

Pedro Inácio Rodrigues Pontes

Prática 15

Laboratório de AEDS

Belo Horizonte, Brasil

2024

1 Introdução

Foi disponibilizado um código que criava um grid infinito no ambiente de programação Processing, ao clicar com o botão esquerdo em uma célula do grid, ela fica preta. Clicando com o botão direito ela fica branca (vazia) novamente. O código permitia movimentação dentro do grid a partir do segurar e arrastar o mouse. E também permitia o zoom, ao mover a roda do mouse.

O problema principal é a implementação de uma nova feature ao código, que é colocar um ciclo de cores para aparecer com os cliques do mouse, ao invés de apenas a cor preta.

A implementação dos conhecimentos de dicionários adquiridos nas aulas de Algoritmo e Estrutura de Dados é necessária para resolver o problema, pois, como o grid é infinito, não seria possível nem prático usar um array de valores guardando o valor de cada célula, ao invés disso, é necessário o uso de um dicionário que armazena a posição e as cores de cada célula.

Portanto, foi objetivada a implementação da mudança cíclica de cores ao clique no botão direito do mouse utilizando a estrutura dicionários, que já foi implementada inicialmente, o necessário era apenas construir código em volta dela utilizando suas características. Vale a pena lembrar que a cor branca não é guardada. O ciclo de cores é: preto, verde, vermelho, azul, amarelo e branco.

2 Desenvolvimento

Segue o código adicionado, com - significando que a linha foi removida e +, que foi adicionada, se não houver nada, ela já existia antes no código.

```
if (mouseButton == LEFT) {
    // Botão esquerdo: preenche a célula com uma cor aleatória
    int novaCor = color(0);
-   coresCelulas.put(chave, novaCor);
+
+   if (coresCelulas.containsKey(chave)) {
+
+       if (coresCelulas.get(chave) == color(0)) {
+           novaCor = color(0, 255, 0);
+       } else if (coresCelulas.get(chave) == color(0, 255, 0)) {
+           novaCor = color(255, 0, 0);
+       } else if (coresCelulas.get(chave) == color(255, 0, 0)) {
+           novaCor = color(0, 0, 255);
+       } else if (coresCelulas.get(chave) == color(0, 0, 255)) {
+           novaCor = color(255, 255, 0);
```

```
+     }  
+     if (coresCelulas.get(chave) == color(255, 255, 0)) {  
+         coresCelulas.remove(chave);  
+     }  
+ } else {  
+     coresCelulas.put(chave, color(0));  
+ }  
+ if (coresCelulas.containsKey(chave))  
+     coresCelulas.put(chave, novaCor);  
+ }
```

Foi adicionado um if para observar se o dicionário `coresCelulas` contém a chave com que queremos trabalhar (a célula que clicamos). Se não a possuir, cairá no bloco `else`, onde colocaremos a chave com a cor preta (0) no dicionário. Se a chave já existir, isso indica que a célula em questão já foi clicada, portanto, será utilizado um bloco *if else if* para averiguar qual a cor armazenada no hashmap (outra forma de se referir ao dicionário). O primeiro verificará se a cor é preta, se for, a variável local `novaCor` será atualizada para verde, essa verificação será repetida com todo o ciclo consecutivo de cores: se for verde a local será atualizada para vermelho, se vermelho para azul... A diferença mais marcante virá quando a cor for amarelo (255,255,0), nesse caso, a variável local não será atualizada, o hashmap da célula em questão que será removido. Isso levará à cor branca.

Após esse bloco if principal *if (coresCelulas.containsKey(chave))*, será feito outro if com os mesmos parâmetros, para atualizar o dicionário com a nova cor. Ele simplesmente chama o método `put` e coloca a nova cor para a chave. Esse código não pode ser acoplado ao primeiro if por haver o caso do amarelo para o branco, onde é removido o hashmap. Se esse acoplamento fosse feito, após ser removido, seria criado novamente o hashmap, fazendo a cor branca não ocorrer, já que ela só ocorre quando o dicionário para a célula não existe.

Observação: `novaCor` teve de ser inicializada com `color(0)` por haver um erro dizendo que ela não foi inicializada, tal ocorre no segundo if principal. Isso vem do verificador de erros do Processing não conseguir averiguar se nova cor existirá com um valor válido quando o segundo if for chamado.

3 Resultados

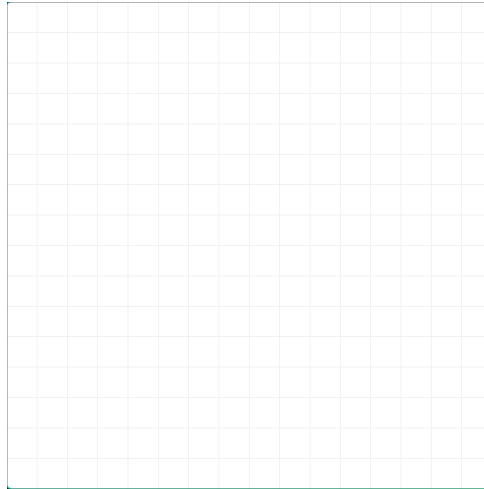


Figura 1 – Grid Inicial

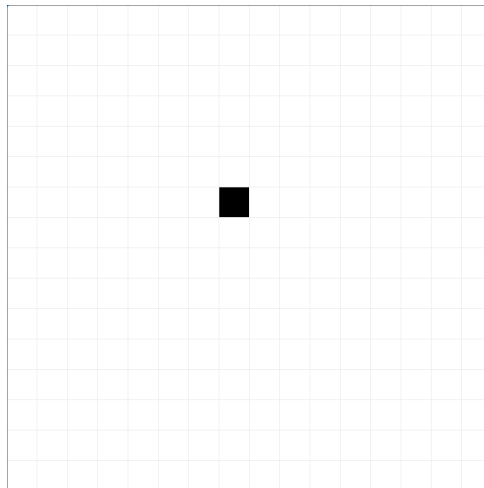


Figura 2 – Após clique nº1 na célula em questão

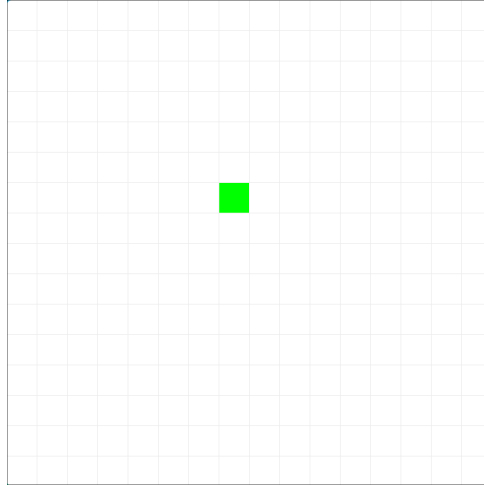


Figura 3 – Após clique nº2

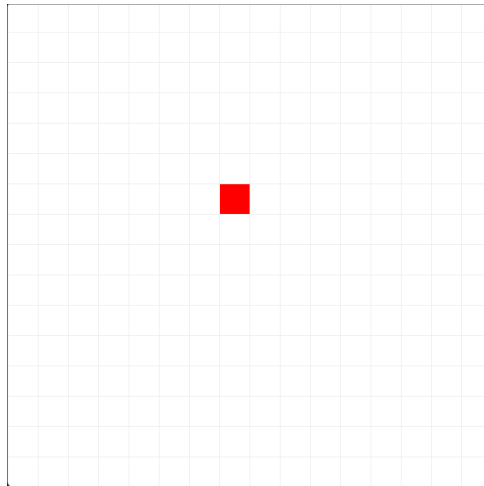


Figura 4 – Após clique nº3

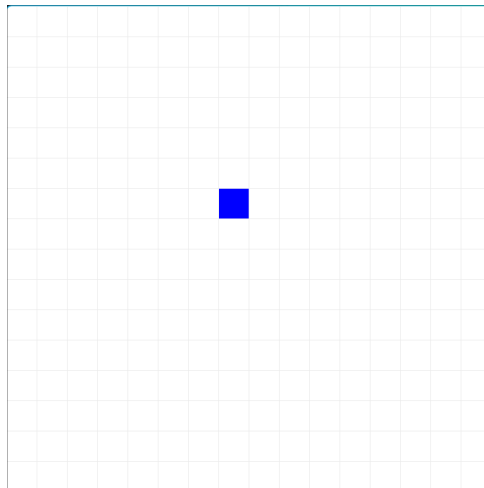


Figura 5 – Após clique nº4

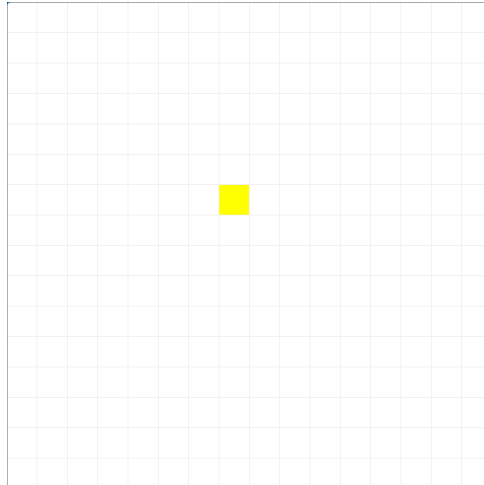


Figura 6 – Após clique nº5

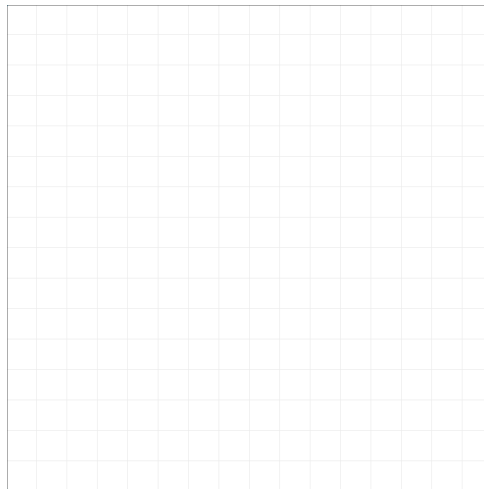


Figura 7 – Após clique nº6

4 Conclusão

- Os objetivos foram devidamente alcançados.
- A maior dificuldade foi fazer a célula ficar branca, pois havia o conflito de ter que remover o dicionário nessa parte e após isso fazer ele não ser criado novamente até o próximo clique. Isso foi resolvido criando um if para ver se havia algo no dicionário e se houvesse, alterar a cor, onde o else levava à criação dele com a cor preta, e outro if que verificava a mesma coisa, mas se existisse ia ser colocada a variável local novaCor dentro do dicionário.
- Foi aprendido o uso prático dos hashmaps e dicionários, principalmente no que se trata aos métodos inerentes deles.