



I Wish I Knew How To ...

*Program the Canvas Control
with Xojo Desktop*

July 2014 Edition (1.0)

Written By Eugene Dakin
Contributions by Alain Bailleul

About the Author/Contributor

Eugene Dakin is an author, Professional Chemist, oilfield consultant and programmer in Canada. When he is in-between courses and oilfield projects, he and his wife spend valuable time raising their wonderful children. He has been hobby programming on computer's for over 25 years. He has created over 185 technical and industrial simulation programs.

Eugene can be contacted via:

E-mail: eugenedakin@hotmail.com

Website: <http://www.scispec.com>

Dedication

Alain Bailleul is a contributor to this book and would like to thank his parents Gilbert and Irene. At a very young age, they gave him his first computer and stimulated him to push his boundaries ever since.

Website: <http://alwaysbusycorner.wordpress.com/>

Eugene dedicates this book to his patient family. Without their support this would not have been written.

Assumptions

To be able to use the code in this book, you need a version of Xojo. These test programs were created and tested on Windows 8.1 and Maverick. Code for this book was written in my licensed version of Xojo Pro 2014 R2.

Notice

This information is intended for Xojo programmers to learn how to reasonably use the Canvas control. This is not intended to be a complete reference for Xojo or the Canvas control.

Table of Contents

Chapter 1 - Introduction to the Canvas	10
What a Canvas IS and IS NOT	11
Raster vs Vector Drawing	12
Canvas Fundamentals	13
Drawing a Line.....	14
Constructors and Destructors	15
Buffers Explained.....	16
Single Buffer	16
Double-Buffer	16
Blanking Interval.....	17
Fundamental Shapes.....	18
Fill Shapes with Colour	22
Colouring in Hexadecimal	25
Pens	28
Pixel	30
Convert Colour Values.....	31
Graphics Object.....	35
Performance.....	36
Chapter 2 - Files	38
Loading a Graphics Picture.....	38
Drag and Drop Picture File	41
Open Xojo Picture	43
Saving A Picture Backdrop.....	45
Saving File Formats.....	47
Saving Canvas Graphics Layer	48
Setup Picture Graphics (VERY IMPORTANT – Double Buffer).....	48
Saving Picture and Graphics Layer	51
Chapter 3 – Text.....	54
Draw String.....	54
Rotating Text	57
Wrap Text	60
Print Text	62
Scroll Bar.....	65
Zooming Text (Picture).....	71
Horizontal Centred Text	74
Zooming Text (Font Size).....	77
Chapter 4 – Use the Mouse	82

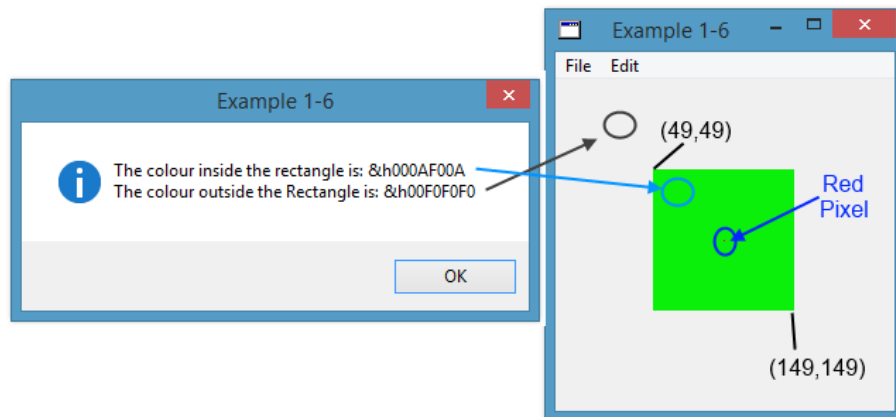
Freehand Drawing with the Mouse	83
Move Picture with Mouse.....	86
Mouse Zoom	89
Chapter 5 – Canvas Class	92
Begin a Canvas Class.....	92
Things that Happen when using a Class.....	96
Making an Event.....	97
Chapter 6 – Charts	98
Line Chart (Introduction).....	98
Line Chart (Intermediate).....	103
Line Chart (Customize)	110
Bar Chart.....	116
Pie Chart	124
Chapter 7 – Objects.....	129
First Object Program	129
Getting Ready to use the Object	138
Mouse Moving the Objects.....	142
Make a Simple Game	151
Multiple Object Types (Object 2D).....	167
Chapter 8 – Graphics.....	175
Picture Resizing	175
Crop a Picture (Pixel).....	179
Rotate a Picture.....	185
Flipping Horizontal and/or Vertical-Pixel Slower	190
Flipping Horizontal and/or Vertical- Row/Column Fast.....	197
Blur (RGBSurface – IMPORTANT)	204
Sharpening Filter	216
Emboss Filter	225
Edge Detection	231
Picture with Mirror Image (Vertical).....	237
Posterization	242
Pixelation.....	248
Grey Scale.....	253
Merging Two Images.....	258
Gaussian Blur.....	264
Gaussian Sharpening.....	269
Crop a Picture (DrawPicture)	275
Chapter 9 - Masks and Alpha Channel.....	282
Implementing a Mask.....	283

GreyScale Mask	288
Timer Fade.....	294
Alpha Channel	302
Chapter 10 – Building Controls	308
Canvas Button – Simple.....	308
Canvas Button – Detail	312
Canvas Switch.....	321
Chapter 11 – Animation	327
Creating Static Template.....	328
Moving the Picture with Keyboard Keys.....	332
Boundary Bounce	335
Sprite Sheet Animation	339
Creating/Destroying an Object	344
Open Event	346
Create Objects with PushButton	349
Delete Object with PushButton.....	350
Drag and Drop Deletion.....	351
Chapter 12 – Games.....	358
Puzzle.....	358
Open Event	360
Pressing Pushbutton.....	364
MouseDown	366
SpaceShip Shooter.....	372
Trigonometry.....	372
Ship Setup Basics (12-2a).....	374
Ship Movement Basics.....	379
Setting Up Frameworks (12-2b)	381
Event Handling (12-2c)	386
Adding a Ship (12-2d)	390
Ship Shooting (12-2e)	398
Building Asteroids (12-2F)	403
Collision (Final – 12-2g)	407
Index.....	415

Pixel

A *Canvas* is made of many individual dots or pixels, and when a canvas has a width of 200 by 200, that means that there are 200 pixels in the X direction, and 200 pixels in the Y direction. Because *Canvas* pixels are zero-based, this means that the first of 200 pixels starts at zero (0), and the 200th pixel ends at 199. The pixel method in the graphics layer of the canvas can get or set the colour of the pixel.

Figure 9. Example 1-6: Screen Grab



The above screen grab shows the results of the following code.

Code 23. Example 1-6: Pixel Code

```
'Draw a green filled rectangle
g.ForeColor = RGB(10, 240, 10) 'Green
g.FillRect(49, 49, 100, 100) 'Filled Rectangle

'Colour the centre pixel red
g.Pixel (99, 99) = RGB(240, 10, 10)

'Get the pixel values outside and inside the Rectangle
MsgBox "The colour inside the rectangle is: " + CStr(g.Pixel(60,60)) + chr(13) + _
"The colour outside the Rectangle is: " + CStr(g.Pixel(20,20))
```

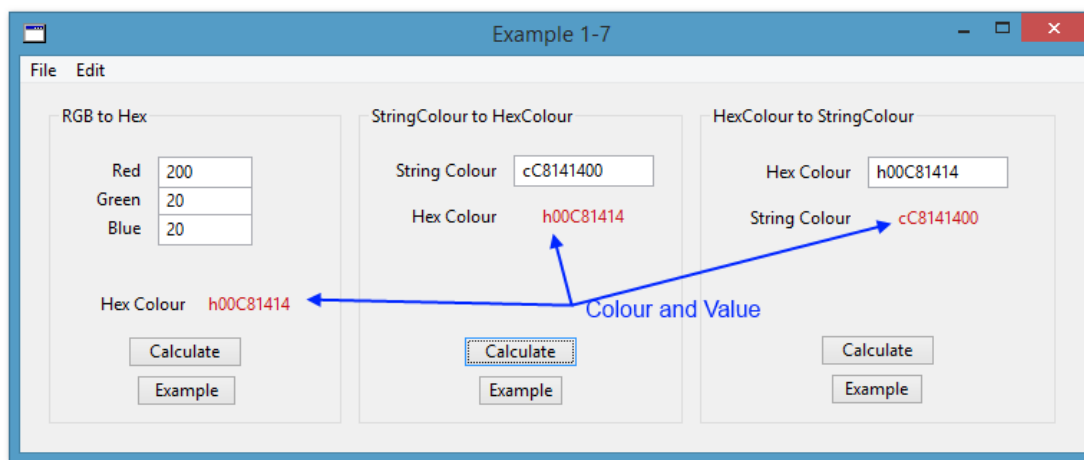
The top layer of the *Canvas* (*ForeColor*) is set to a green colour with the RGB values of 10 red, 240 green, and 10 blue. A filled rectangle is drawn with the upper-left hand coordinates of 49 and 49, with a width and height of 100 x 100. The filled rectangle is drawn from the pixel set 49,49 to 149,149. To draw the pixel at the very centre of the rectangle a red colour, the pixel

set 99,99 is chosen (reminder: a 200x200 rectangle is zero-based and has dimensions from 0-199). The pixel is coloured to red with the RGB value of 240-Red, 10-Green, and 10-Blue. The next code shows the hexadecimal colour value for a pixel which is inside the rectangle and has a green colour with the value &h000AF00A and is shown in a message box. The colour outside the rectangle is &h00F0F0F0.

Convert Colour Values

The previous example gets and uses the hexadecimal value of the string. There are three beginning ways to have a colour represented, RGB, String Colour, and Hex Colour. Hexadecimal colour was introduced in the previous example and will be added in this example for thoroughness of the topic.

Figure 10. Example 1-7: Converting Colours



This example has three sections: 1) RGB to Hex, 2) String Colour to Hex Colour, and 3) Hex Colour to String Colour. Programming with the Canvas can create a need to understand this conversion fundamental.

The first example converts the RGB value to hexadecimal colour and changes the text of the Hex Colour label to the RGB colour. In this case the h00C81414 value is the colour red and the text colour is changed to this shade of red. The hexadecimal colour is the standardized way in which textcolors are changed in Xojo. When the RGB value of red, green, and blue are converted to a value, the value is hexadecimal and this value is entered in the textcolor for LbHex.

Index

- #If, 90
- #If...#Elseif
 - DoubleBuffer, 90
 - EraseBackground, 90
 - TargetLinux, 90, 94
 - TargetMacOS, 90, 94
 - TargetWin32, 36, 60, 90, 94
 - Transparent, 90
- &, 34
- Abs, 380
- Absolute, 380
- AcceptFileDrop, 42
- Add Class, 135
- AddLine, 173
- Alpha Channel, 33, 282, 302
- Animation, 327
 - Create Object, 344
 - Destroy Object, 344
- Animation Sprite Sheet, 339
- Animation Template, 328
- Append, 347, 353
- Append Class, 146
- ArcAngle, 172
 - FillColor, 127
 - PieChart, 127
 - Scale, 127
 - StartAngle, 127
- ArcShape, 167, 172
- Array
 - Append, 347, 353
 - Remove, 350, 353
- Array Shuffle, 365
- AsyncKeyDown, 397, 399
- Backdrop, 42
- Backdrop Layer, 11
- Bar Chart, 116
- Begin Canvas Class, 92
- Behavior Properties, 314
- Bezier, 35, 173
- Bicycle Movement, 332
- Blur, 204
- Blur Gaussian, 264
- Blurring, 204
- Bold, 55, 174
- Border, 171
- BorderColor, 171
- BorderWidth, 171
- Bounce Boundary, 335
- Boundary Bounce, 335
- Buffers
 - Blanking Interval, 17
 - Double Buffer, 16
 - Explained, 16
 - Single Buffer, 16
 - Triple Buffer, 17
- Building Controls, 308
- ByRef, 223
- Byte, 34
- Canny Edge Detection, 231
- Canvas Button - Detail, 312
- Canvas Button - Simple, 308
- Canvas Class, 92
- Canvas Coordinates, 10
- Canvas Fundamentals, 13
- Canvas Switch, 321
- Case, 324
- Centre Text, 74
- Channel
 - Alpha, 302
 - Blue, 302
 - Green, 302
 - Red, 302
- Chart Bar, 116

- Chart Line, 98
- Chart Pie, 124
- Charts, 98
- Class Canvas, 92
- Class Event Create, 97
- Class Events, 96
- Class Remove, 136
- CleanMemory, 141
- Colour Channels, 33
- Colour Conversion, 31
- Colouring in Hexadecimal, 25
- Constructor, 15
- Controls Building, 308
- ControlX, 173
- ControlY, 173
- Convert Colours, 31
- Convert Integer, 292
- Coordinates, 10
- CornerHeight, 172
- CornerWidth, 172
- Cos, 374
- Create Class Event, 97
- Crop Picture, 179, 275
- CStr, 26
- CType, 292
- Cursors, 147
 - FingerPointer, 147
 - StandardPointer, 147
- CurveShape, 35, 167, 173
- Customized Line Chart, 110
- Destructor, 15
- Detect Edge, 231
- Dialog Title, 40
- Double Buffer, 16, 48
- DoubleBuffer, 36, 54, 94
- Drag and Drop Picture File, 41
- Draw String, 54
- Drawing a Line, 14
- Drawing Shapes, 18
- drawinto, 46
- DrawLine, 14, 19
- DrawNotelcon, 21
- DrawObject, 171
- DrawOval, 19
- DrawPicture, 178
- DrawPolygon, 19
- DrawRect, 20
- DrawRoundRect, 20
- DrawStoplcon, 21
- DrawString, 21
 - Bold, 55
 - ForeColor, 55
 - Italic, 55
 - Text, 55
 - TextAscent, 55
 - TextFont, 55
 - TextHeight, 56
 - TextSize, 56
 - Underline, 56
- Edge Detection, 231
- Embossed Filter, 225
- EndTime, 383
- EraseBackground, 36, 54
- Examples
 - 01-01 Draw a Line, 14
 - 01-01a Constructor, 15
 - 01-01b Destructor, 15
 - 01-02 Fundamental Shapes, 18
 - 01-03 Filling Shapes, 22
 - 01-04 Open, 26
 - 01-04 Paint, 26
 - 01-05 Pen Width and Height, 28
 - 01-06 Pixel, 30
 - 01-07 Hex sto String Colour Calculate, 34
 - 01-07 RGB to Hex, 32
 - 01-07 RGB to Hex Calculate, 32
 - 01-07 String to Hex Calculate, 33

- 01-08 Bezier Curve, 35
- 02-01 Load File Pushbutton, 40
- 02-02 Drag Drop Picture, 42
- 02-03 IDE File Load, 44
- 02-04 Save Event, 46
- 02-05 Save Picture Graphics, 50
- 02-06 Picture and Graphics, 53
- 03-01 DrawText, 55
- 03-02 StringShape, 58
- 03-03 Wrap Text, 61
- 03-04 Print Text, 63
- 03-05 Draw Text Scroll Bar, 70
- 03-05 Open Scroll, 69
- 03-06 Paint Magnification, 73
- 03-07 Centre Text Pushbutton, 75
- 03-08 DrawTheText Method, 79
- 04-01 Clear Canvas, 84
- 04-01 MouseDown Drawing, 83
- 04-01 MouseDrag Drawing, 83
- 04-02 MouseDown Move, 86
- 04-02 MouseDrag Move, 87
- 04-03 MouseWheel Event, 89
- 05-01 Canvas Class Open, 94
- 05-01 Canvas Class Paint, 93
- 06-01 Chart Intro Class Open, 100
- 06-01 DrawLineChart Open, 101
- 06-02 DrawLineChart Method, 107
- 06-03 Draw Chart Action Event, 115
- 06-04 DrawBarChart Method, 119
- 06-05 DrawPieChart Method, 127
- 07-01 DrawObjects, 134
- 07-01 LoadClassObjects Method, 140
- 07-02 BringToFront Method, 146
- 07-02 CustomCanvas MouseDown Method, 147
- 07-03 LoadNineCards Method, 157
- 07-03 MouseUp Snap Method, 165
- 07-04 PixMapShape Method, 174
- 07-04 ShowArcShape Method, 172
- 07-04 ShowCurveShape Method, 173
- 07-04 ShowFigureShape Method, 173
- 07-04 ShowOvalShape Method, 171
- 07-04 ShowRectShape Method, 171
- 07-04 ShowRoundRectShape Method, 172
- 07-04 ShowString Method, 174
- 08-01 DrawPicture Resize, 178
- 08-02 Picture Crop Pixel Method, 183
- 08-03 Rotate Picture Method, 189
- 08-04 FlipHor Pixel Method, 194
- 08-04 FlipHorVert Pixel Method, 196
- 08-04 FlipVert Pixel Method, 195
- 08-05 FlipHor Width Method, 201
- 08-05 FlipHorVert Row/Column Method, 203
- 08-05 FlipVert Height Method, 202
- 08-06 Pixel Blur Slow Method, 210
- 08-06 Resize Blur Method, 213
- 08-06 RGBSurface Blur Method, 214
- 08-07 PixelSharpen Method, 223
- 08-07 Sharpen Method, 221
- 08-08 Emboss Method, 229
- 08-09 Edge Detect Method, 235
- 08-10 VMirror Method, 241
- 08-11 Posterize Method, 246
- 08-12 Pixelated Method, 251
- 08-13 GreyScale Method, 256
- 08-14 MergeTwo Method, 262
- 08-15 Gaussian Blur Fast Method, 268
- 08-16 Gaussian Sharpen Fast Method, 274
- 08-17 Picture Crop DrawPicture Method, 280
- 09-01 AddMask Method, 287
- 09-02 DrawGreyScaleMask Method, 293
- 09-03 Timer1 Action Fade, 299
- 09-04 DrawAlphaSquares Method, 306
- 10-01 CustomCanvas Simple Button, 309
- 10-02 CustomCanvas DrawABackground, 316
- 10-02 CustomCanvas DrawButtonWords, 317
- 10-03 CanvasSwitch DrawSwitch Method, 324
- 11-01 CanvasAnimation Template Open Event, 329

- 11-02 CanvasAnimation KeyDown Event, 332
- 11-03 CanvasAnimation StartBouncing Event, 337
- 11-04 CanvasAnimation LoadSprites Method, 341
- 11-04 CanvasAnimation ShowSprites Method, 342
- 11-05 AddObject, 349
- 11-05 PushButton Remove Object, 350
- 12-01 Puzzle SaveImageAs9Pieces, 362
- 12-01 Puzzle ShufflePieces, 364
- 12-01 ReversePieceWithBlank, 368
- 12-02a DrawShip Method, 377
- 12-02a Ship KeyDown Method, 379
- 12-2b FrameWork Thread, 382
- 12-2c KeyDown Event, 388
- 12-2d DrawShip Method, 392
- 12-2e DrawShots Method, 400
- 12-2f DrawAsteroids Method, 405
- 12-2g CheckCollisions Method, 409
- Fast Flip Picture, 197
- FigureShape, 167, 173
- Files, 38
- FileTypes, 42, 46
- FillColor, 171
- Filling Shapes, 22
- FillOval, 23
- FillPolygon, 23
- FillRect, 24
- FillRoundRect, 24
- Filter, 46
- Flip Picture Fast, 197
- Flip Picture Pixels, 190
- FolderItem, 40, 42, 46
- FolderItemAvailable, 42
- Font Size Zoom, 77
- ForeColor, 55
- fps, 336
- FPS, 37, 373
- FramePeriod, 383
- frames per second, 336
- Frames Per Second, 37, 373
- Freehand Mouse, 83
- Games, 358
- Gaussian Blur, 264
- Gaussian Sharpening, 269
- Graphics, 175
- Graphics Device Interface, 36
- Graphics Layer, 11
- Graphics Object, 35
- GreyScale, 253
- GreyScale Mask, 288
- Height, 171
- Image Merge, 258
- Implement Mask, 283
- Infinite Loop, 387
- Inspector Behavior, 313, 314
- Integer Convert, 292
- Intermediate Chart, 103
- Introduction, 10
- Invalidate, 56
- Invalidate(False), 36
- Italic, 55
- JPEG
 - QualityDefault, 47
 - QualityHigh, 47
 - QualityLow, 47
 - QualityMax, 47
 - QualityMedium, 47
 - QualityMin, 47
- Key
 - Chr, 329, 332
 - Down Arrow, 332
 - Left Arrow, 332
 - Right Arrow, 332
 - Up Arrow, 332
- Keyboard, 397
- Kill Thread, 387

- Line Chart, 98
- Livescroll, 67
- Load Picture, 38
- Loop Infinite, 387
- Low-Pass Filter, 204
- Mask Fade, 294
- Mask Fundamentals, 282
- Mask Generate, 288
- Mask GreyScale, 288
- Mask Timer, 294
- Masks, 282
- Math Trigonometry, 372
- Max, 148
- Merge Two Images, 258
- MessageBox
 - Button, 356
 - CancelButton, 356
 - Caption, 356
 - GraphicCaution, 356
 - Show, 356
 - ShowModal, 356
- Microseconds, 383
- Mid, 33, 34
- Mirror Image Vertical, 237
- ModeOff, 387
- Module, 374
- Mouse, 82
 - MouseDown, 82
 - MouseDown, 82
 - MouseEnter, 82
 - MouseExit, 82
 - MouseMove, 82
 - MouseUp, 82
 - MouseWheel, 82
- Mouse Move Picture, 86
- Mouse Zoom, 89
- Move Bicycle, 332
- Moving Objects, 142
- Object Motion, 332
- Object Program, 129
- Object2D, 167
 - ArcShape, 167, 172
 - CurveShape, 167, 173
 - FigureShape, 167, 173
 - OvalShape, 167, 171
 - PixmapShape, 167, 174
 - RectShape, 167, 171
 - RoundRectShape, 167, 172
 - StringShape, 167, 174
- Objects, 129
 - X and O's, 151
- Objects Moving, 142
- Opacity, 294
- Open Picture, 40
- Open Xojo Picture, 43
- OpenAsPicture, 42
- OpenDialog, 40
- OpenPrinterDialog, 64
- Order, 173
- OvalShape, 167, 171
- PageSetupDialogue, 64
- PenHeight, 28
- Pens, 28
- PenWidth, 28
- Performance, 36
- Picture
 - SaveAsGIF, 47
 - SaveAsJPEG, 47
 - SaveAsMacintoshPICT, 47
 - SaveAsPNG, 47
 - SaveAsRasterPICT, 47
 - SaveAsTIFF, 47
 - SaveAsWindowsBMP, 47
 - SaveAsWindowsEMF, 47
 - SaveAsWindowsWMF, 47
- Picture Load, 38
- Picture Mouse Movement, 86
- Picture Objects, 129
- Picture Resizing, 175

Picture.Open, 40
PictureAvailable, 42
Pie Chart, 124
Pixel, 30, 196
Pixel Flip Picture, 190
Pixelation, 248
PixmapShape
 Method, 189
 New, 189
 Rotation, 189
PixMapShape, 167, 174
PNG, 38, 303
Portable Network Graphics, 303
Portable Networks Graphics, 38
Posterization, 242
Print Text, 62
PrinterSetup, 64
Puzzle, 358
QualityDefault, 47
QualityHigh, 47
QualityLow, 47
QualityMax, 47
QualityMedium, 47
QualityMin, 47
Quit, 387
Radian, 57, 373
RaiseEvent, 319, 326
Random Number, 349
Raster, 12
RectShape, 167, 171
ReDim, 137, 141
Refresh, 56
Refresh(False), 36
Remove, 136, 350, 353, 403
Remove Class, 136, 146
RGBSurface, 214
Rnd, 349
Rotate Picture, 185
Rotating Text, 57
Rotation, 171, 174
Round, 148
RoundRectShape, 167, 172
Save a Backdrop Picture, 45
Save Picture and Graphics, 51
SaveAsDialog, 46
SaveAsGIF, 47
SaveAsJPEG, 47
SaveAsMacintoshPICT, 47
SaveAsPNG, 46, 47
SaveAsRasterPICT, 47
SaveAsTIFF, 47
SaveAsWindowsBMP, 47
SaveAsWindowsEMF, 47
SaveAsWindowsWMF, 47
Saving Canvas Graphics Layer, 48
Saving Picture Graphics, 48
Scroll Bar, 65
ScrollBar
 Horizontal, 67, 68
 Horizontal, 69
 Livescroll, 67
 LiveScroll, 69
 Vertical, 67, 68
 Vertical, 69
Select Case, 324
Shape Colour, 22
Shapes
 DrawLine, 19
 DrawNotelcon, 21
 DrawOval, 19
 DrawPolygon, 19
 DrawRect, 20
 DrawRoundRect, 20
 DrawStoplcon, 21
 DrawString, 21
 FillOval, 23
 FillPolygon, 23

FillRect, 24
FillRoundRect, 24
Sharpening Filter, 216
Sharpening Gaussian, 269
Ship Movement Basics, 379
Ship Setup Basics, 374
ShowModal, 40, 46
Shuffle, 365
Simple Canvas Button, 308
Sin, 374
Single Buffer, 16
SpaceShip
 Asteroids, 403
 Event Handling, 386
 Framework, 381
 Ship, 390
 Shooting, 398
SpaceShip Shooter, 372
Sprite, 327
Sprite Sheet, 339
StartAngle, 172
StartTime, 383
Static Template, 328
StringHeight, 317
StringShape, 57, 58, 120, 167, 174
 Bold, 58, 60
 Border, 60
 Rotation, 59
 Text, 59
 TextFont, 58
 TextSize, 60
 Underline, 60
 X, 58
 Y, 58
StringWidth, 75, 317
SuggestedFileName, 46
Switch Canvas, 321
TargetLinux, 90
TargetMacOS, 90
TargetWin32, 36, 90
Text, 54, 174
Text Centre, 74
Text FillColor, 120
Text Rotate, 57
Text Rotation, 120
Text Zoom, 71
TextAscent, 55
TextFont, 55, 174
TextHeight, 56
TextSize, 56
Thread, 382
 Sleep, 383
ThreadEndException, 383
Timer, 336, 382
 Enabled, 299, 300
 mode, 336
 Period, 299, 336
 Timer1, 299
Timer Fade, 294
Timer Mask, 294
Timer ModeOff, 387
Timer off, 387
Title, 46
Title Dialog, 40
Transparency, 302
Transparent, 94, 282
Trigonometry, 372
Triple Buffer, 17
UBound, 141, 347
Underline, 56
Use the Mouse, 82
UseGDIPlus, 36, 60, 94
Vector, 12
While-Wend, 386
Width, 171
Windows Alpha, 33
Windows GDIPlus, 33

Wrap Text, 60
X2, 173
Y2, 173

Zoom Mouse, 89
Zooming Text, 71
Zooming Text (Font Size), 77

The 'I Wish I Knew' series contains technical data and advice that makes sense and contains practical and numerous examples with explanations to allow you to ease into the steep programming curve. You can create custom canvas Desktop applications today!

This book "I Wish I Knew How to ... Program Xojo's Canvas Control" starts with the fundamentals of the Canvas on the Desktop and builds your knowledge in each chapter. It is assumed that the reader has a good fundamental understanding of programming with Xojo before reading this book, as this starts at an intermediate level. The basics of many topics are discussed to give an overview of fundamental concepts. Each of these topics can then be pieced together to create your own gaming, picture editing, or animation Michaelangelo masterpiece!

The book is written as a guide and reference to Xojo programmers who program Desktop Applications in Windows, Mac, and Linux (Ubuntu). There are no dynamic link libraries (dll), COM, or Active X parts to add. This code was tested with Xojo 2014 R2 on Windows, OS X, and Ubuntu machines.

There are 12 chapters and over 400 pages with over 60 example programs.

Examples include two games, animation, cropping, blurs, edge detection, pixilation and more. Many screenshots have been added to show the results of the code with an index to help find topics quickly.

This is one of many books at Great White Software. This book can be purchased at <http://great-white-software.com/rlibrary/> where many great Xojo resources are available.

Happy programming!

Eugene

Alain Bailleul, is a contributor to this book and is the Senior Software Architect at One-Two. He likes to experiment with new technologies like the Kinect, Computer Vision, and Artificial Intelligence.

Eugene Dakin MBA, Ph.D., P.Chem., is an author of Xojo and Real Studio reference materials and has many years of experience in the programming industry.

ISBN: 978-1-927924-06-8