

DOCUMENTAÇÃO DE SOFTWARE

Nome do programa: DrawIcon

Desenvolvedor: Thiago Castilho Elias

Data da publicação: 26 de setembro de 2019

Linguagem utilizada: java

INTRODUÇÃO

O programa DrawIcon foi desenvolvido para desenhar ícones de maneira não-interativa para aplicativos android. Aplicativos android podem rodar em dispositivos de diferentes tamanhos e resoluções. A fim de fornecer um ícone de aplicação adequado para cada dispositivo, o desenvolvedor precisa criar cópias do ícone em diferentes tamanhos e guardá-los nos respectivos diretórios dentro do diretório **res** do projeto de aplicativo android. Os seguintes diretórios com os respectivos tamanho de ícone devem ser criados:

- ◆ **mipmap-ldpi**: armazena ícones de 36×36 px;
- ◆ **mipmap-mdpi**: armazena ícones de 48×48 px;
- ◆ **mipmap-hdpi**: armazena ícones de 72×72 px;
- ◆ **mipmap-xhdpi**: armazena ícones de 96×96 px;
- ◆ **mipmap-xxhdpi**: armazena ícones de 144×144 px;
- ◆ **mipmap-xxxhdpi**: armazena ícones de 192×192 px.

De um modo geral, os ícones podem ser criados por meio de ferramentas embutidas no programa Android Studio, entretanto, este programa consome muita memória e CPU, não sendo adequado para rodar em um computador de recursos limitados. Quando iniciei o desenvolvimento de aplicações android, optei por montar as aplicações por linhas de comando, usando *scripts* para isso. Como precisava criar ícones para os programas e estava sem acesso aos recursos gráficos do Android Studio, desenvolvi o programa DrawIcon aqui documentado. Os ícones são criados de maneira não-interativa, ou seja, o usuário deve editar o arquivo de texto contendo as especificações do ícone e então clicar em **Make icons** para desenhar os ícones automaticamente. O usuário pode abrir um arquivo icx existente ou criá-lo diretamente na tela de edição de texto do programa DrawIcon.

ARQUIVO ICX

O arquivo icx é um arquivo de texto puro que especifica as formas geométricas que serão desenhadas no ícone. Cada linha representa uma forma geométrica, sendo uma das seguintes: linha, círculo, elipse, retângulo, polígono ou polígono regular. As coordenadas das formas geométricas são dadas em porcentagem, com a origem do sistema cartesiano no canto inferior esquerdo. Para centralizar uma forma geométrica, seu centro deve estar em $x = 50$ e $y = 50$, é como se estivesse se desenhando em uma tela 100×100 , internamente o programa converte as coordenadas para o tamanho específico de cada ícone. A seguir define-se como escrever um arquivo icx, em *itálico* estão atributos que deverão ser substituídos por seus valores reais, os demais caracteres devem ser escritos literalmente. Não se admitem espaços em branco entre o nome do atributo, o sinal de dois

pontos (:) e o valor do atributo, apenas um espaço simples deve separar os diferentes atributos e os itens dentro de uma lista. Os atributos devem ser escritos na ordem exata em que aparecem nos exemplos, `colorDraw` e `colorFill` são os únicos atributos opcionais, atribuídos para PRETO e BRANCO, respectivamente, se não forem explicitamente declarados, os demais atributos são obrigatórios. Todos os atributos numéricos devem ser escritos como números inteiros.

O desenho de uma linha é especificado da seguinte forma:

`Line[x:[x1 x2] y:[y1 y2] colorDraw:COLOR]`

`x1` e `y1` são as coordenadas do ponto inicial da linha e `x2` e `y2` são as coordenadas do ponto final da linha. `COLOR` é uma string que define a cor da linha, os seguintes valores são válidos (escritos sem as aspas): “white”, “black”, “blue”, “red”, “green”, “magenta”, “cyan”, “yellow”, “orange”, “pink”, “gray”. Também é possível especificar a cor no formato `_ffffff`, no qual o caractere underline (`_`) é seguido por três números, cada qual no intervalo 00 a ff no formato hexadecimal, especificando as tonalidades de vermelho, verde e azul, respectivamente, para criar uma nova cor.

O círculo é definido da seguinte forma:

`Circle[x:xc y:yc r:rc draw:STATE fill:STATE colorDraw:COLOR colorFill:COLOR]`

`xc` e `yc` são o centro do círculo nas direções `x` e `y`, respectivamente; `rc` é o raio do círculo, `STATE` pode ser “on” ou “off” (escrito sem as aspas), de forma que `draw:on` significa que a circunferência deverá ser desenhada e `fill:on` indica que o círculo deverá ser preenchido, para desativar o desenho ou o preenchimento deve-se fazer `draw:off` ou `fill:off`, respectivamente, mas ambos não podem estar no estado “off” ao mesmo tempo; `colorDraw` indica a cor para desenhar a circunferência e `colorFill` a cor para preencher o círculo.

Para desenhar uma elipse escreve-se:

`Ellipse[x:xc y:yc rw:rw rh:rh draw:STATE fill:STATE colorDraw:COLOR colorFill:COLOR]`

Os atributos são os mesmos que os encontrados para o círculo, exceto que `rw` é o raio da elipse no eixo `x` e `rh` é o raio da elipse no eixo `y`.

A próxima linha define um retângulo:

`Rectangle[x:[x1 x2] y:[y1 y2] draw:STATE fill:STATE colorDraw:COLOR colorFill:COLOR]`

`x1` e `x2` são os limites do retângulo no eixo `x`, e `y1` e `y2` são os limites do retângulo no eixo `y`.

Pode-se também criar um polígono com:

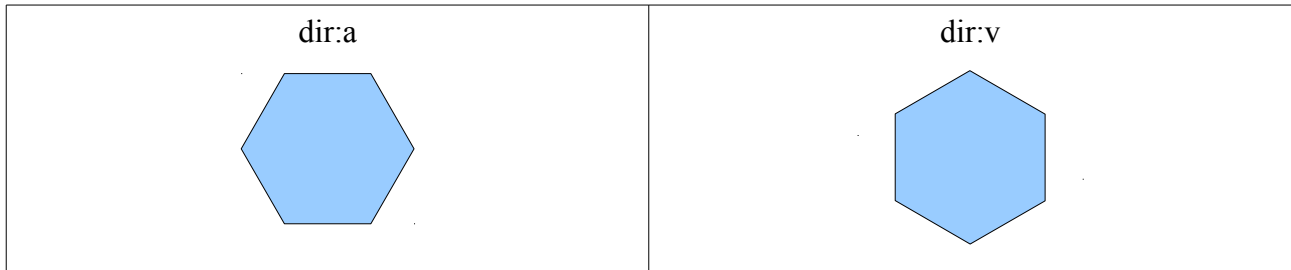
`Poligon[x:[x1 x2 xn] y:[y1 y2 yn] draw:STATE fill:STATE colorDraw:COLOR colorFill:COLOR]`

`x1 x2 xn` é uma lista contendo as coordenadas dos vértices do polígono no eixo `x`, e `y1 y2 yn` é uma lista contendo as coordenadas dos vértices do polígono no eixo `y`. Ambas as listas devem ter o mesmo número de elementos, e o tamanho mínimo das listas é de três elementos.

Por fim, um polígono regular é especificado por:

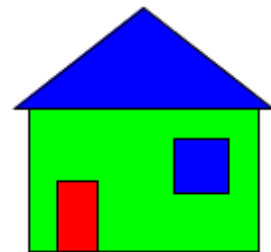
`RegularPoligon[x:xc y:yc r:rc n:n dir:D draw:STATE fill:STATE colorDraw:COLOR colorFill:COLOR]`

xc e yc marcam o centro do polígono nos eixos x e y , respectivamente, n é o número de vértices do polígono, rc é o raio da esfera onde o polígono está inscrito, o atributo `dir` define a direção do polígono, podendo assumir dois valores (escritos sem as aspas): “a”, se o ponto mais alto do polígono for uma aresta, ou “v”, se o ponto mais alto do polígono for um vértice.

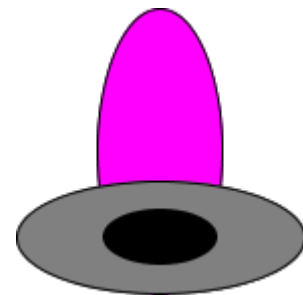


Alguns exemplos de arquivos `icx` e os respectivos ícones criados a partir destes estão listados abaixo:

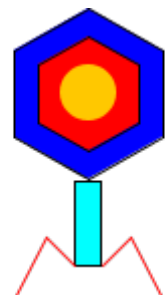
```
Rectangle[x:[10 90] y:[10 60] draw:on fill:on colorFill:green]
Rectangle[x:[20 35] y:[10 35] draw:on fill:on colorFill:red]
Rectangle[x:[60 80] y:[30 50] draw:on fill:on colorFill:blue]
Polygon[x:[5 50 95] y:[60 95 60] draw:on fill:on colorFill:blue]
```



```
Ellipse[x:50 y:50 rw:22 rh:50 draw:on fill:on colorFill:magenta]
Ellipse[x:50 y:20 rw:50 rh:20 draw:on fill:on colorFill:gray]
Ellipse[x:50 y:20 rw:20 rh:10 draw:off fill:on colorFill:black]
```

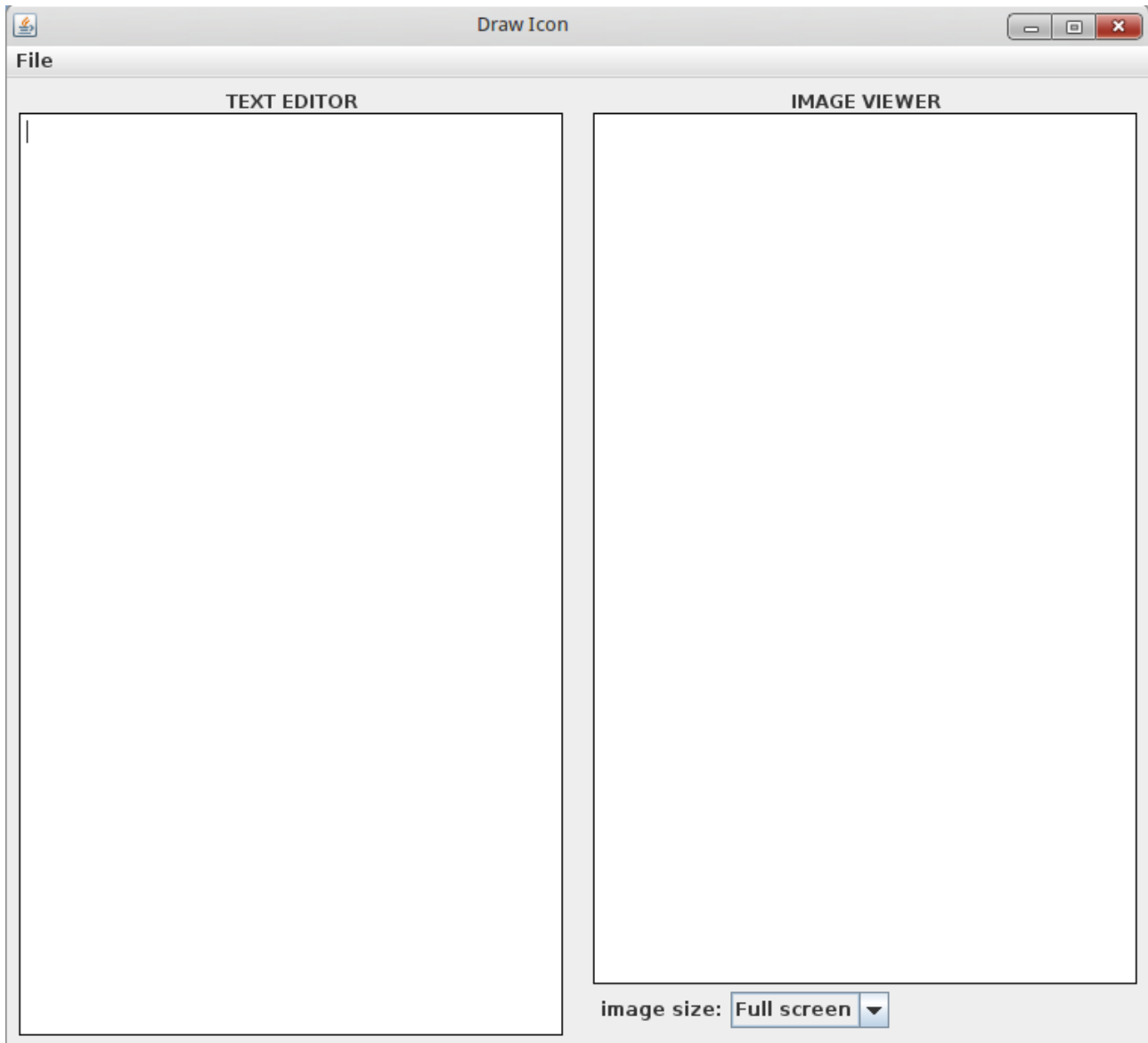


```
RegularPoligon[x:50 y:70 r:30 n:6 dir:v draw:on fill:on colorFill:blue]
RegularPoligon[x:50 y:70 r:20 n:6 dir:v draw:on fill:on colorFill:red]
Rectangle[x:[45 55] y:[10 40] draw:on fill:on colorFill:cyan]
Circle[x:50 y:70 r:10 draw:off fill:on colorFill:orange]
Line[x:[45 35] y:[10 20] colorDraw:red]
Line[x:[35 25] y:[20 0] colorDraw:red]
Line[x:[55 65] y:[10 20] colorDraw:red]
Line[x:[65 75] y:[20 0] colorDraw:red]
```

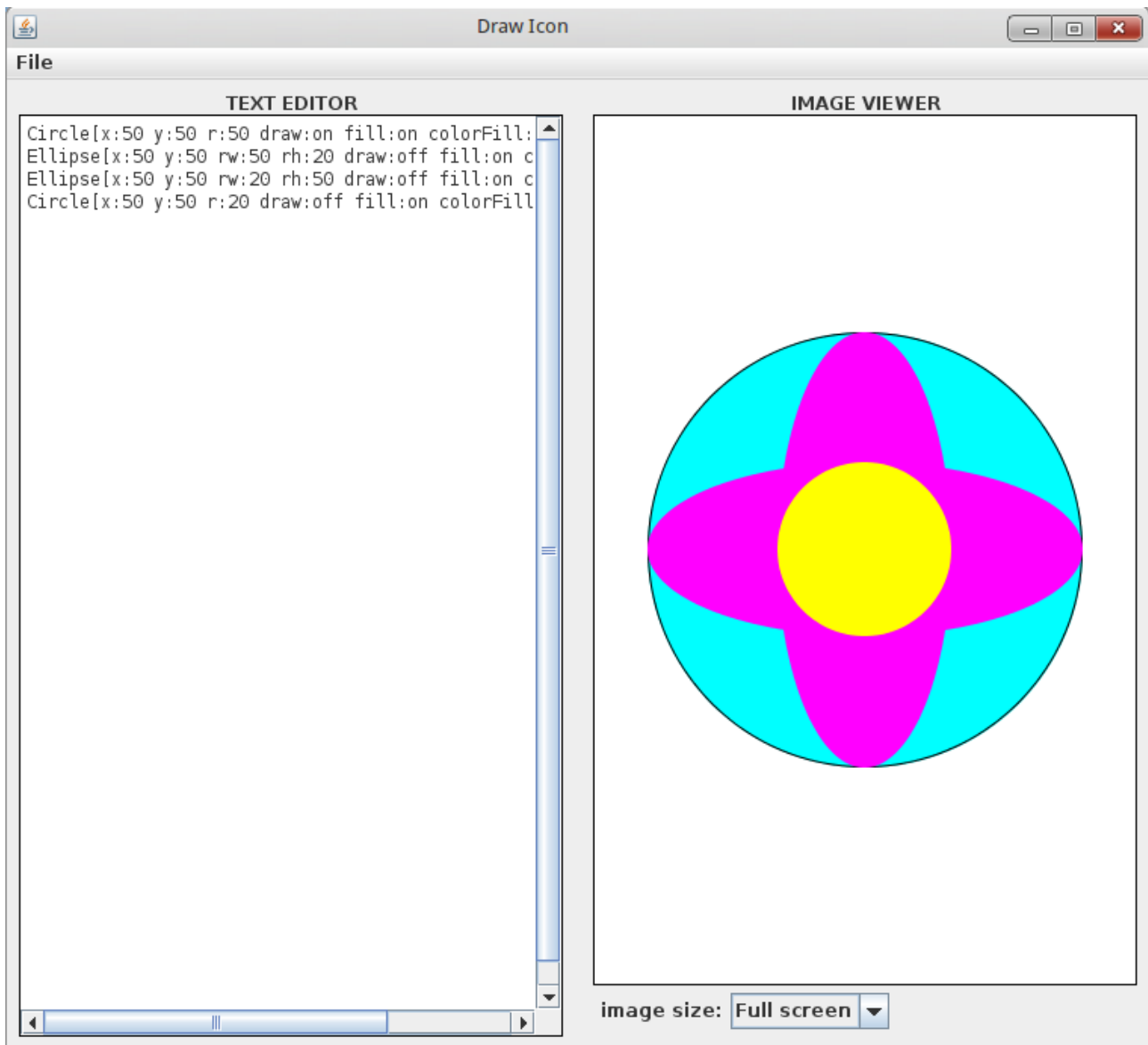


INTERFACE GRÁFICA

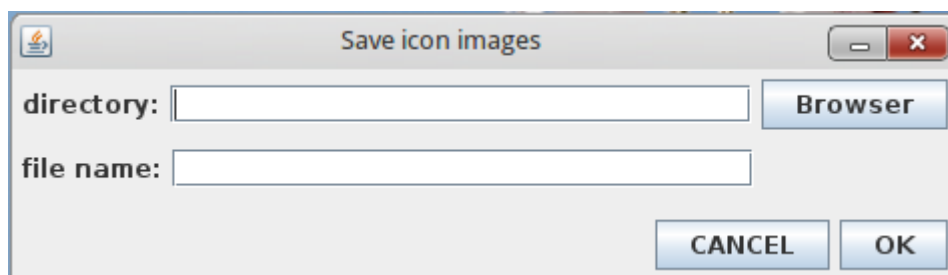
Ao iniciar o programa, a seguinte janela é iniciada:



No lado esquerdo tem-se um painel de edição de texto, onde o usuário pode digitar os elementos gráficos vistos anteriormente. No lado direito tem-se um painel de visualização onde a imagem do ícone é exibida. Para exibir a imagem do ícone, o usuário deve clicar em **File** → **Make icons**, alternativamente, pode-se usar as teclas de atalho Ctrl + M. Abaixo do painel de visualização de imagem há um combo box a partir do qual o usuário pode especificar o tamanho da imagem (ldpi, mdpi, hdpi, xhdpi, xxhdpi, xxxhdpi, ou Full screen), podendo verificar como o ícone ficará nos diferentes tamanhos. Embora o plano de fundo apareça como branco, as imagens salvas têm fundo transparente. A figura seguinte mostra o programa após se digitar algumas linhas no editor de texto e clicar em **File** → **Make icons**.



Para salvar seu ícone, o usuário deve clicar em **File** → **Save icons**, ou usar as teclas de atalho Ctrl + I. É exibida a janela de diálogo abaixo:



O usuário deve especificar na caixa de texto “directory” o diretório onde salvar os ícones, geralmente será o diretório “res” de um projeto android. Dentro desse diretório serão criados automaticamente os subdiretórios descritos na INTRODUÇÃO, dentro dos quais o ícone com tamanho correspondente será salvo. Pode-se clicar no botão “Browser” para procurar o diretório de saída. Na caixa de texto “file name” o usuário deve digitar o nome dos ícones, que serão salvos no formato png, se a extensão .png não for escrita, ela será automaticamente anexada ao nome dos arquivos.

O menu **File** possui cinco itens (entre parênteses as teclas de atalho para acessar o item):

- ◆ Open (Ctrl + O): Abre um arquivo no formato icx e o exibe na tela de edição de texto, automaticamente o ícone é exibido no painel de visualização de imagem;
- ◆ Save (Ctrl + S): Salva o conteúdo da tela de edição de texto para um arquivo icx;
- ◆ Make Icons (Ctrl + M): Gera o desenho do ícone a partir do conteúdo exibido na tela de edição de texto;
- ◆ Save Icons (Ctrl + I): Salva os ícones em seus respectivos diretórios dentro do diretório de saída;
- ◆ Exit (Ctrl + Q): Fecha o programa.