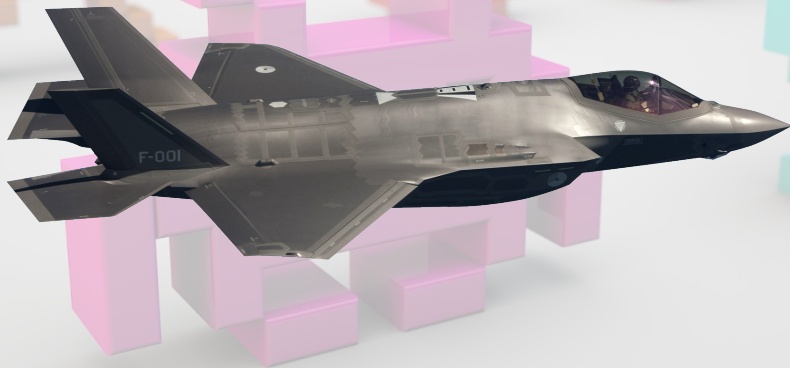
Cenários

- PyGame não tem o conceito de *engine*, logo, o cenário também é parte do código.
- O cenário pode ser criado a partir de um procedimento.
- Mesmo em um ambiente 2D, é possível trazer uma sensação de profundidade.
- Para criar o efeito de *parallax*, você deve criar planos que se movimentam em velocidades diferentes.

Objetos do Jogo

- Uma forma de facilitar a escrita do código é criar objetos de jogo.
- Estes objetos devem prover métodos para criação, movimentação e *rendering*.
- Tratar estes objetos com uma lista de objetos básicos simplifica o código e não afeta a performance.

O Protagonista



Controle

- O controle dos objetos é realizado a partir do loop de eventos.
- Os eventos de teclado são divididos em KEYDOWN e KEYUP.
- Também estão disponíveis eventos de *mouse* e *joystick*.

Mostrando Textos

```
# Desnecessário se você usou pygame.init()
pygame.font.init()

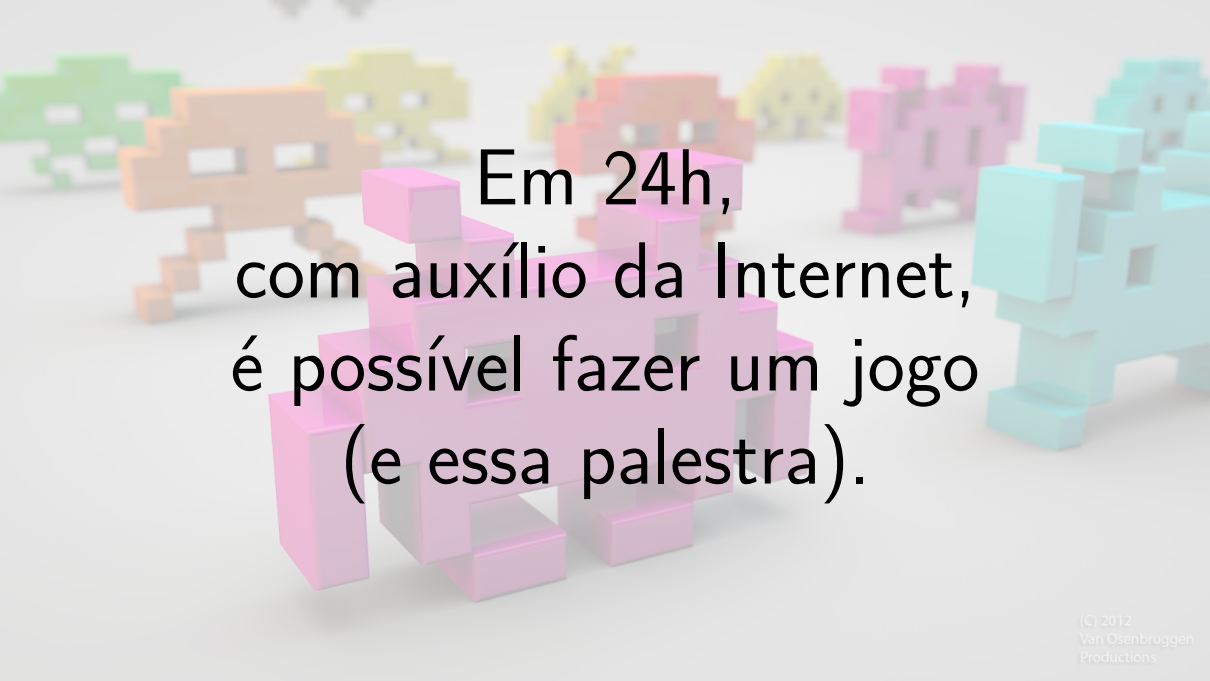
myfont = pygame.font.SysFont('Lucida Sans', 30)

textsurface = myfont.render('Lorem Ipsum', False, (0, 0, 0))

screen.blit(textsurface,(0,0))
```

Quão produtivo é o PyGame?

- Sem conhecer o PyGame...
- Precisando viajar a Pelotas...
- Precisando dormir...
- Precisando ministrar uma aula...
- Precisando criar essa palestras...
- Precisando sair com os cachorros...



Em 24h,
com auxílio da Internet,
é possível fazer um jogo
(e essa palestra).

O Antagonista



The background features several pixelated, 3D blocky characters in various colors (green, orange, yellow, red, pink, cyan) standing on a light gray surface. The characters are rendered in a low-poly, voxel-like style.

E Agora?

<https://pygame.org>

The background features several pixelated, 3D block-like characters in various colors (green, orange, yellow, pink, cyan) arranged in a line, receding into the distance. The characters are constructed from simple rectangular blocks, giving them a retro, digital appearance. They are positioned behind the main text, creating a sense of depth.

Muito Obrigado!

<https://rafaeljeffman.com/tchelixnux>
<https://github.com/rafasgj/tchelixnux-code.git>

rafasgj@gmail.com