Workshop de Python

Vamos aprender Python com a Pytn!

Rui Silveira

Curso L.EIC - Visão Geral

Licenciatura em Engenharia Informática e Computação

Notas últimos candidatos 2023/24

- 1º fase: 173,8
- 2º fase: 183,5

Número de vagas 2023/24:

- 284

Exames

- Matemática A e Física e Química
- Matemática A e Português

Tipos de aula

- Teóricas
- Práticas

Curso L.EIC - Cadeiras

1º Ano				2º Ano				3º Ano			
1º Semestre		2º Semestre		1º Semestre		2º Semestre		1º Semestre (mob)		2º Semestre (mob)	
Escol.	ECTS	Escol.	ECTS	Escol.	ECTS	Escol.	ECTS	Escol.	ECTS	Escol.	ECTS
Álgebra Linear e Geometria Analítica	4,5	Física I	6	Física II	4,5	Métodos Estatísticos	4,5	Interação Pessoa Computador	4,5	Computação Gráfica	6
3 MAT		4 FIS		3 FIS		3 MAT		3 IM		4 IM	
Análise Matemática I	6	Análise Matemática II	6	Bases de Dados	6	Linguagens e Tecnologias Web	6	Laboratório de Bases de Dados e Aplicações Web	6	Projeto Integrador	6
4 MAT		4 MAT		4 SI		4 SI		4 SI		0,15 TM	
Matemática Discreta	6	Teoria da Computação	6	Laboratório de Desenho e Teste de Software	6	Engenharia de Software	6	Programação Funcional e em Lógica	6	Inteligência Artificial	6
4 MAT		4 PRO		4 ES+PRO		4 ES		4 IA+PRO		4 IA	
Fundamentos da Programação	6	Programação	6	Algoritmos e Estruturas de Dados	6	Desenho de Algoritmos	6	Fundamentos de Segurança Informática	6	Compiladores	6
4 PRO		4 PRO		4 PRO		4 PRO		4 CS		4 PRO	
Fundamentos de Sistemas Computacionais	6	Arquitetura de Computadores	6	Sistemas Operativos	6	Laboratório de Computadores	6	Redes de Computadores	6	Computação Paralela e Distribuída	6
4 ACSOR		4 ACSOR		4 ACSOR		4 ACSOR		4 ACSOR		4 ACSOR+PRO	
Projeto UP	1,5			Competências Transversais I	1,5	Competências Transversais II	1,5	Competências Transversais III	1,5		
1 DP				1 QACUP		1 QACUP		1 QACUP			
20	30	20	30	20	30	20	30	20	30	16,15	30

NIAEFEUP - Quem somos e Departamentos

Núcleo de Informática da AEFEUP

- constituído por estudantes a desenvolver para estudantes

Departamentos:

- Projetos
- Eventos
- Comunicação
- Imagem

linktr.ee/niaefeup

NIAEFEUP - Projetos

Aplicação para estudantes

Website de criação de horários

Extensão browser para melhorar o Sigarra

NIAEFEUP - SINF

Workshops

Jogos e Diversão

Networking

Palestras

Competições

Visitas a Empresas

Developer

GitHub

🐍 Gameplay - Pytn

```
🐍 Bem-vindo ao jogo da Pytn!
Usa 'w' (cima), 's' (baixo), 'a' (esquerda) e 'd' (direita) para mover.
⬜ ⬜ ⬜ ⬜ ⬜
⬜ 🍎 ⬜ ⬜ ⬜
⬜ ⬜ 🐍 ⬜ ⬜
⬜ ⬜ ⬜ ⬜ ⬜
⬜ ⬜ ⬜ ⬜ ⬜

Para onde queres mover a Pytn? (w/a/s/d): w
⬜ ⬜ ⬜ ⬜ ⬜
⬜ 🍎 🐍 ⬜ ⬜
⬜ ⬜ ⬜ ⬜ ⬜
⬜ ⬜ ⬜ ⬜ ⬜
⬜ ⬜ ⬜ ⬜ ⬜

Para onde queres mover a Pytn? (w/a/s/d): a
⬜ ⬜ ⬜ ⬜ ⬜
⬜ 🐍 ⬜ ⬜ ⬜
⬜ ⬜ ⬜ ⬜ ⬜
⬜ ⬜ ⬜ ⬜ ⬜
⬜ ⬜ ⬜ ⬜ ⬜

🎉 Parabéns! A Pytn apanhou a comida!
```

🐍 Código - pytn.py

```python
import random

# Tamanho da grelha
GRID_SIZE = 5

# Inicializar posição da Pytn no centro
pytn_x, pytn_y = GRID_SIZE // 2, GRID_SIZE // 2

# Gerar posição aleatória para a comida, garantindo que não está na posição inicial da Pytn
while True:
    comida_x, comida_y = random.randint(0, GRID_SIZE-1), random.randint(0, GRID_SIZE-1)
    if (comida_x, comida_y) != (pytn_x, pytn_y):
        break  # Sai do loop quando a comida não está na mesma posição da Pytn

def desenhar_tabuleiro():
    """Desenha a grelha do jogo com a Pytn e a comida."""
    for y in range(GRID_SIZE):
        linha = ""
        for x in range(GRID_SIZE):
            if x == pytn_x and y == pytn_y:
                linha += "🐍 "  # Pytn
            elif x == comida_x and y == comida_y:
                linha += "🍎 "  # Comida
            else:
                linha += "⬜ "  # Espaço vazio
        print(linha)
    print("\n")  # Linha em branco para espaçamento

print("🐍 Bem-vindo ao jogo da Pytn! \nUsa 'w' (cima), 's' (baixo), 'a' (esquerda) e 'd' (direita) para mover.")
```

Código - pytn.py

```python
while True:
    # Mostrar o tabuleiro
    desenhar_tabuleiro()

    # Pedir input do utilizador
    movimento = input("Para onde queres mover a Pytn? (w/a/s/d): ").lower()

    # Atualizar posição com base no input
    if movimento == "w" and pytn_y > 0:
        pytn_y -= 1  # Sobe
    elif movimento == "s" and pytn_y < GRID_SIZE - 1:
        pytn_y += 1  # Desce
    elif movimento == "a" and pytn_x > 0:
        pytn_x -= 1  # Esquerda
    elif movimento == "d" and pytn_x < GRID_SIZE - 1:
        pytn_x += 1  # Direita
    else:
        print("Movimento inválido. Tenta outra direção!")
        continue

    # Verificar se a Pytn apanhou a comida
    if pytn_x == comida_x and pytn_y == comida_y:
        desenhar_tabuleiro()  # Mostrar o último estado
        print("🎉 Parabéns! A Pytn apanhou a comida!")
        break  # Termina o jogo
```

Developer

GitHub