**RELATÓRIO**

**Nome:** Pâmela Mendonça

**Tema:** Preenchimento de Área com Flood Fill

**Enunciado:** O objetivo deste exercício é a implementação de um algoritmo de preenchimento de áreas usando um algoritmo de Flood Fill/Seed Fill. A função deve receber o ponto inicial (semente) para o preenchimento e a nova cor (R,G,B). Implementado a função com 4 e 8 vizinhos para cada ponto.

**Resposta:** Para o algoritmo funcionar, são passados como parâmetro a semente(o pixel que se quer começar a pintar), e a cor que se deseja ser pintada. A partir disso chamamos uma das duas funções,que recebem o ponto escolhido, uma roda o algoritmo percorrendo 4 vizinhos e a outra com 8 vizinhos, mais adiante entenderemos essa diferença.

Ambos os modos começam verificando se esse ponto possui a mesma cor que o ponto inicial dado. Em caso negativo, o programa retorna. Já em caso afirmativo pintamos o pixel passado como parâmetro com a cor escolhida. Em seguida depende do método ser de 8 ou 4 vizinhos:

No caso do de 8: Com o ponto(pixel) inicial pintado, queremos pegar todos os vizinhos ao redor esse pixel(mostrado como laranja), como mostrado na matriz a seguir.

(0,0)(1,0)(2,0)

(0,1)(1,1)(2,1)

(0,2)(1,2)(2,2)

Para pegar seus vizinhos usamos 2 laços de repetição encadeados, o primeiro para o eixo X, começando em X-1 indo até X+1. E dentro dele o segundo laço para o eixo Y, começando em Y-1 indo até Y+1. Dentro do último laço(o do Y) chamamos o método novamente.

No caso do de 4: Seguimos com o ponto(pixel) inicial pintado, em seguida usamos 2 mesmos laços de repetição do método de 8, porém para esse método os laços devem ser um seguido do outro. Dentro de cada um destes laços, chamamos o método novamente.

**Código desenvolvido:**

**// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**// Pinta Semente**

**// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**void Fill8(int x,int y) //8 vizinhos**

**{**

**if ((Image.ReadR(x,y) == OLD\_COLOR.red)&&(Image.ReadG(x,y) == OLD\_COLOR.green)&&(Image.ReadB(x,y) == (int)OLD\_COLOR.blue)) //se o pixel é da mesma cor do antigo seed**

**{**

**//pita**

**Image.DrawPixel(x,y,NEW\_COLOR.red, NEW\_COLOR.green, NEW\_COLOR.blue);//Image(x,y) = NEW\_COLOR;**

**//Para Cada Vizinho (I,J) de (x,y)**

**for(int i=x-1;i<=x+1;i++){**

**for(int j=y-1;j<=y+1;j++){**

**Fill8(i,j);**

**}**

**}**

**}**

**}**

**void Fill4(int x,int y) //4 vizinhos**

**{**

**if ((Image.ReadR(x,y) == OLD\_COLOR.red)&&(Image.ReadG(x,y) == OLD\_COLOR.green)&&(Image.ReadB(x,y) == (int)OLD\_COLOR.blue)) //se o pixel é da mesma cor do antigo seed**

**{**

**//pita**

**Image.DrawPixel(x,y,NEW\_COLOR.red, NEW\_COLOR.green, NEW\_COLOR.blue);//Image(x,y) = NEW\_COLOR;**

**//Para Cada Vizinho (I,J) de (x,y)**

**for(int i=x-1;i<=x+1;i++){**

**Fill4(i,y);**

**}**

**for(int j=y-1;j<=y+1;j++){**

**Fill4(x,j);**

**}**

**}**

**}**

**void Preenche(int Seed\_x,int Seed\_y,unsigned char Novo\_r,unsigned char Novo\_g,unsigned char Novo\_b)**

**{**

**//Image.DrawLine(0, 0, Seed\_x, Seed\_y, 255, 255, 0);**

**Image.ReadPixel(Seed\_x,Seed\_y,OLD\_COLOR.red, OLD\_COLOR.green, OLD\_COLOR.blue); //pega a cor do pixel**

**//atribui a cor nova**

**NEW\_COLOR.red = Novo\_r;**

**NEW\_COLOR.green = Novo\_g;**

**NEW\_COLOR.blue = Novo\_b;**

**//chama o método recursivo para pintar**

**Fill8(Seed\_x, Seed\_y); //ou um ou outro**

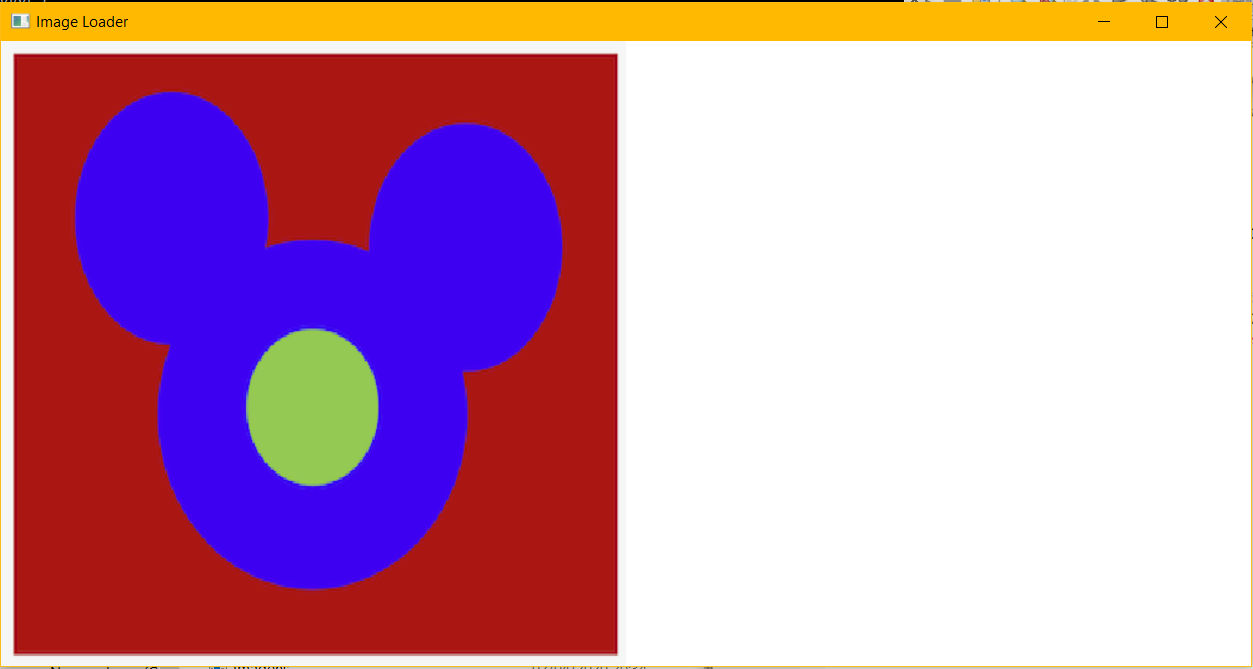
**Fill4(Seed\_x, Seed\_y);**

**display();**

**}**

**Exemplos de Processamento:**

Imagem inicial.

****

Depois de chamar a função para o ponto (180,150) passando a cor branca como desejada.

