UNIVERSIDADE FEDERAL DE SERGIPE
Departamento de Computação

Relatório de POO:
MiniPaint

Professor: André Britto de Carvalho

Alunos: Rafael Silva Andrade
Yves Luis Bastos Canário Rodrigues

Introdução

O projeto a seguir foi desenvolvido pelos alunos de Engenharia de Computação, Rafael Silva Andrade e Yves Luis Bastos Canário Rodrigues, e é pertinente a nota da terceira unidade da matéria programação orientada a objeto, ministrada pelo docente André Britto de Carvalho. O trabalho possui como objetivo a criação de um MiniPaint, utilizando a linguagem Java e os principais pilares de POO e a biblioteca gráfica Swing.

Desenvolvimento do projeto:

O projeto foi pensado em quatro principais partes:

Interface:

A interface criada foi planejada para que fosse limpa, bem-apresentada e principalmente de fácil utilização para o usuário final. Para isto foi feita a aplicação de ícones para os botões de cores mesclando o azul e o roxo, transmitindo a sensação de tranquilidade, serenidade e harmonia. Além disso, foi colocado o menor número de botões possíveis para que a interface não fosse poluída, dessa maneira, foi feita o uso de menus que são abertos quando se clica com o mouse. A área de interação foi dividida para que cada parte tivesse a sua presença aos olhos do utilizador, ademais, cada região e funcionalidade possui textos de apoio ao se passar o mouse por cima e botões bem esclarecedores.

Desenho das figuras:

O MiniPaint possui 3 figuras básicas: círculo, triângulo e retângulo, sendo que cada figura possui sua forma para ser criada na tela. Para o retângulo e o triângulo o usuário informa sua altura e largura criando assim a figura que deseja, já para o círculo, é pego seu diâmetro para calcular como será a figura.

Para a criação das figuras foi feita a utilização de um JPanel denominado **Panel**, no qual o método **paintComponent** foi sobrescrito e nesse é chamado o método **desenha**r, que é o responsável por desenhar as figuras a cada atualização na tela, método criado de forma abstrata na classe **Form**, e reescrito nas classes **Rectangle**, **Circle** e **Triangle**. Além disso, para a criação de novas figuras é utilizada classe **Desenho**, está que possui o método **criarFigura**, que é chamado todas as vezes que o mouse clica na tela e possui como função a criação dos objetos **Rectangle**, **Circle** ou **Triangle**, após isso o método **desenhar** é chamado e o desenho é feito.

Edição das figuras:

Se o usuário quiser editar uma figura específica é necessário que ele clique com o botão direito duas vezes no nome da figura na lista que fica ao lado direito da tela de desenho, com isso irá aparecer um menu com as seguintes opções: editar e apagar. Ao selecionar o editar, mais três opções irão aparecer, mudar a cor do corpo do desenho, a cor da borda, e uma opção chamada ferramentas, que ao ser clicada uma janela irá aparecer pedindo os dados do novo tamanho e da nova posição.

Para isto ser feito, foi criado métodos na classe **Funcoes** que modificam os atributos desejados do desenho, e redesenham o objeto na tela.

Arquivos:

Graças ao menu **Arquivos** o usuário tem como possibilidade salvar e carregar seus projetos, proporcionando a ele a liberdade de continuar seus desenhos quando quiser, sem se preocupar de perdê-los depois.

Sendo assim, foram desenvolvidos os métodos **carregarArquivos** e **salvarArquivos** na classe **Funcoes,** que armazenam e carregam os objetos desenhados em um arquivo chamado de banco.dat.

Principais erros e suas resoluções:

- Desenho sendo feito fora do JPanel e não colorindo:

Tais erros estavam acontecendo pois o método desenhar recebe como parâmetro **Graphics graficos**, e durante a chamada deste método estava sendo passado como parâmetro getGraphics do JFrame e não do JPanel, o que ocasionou nos erros.

- A possibilidade de carregar o arquivo infinitas vezes:

Ao clicar em abrir o arquivo era possível carregar as figuras diversas vezes, o que ocasionava na criação de desenhos infinitos. Para tentar solucionar isto foi criada um atributo denominado de **carregar**, que por padrão é definido como true, porém, ao carregar o arquivo a mesma é setada para false, ao salvar novamente é colocado como true, dessa forma, impossibilitando a abertura infinita de figuras.

- Apagar e editar o desenho:

Para apagar ou editar um figura foram criados os métodos na classe **Funcoes**: **rmForma**, que deleta o desenho, **redimensionarFigura** e **movimentarFigura**, responsáveis por alterar o tamanho e desloca o desenho,

alterarCor e **alterarCorBorda**, modificam a cor do corpo e das bordas do desenho respectivamente. Esses métodos são chamados ao se clicar na determinada opção do menu, e fazem sua função definida, contudo, a tela não se atualizava e ficava o desenho anterior na tela. Para isso, como foi explicado anteriormente no tópico **Desenho das figuras,** foi criado um JPanel chamado “Panel”, que teve seu método paintComponent sobrescrito para desenhar as figuras armazenadas no ArrayList a cada vez que a tela fosse atualizada.

Conclusão:

Assim, na construção do projeto foram feitas várias pesquisas para completá-lo, e mesmo aparecendo alguns empecilhos no caminho foram buscadas soluções que nos proporcionaram evoluir na linguagem Java, desta maneira, foi possível concluir o trabalho proposto.

Referências:

JAVA. [*S. l.*], 2009. Disponível em: https://pt.stackoverflow.com/. Acesso em: 21 mar. 2019.

JAVA. [*S. l.*], 2015. Disponível em: https://www.devmedia.com.br/. Acesso em: 15 mar. 2019.