

## Objetivos

- Criar um *custom* control que se desenha a si próprio;
- Eventos relacionados com o teclado;

## Introdução

Muitas vezes os controlos que o *Lazarus* nos disponibiliza não são suficientes para resolver os problemas que temos de enfrentar ao desenvolver um novo *software*.

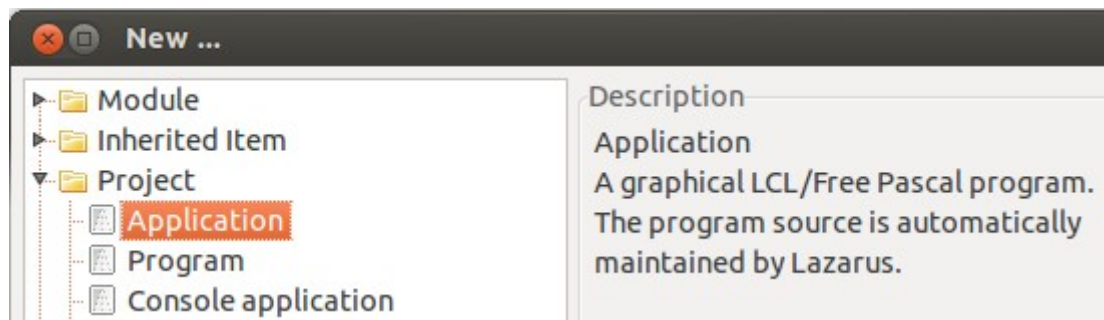
Assim, esta ficha contém um exemplo de um controlo que se desenha dentro de um *form*. Desta forma controlamos completamente o conteúdo e a funcionalidade do controlo.

## Tarefas

1. Criar o projeto;
2. Inserir e configurar uma *TPaintBox*;

## Tarefa 1 - Criar o projeto

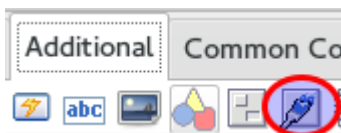
1. Crie uma nova pasta para este projeto. Por exemplo "customControl".
2. Crie um novo projeto no *Lazarus* do tipo "Application";

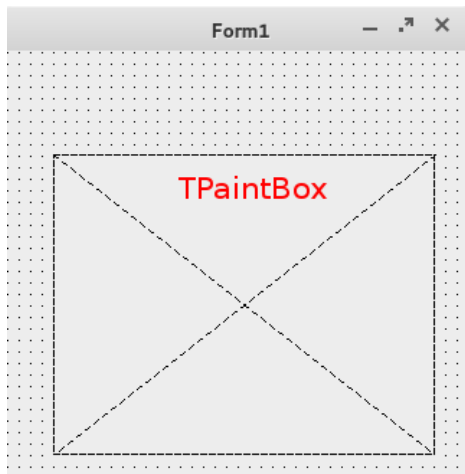


3. Grave o projeto como "customControl";  
**nota:** atribua o nome "customControl" ao projeto e "Main" à unit1;

## Tarefa 2 – Inserir e configurar uma *TPaintBox*;

4. Adicione uma *TPaintBox* ao Form principal da aplicação;





5. Agora vamos “desenhar” dentro da TpaintBox. Para isso temos de programar o *handler* para o evento onPaint da TpaintBox inserida no form.

No evento onPaint do objeto PaintBox1 coloque o seguinte código:

```
procedure TForm1.PaintBox1Paint(Sender: TObject);
var
  Bitmap: TBitmap;
  rectangleHeight, rectangleWidth: integer;
begin
  Bitmap := TBitmap.Create;
  // Initializes the Bitmap Size
  Bitmap.Height := PaintBox1.Height;
  Bitmap.Width := PaintBox1.Width;

  //Draw the background in white
  Bitmap.Canvas.Pen.Color := clWhite; //Line Color
  Bitmap.Canvas.Brush.Color := clWhite; //Fill Color
  Bitmap.Canvas.Rectangle(0, 0, PaintBox1.Width, PaintBox1.Height);

  // Draws squares
  Bitmap.Canvas.Pen.Color := clBlack; //Line Color
  Bitmap.Canvas.Brush.Color := clYellow; //Brush color

  rectangleHeight := PaintBox1.Height div 8;
  rectangleWidth := PaintBox1.Width div 8;
  Bitmap.Canvas.Rectangle(0, 0, rectangleWidth, rectangleHeight);

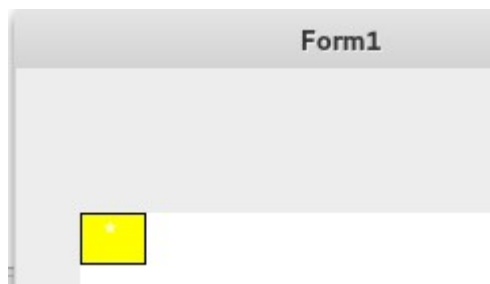
  //Write some text, in this case an *
  //Define Font properties
  Bitmap.Canvas.Font.Name := 'Liberation Mono';
  Bitmap.Canvas.Font.Style := [fsBold];
  Bitmap.Canvas.Font.Size := 10;
  Bitmap.Canvas.Font.Color := clWhite;
  //Write the text
  Bitmap.Canvas.TextOut(rectangleWidth div 3, rectangleHeight div 6, '*');

  PaintBox1.Canvas.Draw(0, 0, Bitmap);

  Bitmap.Free; //Free the memory used by the object Bitmap
end;
```

6. Corra o Programa (F9) e verifique que neste momento tem um retângulo amarelo no canto superior

esquerdo da PaintBox, com um "\*" branco.



7. Agora vamos mudar a cor do retângulo à medida que o utilizador pressiona as teclas de cursor esquerda, direita, cima e baixo. As cores deverão ser as seguintes:

**Cima:** R:150 G:0 B:0

**Baixo:** R:0 G:100 B:150

**Esquerda:** R:130 G:200 B:0

**Direita:** R:255 G:130 B:0

Mas para isso é necessário criar uma variável para guardar a cor que o retângulo deverá ter:

```
var
  Form1: TForm1;
  myBrushColor: TColor;
```

A variável myBrushColor será do tipo TColor.

8. De seguida vamos mudar o valor desta variável com as teclas de cursor pressionadas. Para isso vamos programar o *handler* do evento *OnKeyDown* do **Form1**:

```
procedure TForm1.FormKeyDown(Sender: TObject; var Key: word; Shift: TShiftState);
begin
  case Key of
    VK_UP: myBrushColor := RGBToColor(150, 0, 0);
    VK_DOWN: myBrushColor := RGBToColor(0, 100, 150);
    VK_LEFT: myBrushColor := RGBToColor(130, 200, 0);
    VK_RIGHT: myBrushColor := RGBToColor(255, 130, 0);
  end;
  PaintBox1.Invalidate; //Forces the Paint event on PaintBox1 redrawing the board
end;
```

**Nota: incluir a biblioteca LCLType na lista de uses pois é lá que estão definidas as constantes relativas às teclas: VK\_UP, VK\_DOWN,...**

9. Agora, temos de alterar o código do *handler* do evento *OnPaint* da TPaintBox para utilizar a variável em vez da cor clYellow como está agora:

```
// Draws squares
Bitmap.Canvas.Pen.Color := clBlack; //Line Color
Bitmap.Canvas.Brush.Color := myBrushColor;
```

10. No entanto quando o programa inicia, a variável myBrushColor não tem valor atribuído. Isto é um problema pois o valor dela será zero, o que numa cor se traduz em preto.

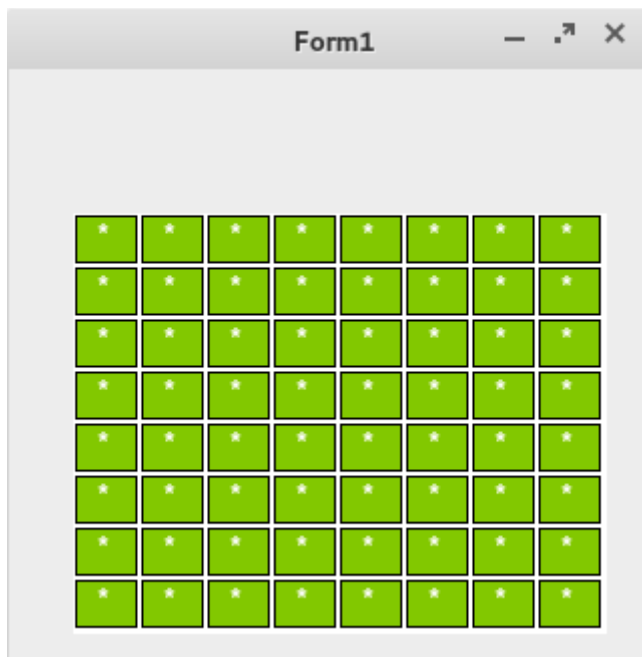
Assim, temos de inicializar a variável na altura em que o form é criado e quando o evento *OnPaint* da TpaintBox ainda não ocorreu. Assim, programe o *handler* do evento *OnCreate* do **Form1** e acrescente o seguinte código:

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin  
  myBrushColor := RGBToColor(130, 200, 0);  
end;
```

11. Volte a compilar o programa e agora teste pressionar as teclas de cursor esquerda, direita, cima, baixo e verifique que a cor do retângulo vai mudando.

12. Altere o programa de modo a que toda a PaintBox fique repleta de retângulos (8 linhas e 8 colunas)!



**Nota:**

Deixe dois pixels de distância entre os retângulos.