

JavaFX Studies/ Aula02: Scenes

Primeiro vamos entender o conceito, temos nossa Scene e aqui vamos criar um Node do tipo Text, temos várias formas de personalizar esse Node mas o principal que devemos tratar agora é entender como adicionamos esse node a nossa Scene.

Sabemos que a nossa Scene possui um root e esse root possui ramificações que possuem os Nodes, logo para adicionar o node ao root da Scene devemos fazer:

```
root.getChildren().add(text);
```

Modificadores do Objeto Text:

```
//Criando um Objeto do tipo Text
Text text = new Text();
text.setText(value:"Eita pouhaaa");//O texto do
nosso obejto, isso pode ser passado direto pelo
construtor
text.setX(value:50);//Declara posições de X e Y em
relação ao root do texto
text.setY(value:50);
text.setFont(Font.font(family:"Verdana",size:50));//
Setando a fonte do texto, ele necessita receber o
objeto do tipo Font, e pode ser passado de
diveeersas formas.
text.setFill(Color.RED);//Modifica a cor do testo e
recebe um Objeto do tipo Paint, e Color é uma
subclasse de Paint, por isso posso usa-lá.

//O text é um tipo de node, logo precisamos
adicionar ele ao nosso root, que já está associado
a nossa Scene.
root.getChildren().add(text);
```

Modificadores do Objeto Line:

```
Line line = new Line();//A maioria
dos métodos abaixo pode ser passado
Run Java pelo próprio construtor.
line.setStartX(value:200);//Declara a
posição de X e Y do início da linha
line.setStartY(value:200);
line.setEndX(value:400);//Declara a
posição de X e Y do fim da linha
line.setEndY(value:400);
line.setStroke(Color.RED);//Modifica
a cor da linha
line.setStrokeWidth(value:5);//
Modifica a largura da linha
line.setOpacity(value:0.5);//Modifica
a opacidade da linha
line.setRotate(value:45);//Rotaciona
a linha em 45 graus

addNode(root, line);//Adiciona a
linha ao root
```

Modificadores do Objeto Rectangle:

```
//Criando um Objeto do tipo Rectangle
Rectangle rectangle = new Rectangle
(); //A maioria dos métodos abaixo
pode ser passado pelo próprio
contrutor.

//Declara a posição de X e Y do
inicio do retangulo
rectangle.setX(value:100);
rectangle.setY(value:100);
//Declara a largura e altura do
retangulo
rectangle.setWidth(value:100);
rectangle.setHeight(value:100);
//Modifica a cor do retangulo
rectangle.setFill(Color.PINK);
//Modifica a opacidade do retangulo
rectangle.setOpacity(value:0.9);
//Aumenta a largura da borda do
retangulo
rectangle.setStrokeWidth(value:5);
//Modifica a cor da borda do retangulo
rectangle.setStroke(Color.BLACK);

//Adiciona o retangulo ao root
addNode(root, rectangle);
```

Modificadores do Objeto Polygon(Criando um triângulo):

```
//Criando um Objeto do tipo Polygon
Polygon triangle = new Polygon();//A
maioria dos métodos abaixo pode ser
passado pelo próprio construtor.

//Adiciona os pontos do triângulo
triangle.getPoints().setAll(
    ...elements:200.0, 200.0,
    400.0, 200.0,
    300.0, 400.0
);
//Modifica a cor do triângulo
triangle.setFill(Color.YELLOW);
//Modifica a opacidade do triângulo
triangle.setOpacity(value:0.9);
//Aumenta a largura da borda do
triângulo
triangle.setStrokeWidth(value:5);
//Modifica a cor da borda do triângulo
triangle.setStroke(Color.BLACK);

//Adiciona o triângulo ao root
addNode(root, triangle);
```

Modificadores do Objeto Circle:

```
//Criando um Objeto do tipo Circle
Circle circle = new Circle();//A
maioria dos métodos abaixo pode ser
passado pelo próprio construtor.

//Declara a posição de X e Y do
centro do círculo
circle.setCenterX(value:200);
circle.setCenterY(value:450);
//Declara o raio do círculo
circle.setRadius(value:100);
//Modifica a cor do círculo
circle.setFill(Color.LIMEGREEN);
//Modifica a opacidade do círculo
circle.setOpacity(value:0.9);
//Aumenta a largura da borda do
círculo
circle.setStrokeWidth(value:5);
//Modifica a cor da borda do círculo
circle.setStroke(Color.BLACK);

//Adiciona o círculo ao root
addNode(root, circle);
```

Adicionando imagens ao root da Scene:

```
//Criando um objeto do tipo  
ImageView, que é um tipo de Node  
ImageView imageView = new ImageView  
(url:"assets/pizza_slice.png");  
//Modifica a posição de X e Y da  
imagem  
imageView.setX(value:450);  
imageView.setY(value:100);  
//Modifica a largura e altura da  
imagem  
imageView.setFitWidth(value:100);  
imageView.setFitHeight(value:100);  
  
//Adiciona a imagem ao root  
addNode(root, imageView);
```

#ID