

Universidade Federal do Rio Grande
Bacharelado em Sistemas de Informação
Prova 4º Bimestre
AED I – 2024

Prova referente ao Trabalho Graphics. Nesta avaliação, são apresentadas algumas questões referentes ao trabalho que tu desenvolveste. As questões exigem que tu expliques como os problemas do teu projeto foram resolvidos.

Deves responder com base em teu trabalho, apresentando uma explicação textual simples e copiando “prints” dos trechos de código relacionados.

1) (2,5 Pontos) Quais os comandos necessários para criação de um botão no seu código?

Primeiramente cria-e um design para botão ficar visível em tela, os botões, em sua maioria, são em forma de quadrilátero, sendo assim utiliza-se do comando `gp.Rectangle(gp.Point(x1,y1),gp.Point(x2,y2))`, “gp.Rectangle” é o parâmetro para criar uma figura quadrilátera, depois devemos identificar os pontos x1 e x2, e os pontos y1 e y2. Depois do botão criado, deve-se desenhar ele usando do comando `draw()`, e foi adicionado um texto dentro dos botões para indicar sua funcionalidade.

```
enviar = gp.Rectangle(gp.Point(200,330), gp.Point(300,300))  
enviar.setFill("#64e3a1")  
enviar.setOutline("black")  
enviar.draw(win)  
enviar_text = gp.Text(enviar.getCenter(), 'Cadastrar')  
enviar_text.draw(win)
```

2) (2,5 Pontos) Como, em seu projeto, tu acessas o CSV como sua base de dados?

No projeto, é usado “import csv” para acessar o CSV como base de dados.

```
import csv
```

a) Qual comando foi utilizado para Ler um arquivo CSV?

Para ler o CSV foi criada uma função, para cada vez que fosse necessário acessar o csv, utiliza-se a função em vez de ler o CSV com todos o comandos.

```
def estoquecsv():  
  
    with open('estoque.csv','r') as arq:  
        linhas = arq.readlines()  
        linhas_sem_cabecalho = linhas[1:]  
        estoque = []  
        for linha in linhas_sem_cabecalho:  
            dados = linha.strip().split(',')  
            estoque.append(dados)  
        return estoque
```

Primeiramente é importado o CSV usando o comando “with open(‘arquivo.csv’, ‘r’) as arq”, o parâmetro ‘r’ especifica que estamos importando ele apenas para leitura. “arq.readlines()” lê cada linha do csv e armazena na variável “linhas”, depois exclui-se a primeira linha, pois é o cabeçalho e não precisamos dele no código, cria-se uma lista “estoque” e para cada linha na variável “linha_sem_cabeçalho”, adicionamos essa linha na lista de “estoque”.

b) Qual comando foi utilizado para escrever um arquivo CSV?

Para escrever um CSV utilizamos dois comandos.

Para reescrever o CSV:

```
with open('estoque.csv', 'w', newline='') as arq:

arq = csv.writer(arq)
arq.writerow(['ID', 'Nome', 'Valor', 'Estoque', 'Quantidade Vendida'])
arq.writerows(ver_estoque)
```

Para adicionar uma nova linha ao CSV:

```
with open('estoque.csv', 'a', newline='') as arq:

arq = csv.writer(arq)
arq.writerow([novo_id, nome, valor, estoque, quantidade])
```

O parâmetro ‘w’, especifica que reescreveremos por cima do CSV, enquanto o parâmetro ‘a’ especifica que adicionaremos um novo conteúdo ao CSV. Utiliza-se também o comando “newline=’’” para dizer que queremos que os conteúdos sempre sejam escritos em uma nova linha.

3) (2,5 Pontos) Qual comando é utilizado para identificar o clique do mouse no botão?

Para detectar o clique do mouse no botão, foi criada a função “foi_no_botão()”.

```
def foi_no_botao(botao, onde_cliquei):

btX = botao.getP1().getX()
btY = botao.getP1().getY()
bt2X = botao.getP2().getX()
bt2Y = botao.getP2().getY()
print(f'({btX},{btY}) ({bt2X},{bt2Y}) --- ({onde_cliquei.getX()},
{onde_cliquei.getX()} ({onde_cliquei.getY()}, {onde_cliquei.getY()}))')
if onde_cliquei.getX() > btX and onde_cliquei.getX() < bt2X and
onde_cliquei.getY() < btY and onde_cliquei.getY() > bt2Y:
return True
return False
```

A função “foi_no_botao()” recebe dois parâmetros, o próprio valor do botão para identificar onde ele está desenhado, e onde_cliquei, que é a função que identifica o clique do mouse, então a função “foi_no_botao()” compara se o valor do clique do mouse está dentro do intervalo dos valores (x1,y1) e (x2,y2) do botão, ou seja, dentro do botão.

4) (2,5 Pontos) Indique alguma linha do código que gera mensagem de erro ao usuário quando algo lançado está incorreto.

```
if nome == '' or valor == '' or estoque == '':  
  
erro = gp.Rectangle(gp.Point(0, 440), gp.Point(500, 410))  
erro_text = gp.Text(erro.getCenter(), "Erro ao adicionar o produto, preencha  
todos os campos!")  
erro_text.draw(win)
```

Esse erro está presente dentro da função de cadastro, em que deve-se especificar o nome, o valor e a quantidade do produto a ser colocada no estoque, se qualquer um desses valores não forem inseridos, ao adicionarmos o input nas variáveis de “nome”, “valor” e “estoque”, a partir do parâmetro `.getText()`, a variável recebe nulo, então o “if” testa se qualquer dessas variáveis estiverem nula uma mensagem de erro aparece: “Erro ao adicionar o produto, preencha todos os campos!”