

Desenhando um tabuleiro de xadrez

Precisamos criar um tabuleiro de xadrez, com casas pintadas de **azul** e **preto**.

Para isto vamos usar uma matriz com zeros e uns para simbolizar as casas do tabuleiro, e você precisará completar as lacunas no código, para finalizar a função `pintar_linha()`.

No código abaixo estão as constantes, a matriz e as funções `pintar()` e `pintar_linha()` adaptadas para funcionarem sem a necessidade de uma classe:

```
PRETO = (0, 0, 0)
AZUL = (0, 0, 255)
TAMANHO = 30
matriz = [
    [1, 0, 1, 0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 1, 0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 1, 0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0, 1, 0, 1],
    [1, 0, 1, 0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0, 1, 0, 1]
]

# O tabuleiro também pode ser feito utilizando um recurso do Python
# para criação de matrizes descomente o código abaixo para testar
# matriz = [[(c+1) % 2 == 0 for c in range(8)] for l in range(8)]
def pintar_linha(numero_linha, linha):
    for numero_coluna, coluna in enumerate(linha):
        x = _____ * TAMANHO
        y = _____ * TAMANHO
        cor = PRETO
        if coluna == 1:
            cor = AZUL
        pygame.draw.rect(tela, cor, (____, ____, ____, ____), 0)

def pintar():
    for numero_linha, linha in enumerate(matriz):
        pintar_linha(numero_linha, linha)
```

Qual alternativa abaixo contém o código correto para a função `pintar_linha()`, calculando os valores das variáveis `x` e `y` e preenchendo as informações do retângulo, para a função `draw.rect()`:

```
pygame.draw.rect(surface, color, (left, top, width, height), width=0)
```

Selecione uma alternativa

A

```
def pintar_linha(numero_linha, linha):
    for numero_coluna, coluna in enumerate(linha):
```

```
x = numero_coluna * TAMANHO
y = numero_linha * TAMANHO
cor = PRETO
if coluna == 1:
    cor = AZUL
pygame.draw.rect(tela, cor, (x, y, TAMANHO, TAMANHO), 0)
```

B

```
def pintar_linha(numero_linha, linha):
    for numero_coluna, coluna in enumerate(linha):
        x = coluna * TAMANHO
        y = linha * TAMANHO
        cor = PRETO
        if coluna == 1:
            cor = AZUL
        pygame.draw.rect(tela, cor, (numero_coluna, numero_linha, TAMANHO, TAMANHO), 0)
```

C

```
def pintar_linha(numero_linha, linha):
    for numero_coluna, coluna in enumerate(linha):
        x = numero_linha * TAMANHO
        y = numero_coluna * TAMANHO
        cor = PRETO
        if coluna == 1:
            cor = AZUL
        pygame.draw.rect(tela, cor, (x, y, TAMANHO, TAMANHO), 0)
```

D

```
def pintar_linha(numero_linha, linha):
    for numero_coluna, coluna in enumerate(linha):
        x = numero_coluna * TAMANHO
        y = numero_linha * TAMANHO
        cor = PRETO
        if coluna == 1:
            cor = AZUL
        pygame.draw.rect(tela, cor, (x, y, coluna, linha), 0)
```

E

```
def pintar_linha(numero_linha, linha):
    for numero_coluna, coluna in enumerate(linha):
        x = linha * TAMANHO
        y = coluna * TAMANHO
        cor = PRETO
        if coluna == 1:
            cor = AZUL
        pygame.draw.rect(tela, cor, (numero_linha, numero_coluna, TAMANHO, TAMANHO), 0)
```