SIMULADOS & TUTORIAIS

TUTORIAIS FLASH

O Componente ColorPicker

JORGE EIDER

Copyright 2013 – Todos os Direitos Reservados
Jorge Eider F. da Silva

Proibida a reprodução deste documento no todo ou em parte por quaisquer meios, seja digital, eletrônico ou impresso sem a expressa autorização do autor por escrito. Os infratores serão punidos de acordo com a Lei.

O Componente ColorPicker

Introdução

O componente **ColorPicker** permite que o usuário selecione uma cor a partir de uma paleta de cores. O modo padrão desse componente mostra um botão com apenas uma cor. Veja **Figura 5.1** abaixo:

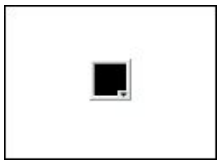


Figura 5.1 – Componente ColorPicker

Quando o usuário clicar no botão, uma paleta de cores disponíveis é mostrada, juntamente com um campo texto exibindo o código hexadecimal da cor selecionada. Veja **Figura 5.2** abaixo:

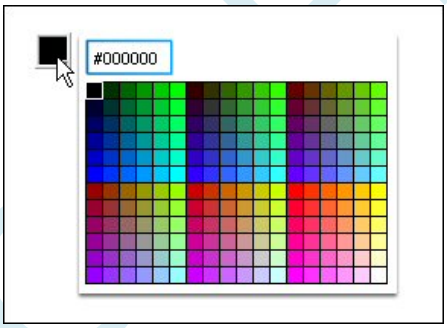


Figura 5.2 – Componente ColorPicker com paleta de cores

Você pode configurar as cores que devem aparecer na paleta do **ColorPicker** simplesmente alterando a propriedade **colors** com os valores das cores que você deseja mostrar, utilizando uma matriz.

Os parâmetros do componente ColorPicker

Você poderá configurar manualmente os parâmetros desse componente tanto no painel **Properties** quanto no painel **Component Inspector**. Lembramos que esses mesmos parâmetros possuem correspondentes no ActionScript com os mesmos nomes, e com isso, você poderá criar aplicações mais robustas em tempo de execução. Veja que parâmetros são esses:

Parâmetro	Descrição
enabled	Esse parâmetro é um valor booleano que indica se o componente deverá estar habilitado ou desabilitado quando a aplicação for executada. O valor padrão é **true**.
selectedColor	Esse parâmetro permite-lhe selecionar a cor padrão da paleta de cores do **ColorPicker**.
showTextField	Esse parâmetro permite-lhe optar pela exibição ou não da caixa de texto que mostra o código hexadecimal da cor. O valor padrão é **false**.
visible	Esse parâmetro é um valor booleano que indica se o

	componente deverá aparecer visível ou oculto quando a aplicação for executada. O valor padrão é: **true**.

Interação do usuário com o componente ColorPicker

O componente **ColorPicker** permite ao usuário selecionar uma determinada cor da paleta e aplicá-la a um outro objeto dentro de sua aplicação. Por exemplo, se você quiser que o usuário personalize os elementos de sua aplicação, tais como a cor de fundo ou a cor de um texto qualquer, você poderia incluir um componente **ColorPiker** e aplicar a cor que o usuário selecionou.
O usuário pode escolher uma cor selecionando na paleta ou digitando o código hexadecimal da cor diretamente na caixa de texto. Após selecionada a cor, você pode usar a propriedade **selectedColor** e aplicar a cor ao texto ou outro objeto qualquer da aplicação.

Uma instância do componente **ColorPicker** recebe o foco se o usuário mover o ponteiro do mouse sobre ele ou usar a tecla **TAB**. Quando a paleta de cores de um componente **ColorPicker** estiver aberta, você poderá usar as seguintes teclas para selecionar a cor desejada:

Tecla	Descrição
Home	Move a seleção para a primeira cor da paleta.
Up	Move a seleção uma cor acima da cor atual.
Down	Move a seleção uma cor abaixo da cor atual.
Right	Move a seleção uma cor para a direita.
Left	Move a seleção uma cor para a esquerda.
End	Move a seleção para a última cor.

Além dos parâmetros especificados anteriormente, você poderá também utilizar as seguintes propriedades, métodos e eventos específicos do componente **ColorPicker**:

As propriedades do componente ColorPicker

São as seguintes as propriedades disponíveis para o componente **ColorPicker**:

Propriedade	Descrição
colors	Permite criar uma matriz de cores personalizadas para serem mostradas na paleta de cores do componente **ColorPicker**.
editable	É um valor booleano que indica se o campo de texto interno do componente **ColorPicker** é editável ou não. O valor padrão é **true**.
enabled	É um valor booleano que indica se o componente poderá aceitar ou não a interação do usuário. O valor padrão é **true**.
hexValue	É uma string que representa o valor da cor selecionada, em hexadecimal.
imeMode	É uma string que permite configurar o modo do método de entrada do editor (IME).
selectedColor	Indica a cor atualmente selecionada na paleta de cores do componente **ColorPicker**.
showTextField	É um valor booleano que indica se o campo de texto interno do componente **ColorPicker** será mostrado.
textField	É uma referência ao campo de texto interno do componente **ColorPicker**.

Os métodos do componente ColorPicker

São os seguintes os métodos disponíveis para o componente **ColorPicker**:

Método	Descrição
ColorPicker	Cria uma nova instância da classe **ColoPicker**.
close	Oculta a paleta de cores.
getStyleDefinition	Recupera o estilo de mapa padrão para o componente atual.
open	Mostra a paleta de cores.

JORGE EDER

Os eventos do componente ColorPicker

Eventos são ações utilizadas pelos componentes para executar tarefas específicas, de acordo com a finalidade de cada situação. Veja a seguir os eventos disponíveis para esse componente e para que servem.

Evento	Descrição
change	É executado quando o usuário clica em uma cor na paleta de cores do **ColorPicker**.
close	É executado quando o usuário fecha a paleta de cores.
enter	É executado quando o usuário pressiona a tecla **Enter** após editar o campo de texto interno do componente **ColorPicker**.
itemRollOut	É executado quando o usuário rola o mouse fora de uma amostra da paleta de cores.
itemRollOver	É executado quando o usuário rola o mouse sobre uma amostra da paleta de cores.
open	É executado quando o usuário abre a paleta de cores.

Veja como utilizar esses recursos nos exemplos mais adiante neste capítulo e no DVD, que acompanha a presente obra.

Criando aplicações com o componente ColorPicker

Vejamos através de exemplos práticos como utilizar esse componente adequadamente nas suas aplicações, inclusive em conjunto com outros componentes:

Exemplo 1

Nesse exemplo inserimos um componente **ColorPicker** a uma aplicação durante o projeto. Aqui, cada vez que uma cor é alterada no painel **ColorPicker**, a função **alteraCor()** chama a função **desenha_quadrado()** para pintar um quadrado com a cor selecionada no **ColorPicker**. Vejamos então como fazer isso:

1. Crie um novo documento (**ActionScript 3.0**);
2. Abra o painel **Components** (**Ctrl + F7**), e arraste um componente **ColorPicker** para o palco;
3. Abra o painel **Properties** e crie uma instância para o componente e chame-a de: **paleta**;
4. Nomeie a camada atual de: **componentes**;
5. Crie uma nova camada e chame-a de: **ações**;
6. Selecione o primeiro frame dessa camada, abra o painel **Actions** (**F9**) e insira o seguinte código:

```
stop();
import fl.events.ColorPickerEvent;
var quadrado:MovieClip = new MovieClip();
// Desenha um quadrado vermelho.
desenha_quadrado (quadrado, 0xFF0000);
addChild(quadrado);
//
paleta.addEventListener(ColorPickerEvent.CHANGE, alteraCor);
//
function alteraCor(event:ColorPickerEvent):void {
   desenha_quadrado (quadrado, event.target.selectedColor);
}
//
function desenha_quadrado (box:MovieClip,color:uint):void {
        box.graphics.beginFill(color, 1);
```

```
        box.graphics.drawRect(100, 150, 100, 100);
        box.graphics.endFill();
}
```

Vejamos como funciona o código:

OBS.: Em todos os exemplos do presente capítulo incluiremos o comando **stop()** na primeira linha do código. Esse comando tem como objetivo interromper a aplicação quando a mesma for executada, mesmo que a aplicação não possua mais de um frame na sua linha do tempo. Consideramos um bom hábito essa prática. Portanto, não comentaremos mais essa linha nos próximos exemplos.

Na linha:

```
import fl.events.ColorPickerEvent;
```

Importamos a classe **ColorPickerEvent** para que possamos utilizar os devidos eventos desse componente.

Na linha a seguir:

```
var quadrado:MovieClip = new MovieClip();
```

Criamos um novo movie clip chamado **quadrado** utilizando o construtor **MovieClip()**.

No bloco de código a seguir:

```
// Desenha um quadrado vermelho.
desenha_quadrado(quadrado, 0xFF0000);
addChild(quadrado);
```

Chamamos a função **desenha_quadrado**, criada mais adiante, para desenhar um quadrado com a cor vermelha. Em seguida, colocamos o respectivo movie clip no palco através do método **addChild()**.

Na linha:

```
paleta.addEventListener(ColorPickerEvent.CHANGE, alteraCor);
```

Utilizamos o evento **CHANGE** do componente **ColorPicker** vinculado à sua instância **paleta**, de forma que quando uma nova cor for selecionada, a função **alteraCor** seja executada, alterando assim a cor do quadrado.

No bloco de código seguinte:

```
function alteraCor(event:ColorPickerEvent):void {
   desenha_quadrado(quadrado, event.target.selectedColor);
}
```

Criamos a função **alteraCor** para alterar a cor do quadrado quando uma nova cor for selecionada no **ColorPicker**.

Finalmente, no bloco de código:

```
function desenha_quadrado(box:MovieClip,color:uint):void {
        box.graphics.beginFill(color, 1);
        box.graphics.drawRect(100, 150, 100, 100);
        box.graphics.endFill();
}
```

JORGE EIDER

Criamos a função **desenha_quadrado()** para criar o quadrado inicial preenchido com a cor definida no parâmetro **color**, quando a aplicação for executada.

7. Execute a aplicação (**Ctrl + Enter**) e confira o resultado com o mostrado na **Figura 5.3** abaixo:

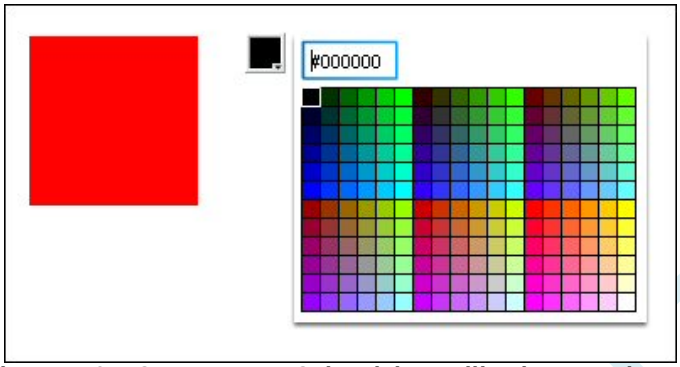


Figura 5.3 – Componente ColorPicker utilizado para alterar a cor de fundo de um movie clip.

8. Clique no componente e selecione uma cor qualquer. O quadrado deverá ficar com a cor selecionada.

*****

Exemplo 2

No exemplo a seguir utilizaremos o construtor **ColorPicker()** para criarmos esse componente em tempo de execução. Em seguida, configuraremos a propriedade **colors** para os valores das cores **Red** (0xFF0000), **Green** (0x00FF00) e **Blue** (0x0000FF) para especificar as cores que o **ColorPicker** deverá mostrar. Criaremos também um componente **TextArea** e inserimos um texto qualquer nele. A cada vez que você selecionar uma cor diferente no **ColorPicker**, a cor do texto do componente **TextArea** será alterada. Vejamos como fazer isso:

1. Crie um novo documento (**ActionScript 3.0**);
2. Abra o painel **Components** (**Ctrl + F7**), e arraste um componente **ColorPicker** para a biblioteca;
3. Arraste também um componente **TextArea** diretamente para a biblioteca;
4. Nomeie a camada atual de: **ações**;
5. Selecione o primeiro frame dessa camada, abra o painel **Actions** (**F9**) e insira o seguinte código:

```
stop();
import fl.controls.ColorPicker;
import fl.controls.TextArea;
import fl.events.ColorPickerEvent;
//
var paleta:ColorPicker = new ColorPicker();
var texto1:TextArea = new TextArea();
var texto2:TextFormat = new TextFormat();
//
paleta.move(100, 100);
paleta.colors = [0xff0000, 0x00ff00, 0x0000ff];
paleta.addEventListener(ColorPickerEvent.CHANGE, alteraCor);
//
```

JORGE FEIDER

```
texto1.text = " Jorge Eider Florentino da Silva é brasileiro, natural de Natal, Estado do Rio Grande do Norte. Foi funcionário do Banco do Brasil S.A. durante 21 anos, onde atuou como consultor, programador e instrutor de vários segmentos na área de informática. Possui cursos oficiais de Flash, Fireworks e Dreamweaver da Adobe. Atualmente ministra cursos de informática em várias Escolas do Estado, é consultor particular, programador e Web Designer. Freqüenta o 5° período (penúltimo) do Curso Superior de Desenvolvimento de Sistemas para a Internet na FATERN-Gama Filho. Desenvolve aplicativos e sites dinâmicos para a Internet utilizando PHP, HTML, CSS, Flash, Dreamweaver, JavaScript, Java, Photoshop, Corel Draw, entre outros. Publicou os seguintes livros: Windows 2000 Professional, CorelDraw 10, Flash 5 e Flash MX, pela Brasport, e Flash MX 2004 Professional e Flash MX 2004 Professional – ActionScript 2.0, pela Campus.";
texto1.setSize(200, 200);
texto1.move(200,100);
//
addChild(paleta);
addChild(texto1);
//
function alteraCor(event:ColorPickerEvent):void {
    if (TextFormat(texto1.getStyle("textFormat"))) {
            texto2 = TextFormat(texto1.getStyle("textFormat"));
    }
    texto2.color = paleta.selectedColor;
    texto1.setStyle("textFormat", texto2);
}
```

Vejamos os comentários sobre o código:

Nas linhas:

```
import fl.controls.ColorPicker;
import fl.controls.TextArea;
import fl.events.ColorPickerEvent;
```

Importamos as classes dos respectivos componentes que iremos utilizar na nossa aplicação.

No bloco de código a seguir:

```
var paleta:ColorPicker = new ColorPicker();
var texto1:TextArea = new TextArea();
var texto2:TextFormat = new TextFormat();
```

Criamos instâncias dos respectivos componentes e da formatação que vamos utilizar no texto, através de seus construtores: **paleta**, **texto1** e **texto2**.

Nas linhas seguintes:

```
paleta.move(100, 100);
paleta.colors = [0xff0000, 0x00ff00, 0x0000ff];
paleta.addEventListener(ColorPickerEvent.CHANGE, alteraCor);
```

Definimos a posição do componente **ColorPicker** (instância: **paleta**) no palco utilizando o método **move()**. Definimos também as cores que deverão ser mostradas no **ColorPicker** através da propriedade **colors**, e por fim, utilizamos um evento **CHANGE** do componente, de forma que quando uma nova cor for selecionada na paleta, a função **alteraCor** seja executada (veja o que essa função faz mais adiante).

JORGE EIDER

No bloco de código seguinte:

```
texto1.text = " Jorge Eider Florentino da Silva é brasileiro, natural de Natal, Estado do Rio Grande do Norte. Foi funcionário do Banco do Brasil S.A. durante 21 anos, onde atuou como consultor, programador e instrutor de vários segmentos na área de informática. Possui cursos oficiais de Flash, Fireworks e Dreamweaver da Adobe. Atualmente ministra cursos de informática em várias Escolas do Estado, é consultor particular, programador e Web Designer. Freqüenta o 5º período (penúltimo) do Curso Superior de Desenvolvimento de Sistemas para a Internet na FATERN-Gama Filho. Desenvolve aplicativos e sites dinâmicos para a Internet utilizando PHP, HTML, CSS, Flash, Dreamweaver, JavaScript, Java, Photoshop, Corel Draw, entre outros. Publicou os seguintes livros: Windows 2000 Professional, CorelDraw 10, Flash 5 e Flash MX, pela Brasport, e Flash MX 2004 Professional e Flash MX 2004 Professional – ActionScript 2.0, pela Campus.";
texto1.setSize(200, 200);
texto1.move(200,100);
```

Definimos o conteúdo do componente **TextArea** através da propriedade **text**, como também o seu tamanho através do método **setSize()** e a sua posição no palco através do método **move()**.

Nas linhas seguintes:

```
addChild(paleta);
addChild(texto1);
```

Utilizamos o método **addChild()** para colocarmos os devidos componentes no palco.

Finalmente, no bloco de código:

```
function alteraCor(event:ColorPickerEvent):void {
    if (TextFormat(texto1.getStyle("textFormat"))) {
            texto2 = TextFormat(texto1.getStyle("textFormat"));
    }
    texto2.color = paleta.selectedColor;
    texto1.setStyle("textFormat", texto2);
}
```

Criamos uma função chamada **alteraCor** de forma que quando uma nova cor for selecionada no componente **ColorPicker** (instância **paleta**), o texto do componente **TextArea** assuma essa nova cor.

6. Execute a aplicação (**Ctrl + Enter**) e confira o resultado com o mostrado na **Figura 5.4** a seguir:

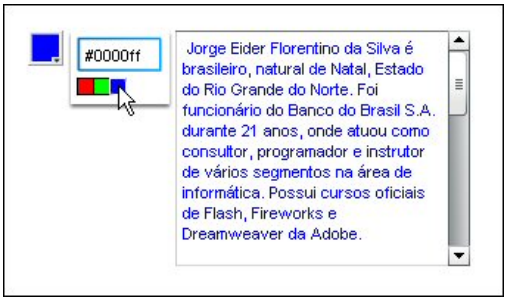


Figura 5.4 – Componente ColorPicker mostrando apenas as cores que queremos utilizar para alterar a cor do texto inserido em um componente TextArea.

JORGE EIDER

7. Clique no componente e selecione uma das três cores disponíveis. A cor do texto do componente **TextArea** deverá ser alterada para a cor selecionada.

*****

Exemplo 3

Nesse exemplo criaremos uma instância de um componente **ColorPicker()** utilizando o seu construtor para exibir apenas oito tonalidades da cor verde. Para isso, criamos uma matriz para armazenar os códigos hexadecimais dessas cores através da propriedade **colors**. Vejamos como fazer isso:

1. Crie um novo documento (**ActionScript 3.0**);
2. Abra o painel **Components** (**Ctrl + F7**), e arraste um componente **ColorPicker** para a biblioteca;
3. Nomeie a camada atual de: **ações**;
4. No primeiro frame dessa camada insira o seguinte código:

```
stop();
import fl.controls.ColorPicker;
//
var paleta:ColorPicker = new ColorPicker();
paleta.move(50,30);
paleta.colors = [ 0x001100,
        0x003300,
        0x005500,
        0x007700,
        0x009900,
        0x00BB00,
        0x00DD00,
        0x00FF00 ];
addChild(paleta);
```

Vejamos os comentários sobre o código:

Na linha:

```
import fl.controls.ColorPicker;
```

Importamos a classe necessária do componente **ColorPicker** para que possamos utilizá-lo adequadamente.

Na linha a seguir:

```
var paleta:ColorPicker = new ColorPicker();
```

Criamos uma nova instância do componente **ColorPicker** através de seu construtor e atribuímos à variável **paleta**.

No bloco de código abaixo:

```
paleta.move(50,30);
paleta.colors = [ 0x001100,
        0x003300,
        0x005500,
        0x007700,
        0x009900,
        0x00BB00,
```

JORGE EIDER

```
0x00DD00,
0x00FF00 ];
```

Utilizamos o método **move()** para posicionar o componente no palco. Em seguida, criamos uma matriz com os códigos hexadecimais das cores que desejamos mostrar através do **ColorPicker** utilizando a propriedade **colors**. No nosso caso, as cores são variações da cor verde.

E finalmente na linha:

```
addChild(paleta);
```

Utilizamos o método **addChild()** para colocar o respectivo componente no palco.

5. Execute a aplicação (**Ctrl + Enter**) e confira o resultado com o mostrado na **Figura 5.5** a seguir.

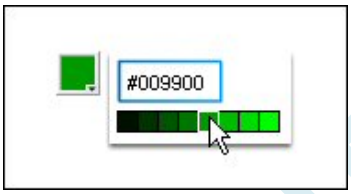


Figura 5.5 – Componente ColorPicker mostrando diferentes tonalidades de uma cor.

*****

Exemplo 4

Nesse exemplo utilizaremos um componente **ColorPicker()** para exibir as cores da paleta padrão, juntamente com um componente **CheckBox**, em tempo de execução, de forma que quando o **CheckBox** estiver selecionado, a caixa de texto que informa o código hexadecimal da cor selecionada será mostrada, caso contrário, ficará oculta. Vejamos então como fazer isso:

1. Crie um novo documento (**ActionScript 3.0**);
2. Abra o painel **Components** (**Ctrl + F7**), e arraste um componente **ColorPicker** e um componente **CheckBox** para a biblioteca;
3. Nomeie a camada atual de: **ações**;
4. No primeiro frame dessa camada insira o seguinte código:

```
stop();
import fl.controls.CheckBox;
import fl.controls.ColorPicker;
//
var meuCB:CheckBox = new CheckBox();
meuCB.textField.autoSize = TextFieldAutoSize.LEFT;
meuCB.label = "Mostra caixa do código das cores";
meuCB.move(10, 10);
meuCB.selected = true;
addChild(meuCB);
//
meuCB.addEventListener(Event.CHANGE, alteraCampo);
//
var paleta:ColorPicker = new ColorPicker();
paleta.showTextField = meuCB.selected;
paleta.move(meuCB.x, meuCB.y + meuCB.height);
```

```
paleta.open();
addChild(paleta);
//
function alteraCampo(event:Event):void {
    paleta.showTextField = meuCB.selected;
}
```

Vejamos como funciona o código:

Nas linhas:

```
import fl.controls.CheckBox;
import fl.controls.ColorPicker;
```

Importamos as classes dos respectivos componentes para que possamos utilizar os seus recursos convenientemente.

Na linha a seguir:

```
var meuCB:CheckBox = new CheckBox();
```

Criamos uma instância do componente **CheckBox** utilizando o respectivo construtor e associamos à variável **meuCB**.

No bloco de código seguinte:

```
meuCB.textField.autoSize = TextFieldAutoSize.LEFT;
meuCB.label = "Mostra caixa do código das cores";
meuCB.move(10, 10);
meuCB.selected = true;
```

Definimos a localização e o rótulo do **CheckBox**. Definimos a posição do **CheckBox** no palco (**move**), e decidimos manter o **CheckBox** já selecionado (**selected**) quando ao aplicação for executada.

No bloco a seguir:

```
meuCB.addEventListener(Event.CHANGE, alteraCampo);
addChild(meuCB);
```

Utilizamos um evento **CHANGE** para o **CheckBox** (**meuCB**), de forma que quando o mesmo for clicado a função **alteraCampo** seja executada. Em seguida, utilizamos o método **addChild()** para colocar o respectivo componente no palco.

Na linha a seguir:

```
var paleta:ColorPicker = new ColorPicker();
```

Criamos uma nova instância do componente **ColorPicker** (**paleta**) utilizando o respectivo construtor do componente.

No bloco seguinte:

```
paleta.showTextField = meuCB.selected;
paleta.move(meuCB.x, meuCB.y + meuCB.height);
paleta.open();
addChild(paleta);
```

JORGE EIDER

Definimos que o campo de texto do componente **ColorPicker** será mostrado (**showTextField**) quando a aplicação for executada, tendo em vista que o **CheckBox** estará selecionado. Definimos também a posição (**move**) da paleta de cores no palco e que a mesma será aberta (**open**) quando a aplicação for executada. Em seguida, utilizamos o método **addChild()** para colocar o respectivo componente no palco.

E, finalmente no código a seguir:

```
function alteraCampo(event:Event):void {
    paleta.showTextField = meuCB.selected;
}
```

Criamos uma função chamada **alteraCampo** de forma que quando for executada, mostre o campo de texto da paleta de cores do componente **ColorPicker**.

5. Execute a aplicação (**Ctrl + Enter**) e confira o resultado com o mostrado na **Figura 5.6** a seguir.

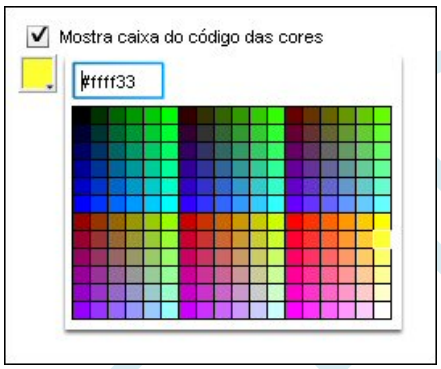


Figura 5.6 – Componente ColorPicker utilizado conjuntamente com um componente CheckBox para mostrar/ocultar a caixa de texto que mostra o código hexadecimal da cor atual do ColorPicker.

*****

Exemplo 5

Nesse exemplo utilizaremos um componente **ColorPicker** em tempo de execução, de forma que quando uma cor for selecionada na paleta, seu valor hexadecimal seja mostrado no painel **Output**. Vejamos então como fazer isso:

1. Crie um novo documento (**ActionScript 3.0**);
2. Abra o painel **Components** (**Ctrl + F7**), e arraste um componente **ColorPicker** para a biblioteca;
3. Nomeie a camada atual de: **ações**;
4. No primeiro frame dessa camada insira o seguinte código:

```
stop();
import fl.controls.ColorPicker;
import fl.events.ColorPickerEvent;
//
var paleta:ColorPicker = new ColorPicker();
paleta.move(10, 10);
addChild(paleta);
//
paleta.addEventListener(ColorPickerEvent.CHANGE, mostraValor);
function mostraValor (event:ColorPickerEvent):void {
```

```
    trace("Cor selecionada:", event.color, "(#" + event.target.hexValue + ")");
}
```

Vejamos como funciona o código:

Nas linhas:

```
import fl.controls.ColorPicker;
import fl.events.ColorPickerEvent;
```

Importamos as classes necessárias para que possamos utilizar o componente adequadamente.

No código seguinte:

```
var paleta:ColorPicker = new ColorPicker();
paleta.move(10, 10);
addChild(paleta);
```

Criamos uma nova instância do componente **ColorPicker** utilizando seu respectivo construtor. Em seguida, definimos sua posição no palco utilizando o método **move()**, e logo após o colocamos no palco através do método **addChild()**.

No bloco de código:

```
paleta.addEventListener(ColorPickerEvent.CHANGE, mostraValor);
// 
function mostraValor (event:ColorPickerEvent):void {
    trace("Cor selecionada:", event.color, "(#" + event.target.hexValue + ")");
}
```

Utilizamos um evento **CHANGE** do componente **ColorPicker** vinculado à instância **paleta**, de forma que quando uma cor da paleta for selecionada a função **mostraValor** seja executada. Logo em seguida criamos essa função cuja finalidade é mostrar no painel **Output** o código hexadecimal dessa cor.

5. Execute a aplicação (**Ctrl + Enter**) e confira o resultado com o mostrado na **Figura 5.7** abaixo:

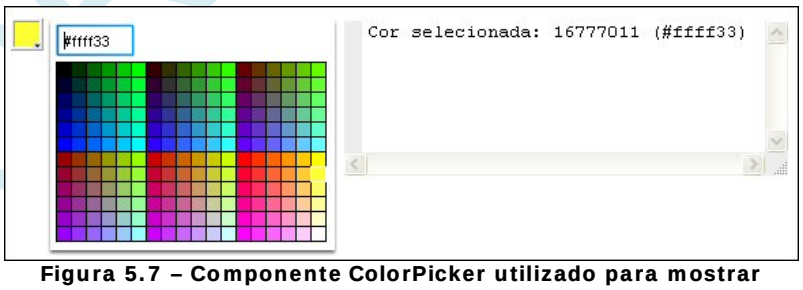


Figura 5.7 – Componente ColorPicker utilizado para mostrar no painel Output o código hexadecimal da cor selecionada.

*****

Exercícios de Fixação

1. Se quiséssemos alterar a cor padrão do componente **ColorPicker** que parâmetro do painel **Properties** utilizaríamos?

   a) showColor

JORGE EIDER

b) changeColor
c) moveColor
d) selectedColor
e) Não é permitido alterar

2. A caixa de texto que acompanha o componente **ColorPicker** serve para:

a) Inserir o nome da cor que se deseja utilizar.
b) Inserir o código decimal da cor desejada.
c) Inserir o código hexadecimal da cor desejada.
d) Definir as cores que deverão ser mostradas no componente.

3. Se não quisermos que a caixa de texto que acompanha o componente **ColorPicker** seja mostrada, que parâmetro do painel **Properties** devemos alterar para **false**?

a) selectedText
b) selectedTextField
c) showTextField
d) showText
e) visibleText

4. Em um componente **ColorPicker**, se estivermos com a paleta de cores aberta, qual a tecla que podemos utilizar para selecionar a última cor?

a) Ctrl+End
b) Shift+Alt+End
c) Seta direita+End
d) Alt+End
e) End

5. Qual a propriedade que permite ao usuário editar o conteúdo da caixa de texto de um componente **ColorPicker**?

a) enabled
b) editable
c) enable
d) editabled

6. Qual dos métodos abaixo é utilizado para ocultar a paleta de cores de um componente **ColorPicker**?

a) hidden
b) enter
c) close
d) hide

7. Qual a propriedade que devemos utilizar para mostrar uma paleta de cores de um componente **ColorPicker**?

a) show
b) showColor
c) openColor
d) open

8. O que faz o evento **itemRollOver** de um componente **ColorPicker**?

a) É executado quando o usuário rola o mouse sobre o componente **ColorPicker**.
b) É executado quando o usuário rola o mouse sobre uma amostra da paleta de cores.
c) É executado quando o usuário arrasta o componente **ColorPicker**.

JORGE EIDER

d) É executado quando o usuário rola o mouse para fora da paleta de cores.

9. A linha de código a seguir:

   paleta.colors = [0x000000, 0xffffff];

   Indica que:

   a) Serão mostradas as cores preta e branca na paleta.
   b) Serão mostradas todas as cores da paleta, do preto ao branco.
   c) Nenhuma cor será mostrada, pois o código está errado.
   d) Serão mostradas as cores preta e vermelha.

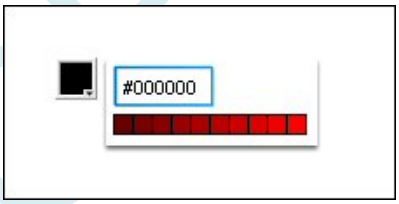
10. Qual das alternativas abaixo cria uma nova instância de um componente **ColorPicker** chamada **paleta**?

    a) var paleta:ColorPicker = newColorPicker();
    b) var paleta:ColorPicker = ColorPicker();
    c) var paleta:ColorPicker = new ColorPicker();
    d) var paleta:ColorPicker = new.ColorPicker();

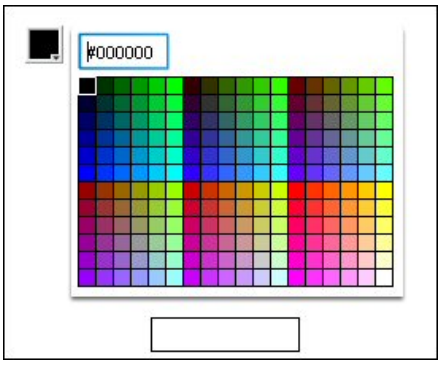
****

Exercícios Propostos

1. Crie uma aplicação de forma que quando o componente **ColorPicker** for clicado mostre na paleta de cores dez tonalidades da cor vermelha. Veja uma sugestão para o layout desse exercício na figura abaixo:

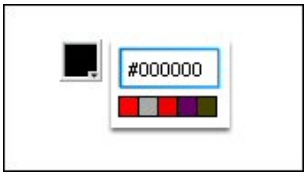


2. Crie uma aplicação utilizando um componente **ColorPicker** de forma que quando a tecla **ENTER** for pressionada mostre em um campo de texto dinâmico o valor da cor selecionada. Veja uma sugestão para o layout desse exercício na figura abaixo:



3. Crie uma aplicação de forma que a paleta de cores mostre 5 cores aleatórias de uma matriz de 15 cores, toda vez que a aplicação for executada. Veja uma sugestão para o layout desse exercício na figura abaixo:

JORGE EIDER



*****

JORGE EIDER