Como criar Menus em Canvas

Existem alguns problemas em algoritmos de renderização de menus que podem impedir uma renderização mais aproriada. Use este código com precaução

Problema

Como exibir um Menu em Canvas, permitir ao usuário fazer rolagem através de diferentes opções, selecionar uma opção e redirecionar o fluxo para o módulo selecionado da aplicação.

Solução

Variáveis

Nós criamos um array de String para as opções dos menus, uma variável (*selectedRowPos*) para manter o registro da opção do menu selecionada e definimos duas fontes e três cores para renderizar os diferentes elementos na tela.

```
 String[] options={"Module1","Module2","Module3"};
 int  selectedRowPos=0;
 int bgcolor=(255<<16)+(255<<8)+255; // Background color - white
 int fgcolor=(0<<16)+(0<<8)+0;  // Foreground color - black
 int hilightcolor=(84<<16)+(171<<8)+67;  // Menu highlight color color
 Font bldFont = Font.getFont(Font.FACE_SYSTEM, Font.STYLE_BOLD, Font.SIZE_SMALL); //
Small Bold
 Font hedFont = Font.getFont(Font.FACE_SYSTEM, Font.STYLE_BOLD, Font.SIZE_MEDIUM); //
Medium Bold
```

Desenhando o Menu

O método apresentado abaixo realiza o desenho de todas as opções do menu usando um menu em Canvas, então a opção do menu selecionada é colocado em destaque desenhando um retângulo preenchido com a cor de destaque e a opção do menu é desenhado dentro dele de novo. Este método pode ser chamado pelo método **paint()** da classe *MenuCanvas*.

```
 private void showMainMenu(Graphics g)
 {
        int WIDTH=Device.WIDTH;
        int HEIGHT=Device.HEIGHT;
        int gap=30

        // Adiciona Background e desenha o cabeçalho
        g.setColor(bgcolor);
        g.fillRect(0,0,WIDTH,HEIGHT);
        g.setColor(fgcolor);
        g.setFont(hedFont);
        g.drawString("MainMenu",5+(WIDTH/2),5,g.TOP|g.HCENTER);

        //Desenha o menu de opções
 g.setFont(bldFont);
 for(int x=0;x<2;x++) {
           g.drawString(options[x],60,65*gap,g.TOP|g.LEFT);  // Print the Menu
        }
```


(C) Copyright Nokia 2014. All rights reserved.

```
 //Destacando a opção de Menu selecionado
        g.setColor(hilightcolor);
        g.fillRoundRect(2, 55+( selectedRowPos)*gap, WIDTH-2, gap-3, 3, 3);
        g.setColor(fgcolor);
        g.drawString(options[selectedRowPos],60,65+( selectedRowPos)*gap,g.TOP|g.LEFT);

 //Desenha uma linha e pinta o rodapé
        g.drawLine(0,HEIGHT-35,WIDTH,HEIGHT-35);
        g.drawString("Select", 2 , HEIGHT-31  , g.LEFT|g.TOP);
        g.drawString("Fire", WIDTH/2 , HEIGHT-31  , g.HCENTER|g.TOP);
        g.drawString("Exit", WIDTH - 2 , HEIGHT-31  , g.RIGHT|g.TOP);
 }
```

Realizando rolagem e seleção dos menus

O processo de rolagem das opções do menu é realizado pelo método **KeyPressed()** do Canvas, onde o valor *selectedRowPos* é alterado quando o usuário aperta ou libera a tecla, e a opção do menu é selecionada na tecla *Left Soft* ou a tecla *Fire* e *Right Soft* pode ser usada para voltar ou sair. O método para redesenho do Canvas é chamado para exibir as mudanção da seleção do menu.

```
public void keyPressed(int keyCode) {
          switch(keyCode) {
                case Device.UP: if(selectedRowPos >0) selectedRowPos --;
                break;
                case Device.DOWN: if(selectedRowPos <2) selectedRowPos ++;
                break;
                case Device.RSK: // Exit
                    selectedRowPos =0;
                    // display.setCurrent(new ExitCanvas(this));
                    break;
                case Device.LSK: // Select
                case Device.FIRE: // Fire
                    gotoModule();
                    break;
            }
        repaint();
    }
```

Invocando o módulo

O fluxo da aplicação é redirecionado basendo-se na seleção dos usuários quando as teclas fire/select são pressionadas.

```
private void gotoModule() {
        switch(selectedRowPos) {
           case 0: //display.setCurrent(new Module1Canvas(this)); // Module1
            break;
            case 1: //display.setCurrent (new Module2Canvas(this)); //Module2
            break;
            case 2: //display.setCurrent (new Module3Canvas(this)); //Module13
            break;
        }
```


(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.