How to make Menu on Canvas

Problem

How to display Menu on Canvas, allow the user to scroll through different options, select an option and to redirect the flow to the selected module of the application.

Solution

Variables

We take an array of Strings for Menu options, a variable (selectedRowPos) to keep the track of selected Menu option and define two fonts and three colors to render different elements on the screen.

```
 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
```

Painting the Menu

The below given method paints all the menu options using a for loop on the Canvas, then the selected menu option is Highlighted by painting a rectangle filled with highlight color and the menu option is drawn over it again. This method may be called from the `paint()` method of the main `MenuCanvas` class.

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

        // Set Background and paint the Header
        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);

        //Drawing the Menu Options
 g.setFont(bldFont);
 for(int x=0;x<3;x++) {
           g.drawString(options[x],60,65+x*gap,g.TOP|g.LEFT);  // Print the Menu
        }

 //Highlighting the selected menu option
        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);
```


(C) Copyright Nokia 2014. All rights reserved.

```
 //Draw Line and paint the Footer
        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);
 }
```

Scrolling Through the Menu & Selection

Scrolling through the menu option is managed by the `KeyPressed` method of the Canvas, in which the value of `selectedRowPos` is changed when user presses up or down key, and Menu option action is selected on Left Soft key or the Fire Key and Right Softkey can be used for back or exit. Repaint method of the Canvas is called to show the change of selection on the menu.

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

Go to Module

The application flow is re-directed based on users selection of option on which the fire/select key is pressed.

```
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.