Funciones equivalentes de C++ a paccal

|  |  |
| --- | --- |
| C ++ | PASCAL |
| static const char \*processText[] = {  "NO PROCESSING",  "COLOR GRAYSCALE",  "COLOR TINT",  "COLOR INVERT",  "COLOR CONTRAST",  "COLOR BRIGHTNESS",  "FLIP VERTICAL",  "FLIP HORIZONTAL"  }; | var  .  .  processText: array [1..NUM\_PROCESSES] of String = (  'NO PROCESSING', 'COLOR GRAYSCALE',  'COLOR TINT', 'COLOR INVERT', 'COLOR CONTRAST',  'COLOR BRIGHTNESS', 'FLIP VERTICAL', 'FLIP HORIZONTAL' ); |
|  | type  TtoggleRecs = array [1..NUM\_PROCESSES] of TRectangle;  Var  toggleRecs :TtoggleRecs; |
| for (int i = 0; i < NUM\_PROCESSES; i++)  {  .  .  .  } | for i := 0 to NUM\_PROCESSES do  begin  .  .  end; |
| if ( (int a=0 ))  {  mouseHoverRec = i;  .  .  break;  }  else mouseHoverRec = -1; | if (a=0 ) then  begin  mouseHoverRec := i;  .  .  break;  end  else mouseHoverRec := -1; |
| if (IsMouseButtonDown(MOUSE\_BUTTON\_RIGHT))  DrawCircleLines((int)mousePos.x, (int)mousePos.y, brushSize, GRAY);  else DrawCircle(GetMouseX(), GetMouseY(), brushSize, colors[colorSelected]); | if (IsMouseButtonDown(MOUSE\_BUTTON\_RIGHT)) then DrawCircleLines(round(mousePos.x), round(mousePos.y), brushSize, GRAY)  else DrawCircle(GetMouseX(), GetMouseY(), brushSize, colors[colorSelected]); |
| break; | break; |
| currentProcess++; | currentProcess := currentProcess + 1 ; Inc( ) |
| DrawRectangle(0, 0, screenWidth, screenUpperLimit, collision? RED : BLACK); | if collision then clr:= RED else clr:= BLACK;  DrawRectangle(0, 0, screenWidth, screenUpperLimit, clr); |
| switch (currentProcess)  {  case COLOR\_GRAYSCALE: ImageColorGrayscale(&imCopy); break;  case COLOR\_TINT: ImageColorTint(&imCopy, GREEN); break;  case COLOR\_INVERT: ImageColorInvert(&imCopy); break;  case COLOR\_CONTRAST: ImageColorContrast(&imCopy, -40); break;  case COLOR\_BRIGHTNESS: ImageColorBrightness(&imCopy, -80); break;  case FLIP\_VERTICAL: ImageFlipVertical(&imCopy); break;  case FLIP\_HORIZONTAL: ImageFlipHorizontal(&imCopy); break;  default: break;  } | Case currentProcess of  CASE cdata of  'I': Label1.Caption:='1';  'V': Label1.Caption:='5';  'X': Label1.Caption:='10';  'L': Label1.Caption:='50';  'C': Label1.Caption:='100';  'D': Label1.Caption:='500';  'M': Label1.Caption:='1000';  END; |
| while (key > 0)  {  // NOTE: Only allow keys in range [32..125]  if ((key >= 32) && (key <= 125) && (letterCount < MAX\_INPUT\_CHARS))  {  name[letterCount] = (char)key;  name[letterCount+1] = '\0';  letterCount++;  }  key = GetCharPressed();  } | while (key > 0) do  begin  // NOTE: Only allow keys in range [32..125]  if ((key >= 32) and (key <= 125) and (letterCount < MAX\_INPUT\_CHARS)) then  begin  name[letterCount] := key;  name[letterCount+1] = Null;  Inc(letterCount);  end;  key = GetCharPressed();  end; |
| // Check if any key is pressed  // NOTE: We limit keys check to keys between 32 (KEY\_SPACE) and 126  bool IsAnyKeyPressed()  {  bool keyPressed = false;  int key = GetKeyPressed();  if ((key >= 32) && (key <= 126)) keyPressed = true;  return keyPressed;  } | // Check if any key is pressed  // NOTE: We limit keys check to keys between 32 (KEY\_SPACE) and 126  function IsAnyKeyPressed():Boolean;  var  keyPressed:Boolean;  key:Integer;  begin  keyPressed := false;  key := GetKeyPressed();  if ((key >= 32) and (key <= 126)) then keyPressed := true;  Result:=keyPressed;  end; |
| include "raylib.h"  // Draw text using font inside rectangle limits  **static void** DrawTextBoxed(Font font, const char \*text, Rectangle rec, float fontSize, float spacing, bool wordWrap, Color tint);  {  } | // Draw text using font inside rectangle limits  **procedure** DrawTextBoxed(font: TFont; const text:PChar;rec: TRectangle; fontSize, spacing:Single; wordWrap: Boolean; tint: TColor );  **Begin**  **end**; |
| width := (width > minWidth)? ((width < maxWidth)? width : maxWidth) : minWidth; | if (width > minWidth) then  begin  if (width < maxWidth) then container.width := width  else container.width := maxWidth;  end  else container.width := minWidth; |
| state := wordWrap? MEASURE\_STATE : DRAW\_STATE;  isGlyphSelected? selectTint : tint | if wordWrap then state := MEASURE\_STATE else state := DRAW\_STATE; |
| enum { MEASURE\_STATE = 0, DRAW\_STATE = 1 }; | const  MEASURE\_STATE = 0; DRAW\_STATE = 1; |
| for (int i = 0, k = 0; i < length; i++, k++)  {  } | k:=0;  for i := 0 to length-1 do  begin  k:=k+1;  end; |
| const char text[] = "Text cannot escape\tthis container\t...word wrap also works when active so\ here's a long text for testing.\n\nLorem ipsum dolor\  sit amet, consectetur adipiscing elit, sed do eiusmod \  tempor incididunt ut labore et dolore magna aliqua."; | Text:string;  ..  Text:= ‘Text cannot escape’+chr(9)+’this container ...word wrap also works when active so’+’ here's a long text for testing’+chr(10)+chr(10)+’ Lorem ipsum dolor’+  ‘sit amet, consectetur adipiscing elit, sed do eiusmod ‘+  ‘tempor incididunt ut labore et dolore magna aliqua.’; |
| (Int) mousePos.x  Conversion a entero | round(mousePos.x) |
| int i;  int (i); // exact same  So in your case:  int a(1); // initialized with 1  int (a)(1); // exact same |  |
|  |  |
|  |  |

Notas:

El bucle **For .. to .. do** en pascal (lazarus free) las variables que utiliza no se puede modificar dentro del bucle como sucede en otros lenguages (basic, c++).