## VBugs

## Chapter 2

###### Images, Fonts and Colours



# Summary:

For this chapter you will be creating a new project. You will be using .png images and true type font. All of necessary resources are provided. You also will be introduced to ARGB color representation.

#### GetAttachment.jpg

#### GetAttachment1.jpg

#### GetAttachment2.jpg

#### GetAttachment3.jpg

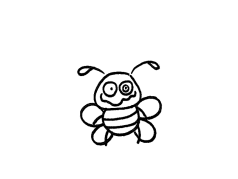
## Part 1

### Importing the Images

#### Open Visual Studio. Create a new SwinGame project and name it “Images”.

In order to have a custom picture in you program, we should, firstly, provide the project with the necessary files. Follow the steps below to provide all necessary files to your project:

1. Open the “Resources” folder. Copy the following images:



back.png

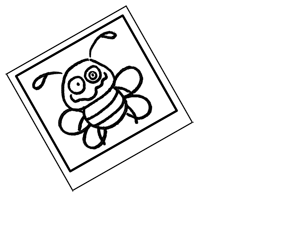


photo.png

1. Paste these images into your image resources folder for the Hello World program, which is usually in My Documents -> Visual Studio 2008 -> Projects -> Images -> Images -> Resources
2. Open GameLogic.vb and delete the code indicated in Figure 1 below:

|  |
| --- |
| Do  'Clears the Screen to Black  SwinGame.Graphics.ClearScreen()  'Draws red rectangle  Code to Graphics.FillRectangle(Color.Red, 20, 150, 500, 50)  delete  'Draws text "Hello World"  Text.DrawText("Hello World!", Color.Aqua, GameFont("ArialLarge"), 50, 50)  'Refreshes the Screen and Processes Input Events  Core.RefreshScreen()  Core.ProcessEvents()  Loop Until SwinGame.Core.WindowCloseRequested() = True |

Figure 1

1. Press the "StartDebugging" button arrow.jpg to see what removing that code does then close the window.

cha 2 - worksheet.png *Question 1: What is happening on your screen? Answer on the worksheet.*

### Loading the Images into Your Program

**NOTE:** We provide you with all necessary images for these exercises. You may want to create your own images if so you must make them as 32 bit PNG images. You can do this in a program like Macromedia Fireworks. On top of this you must consider the size of the screen in SwinGame. It is 800x600 pixels, so picture’s size has to be not more than the size of the screen.

Once we have all necessary information in our project resources folder, we have to load our images into the program in order to use them.

Follow the steps below to load the images:

1. From Solution Explorer open GameResources.vb.
2. Scroll down to the LoadImages() sub.
3. Inside the LoadImages() sub we need to add the details on the two pictures we want to add to the program using the following format: NewImage(“Name of Image”, “Name of file”) where the “Name of Image” is what we will use to refer to this picture in the program itself and filename is the name of the actual file. The code for doing adding the first image is below see if you can figure out what the code for the second image will be, we want to call it “photo”.

|  |
| --- |
| …  NewImage("back", "back.png")    … |

Exercise 1: *Loading images into your program*

cha 2 - worksheet.pngAnswer the following questions on your worksheet:

1. Write the complete code for LoadImages() sub into a worksheet.
2. What does the first parameter (“back”) do in NewImage()? (write your answer on the worksheet)

Press the "StartDebugging" button arrow.jpg to see what that code does then close the window.

#### GetAttachment4.jpg

#### GetAttachment5.jpg

#### GetAttachment6.jpg

## Part 2

### Drawing the Image to the Screen

Now, we can start use our images. Let’s start with drawing the background on the screen. To do so, follow the steps below:

1. Open GameLogic.vb from Solution Explorer.
2. Put the following code inside the game loop, right after ClearScreen statement in the code:

|  |
| --- |
| …  Graphics.DrawBitmap(GameImage("back"), 0, 0)  … |

1. Press the "StartDebugging" button at the top of the screen (it looks like a green arrow arrow.jpg, F5 works too) to see changes.

cha 2 - worksheet.png Question 1: *What do you think DrawBitmap() sub does? Answer on the worksheet.*

**NOTE:** The back.png image size is 800x600 pixels which is the same as the size of the Swingame Screen. Also, spelling of the image name is case sensitive, make sure that you didn’t use any CAPITOL letters unless the name had it originally.

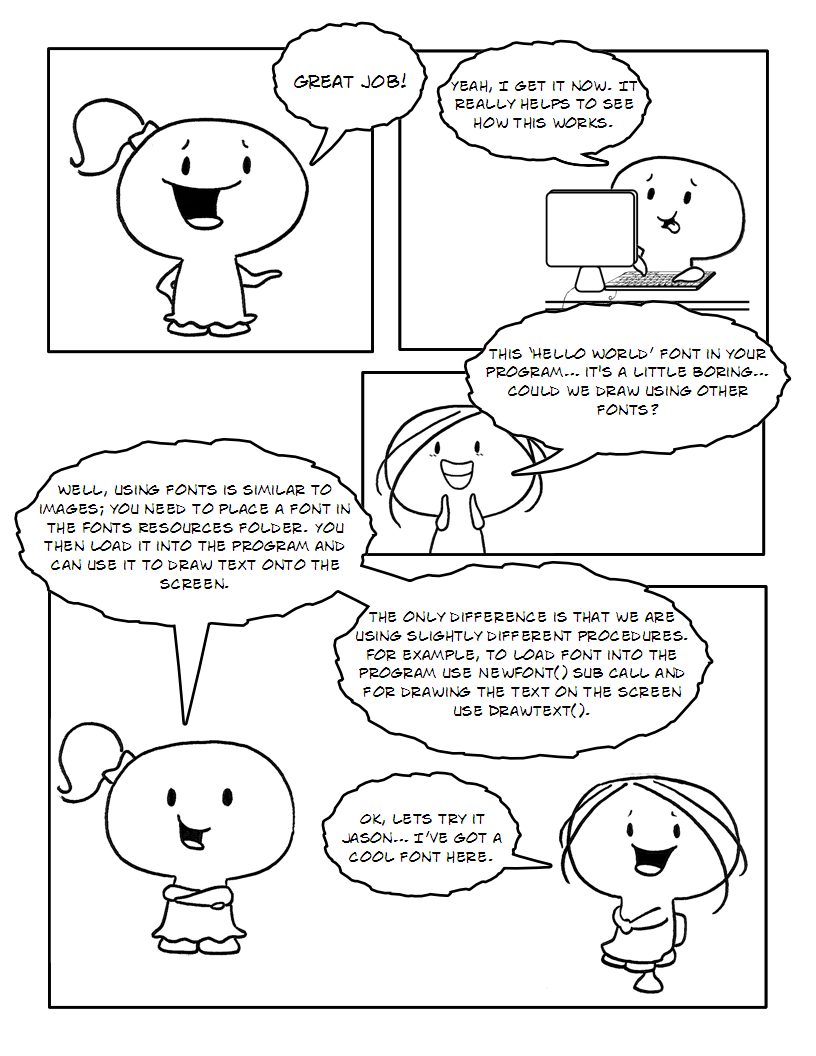
Exercise 1: *Drawing other images on the screen*

cha 2 - worksheet.pngMake the following changes to your program and write your solutions onto the worksheet:

1. Tell the computer to draw photo.png on the screen at a position of your choosing. The code for this needs to be inside the Game Loop.

*Hints:* In order to put another image on the screen, use DrawBitmap() as in previous exercise, but change the position of the image from (0, 0) to your custom one.

**NOTE:** You can put DrawBitmap() sub calls in certain order to have different results. For example, if you want to draw image2 on top of image1, you should use DrawBitmap(image1, posX, posY) before drawing image2. Try to play around to see the difference. Make sure you put them after the ClearScreen line though.

****

## Part 3

### Adding a New Font

In order to use another font in your program, you have to provide it with necessary files (fontname.ttf – true type font file). After that we have to load a new font into our program in order to use it just like you did for images.

Exercise 1: Adding a Font

Follow the steps below to load the font:

1. Copy the BEANTOWN.ttf font from the Resources and paste it into Fonts folder in your project (see exercise 1).
2. Open GameResources.vb, find the LoadFonts() sub and put the following code in it:

|  |
| --- |
| …  NewFont("BeanTown", "BEANTOWN.ttf", 60)  … |

cha 2 - worksheet.png *Write the complete code for the LoadFonts() sub into the worksheet.*

cha 2 - worksheet.png *Question 1: What do you think 60 in the above code does? Change its value and test it to see what find out.*

**NOTE:** We provide only one true type font for this exercise, if you want to test another one, you could download it from many different web sites like <http://www.1001freefonts.com/> for free.

Exercise 1: *Drawing the text by using loaded true type font*

cha 2 - worksheet.png Make the following changes in your program and write your solution onto the worksheet:

1. In GameLogic.vb you will remember earlier we deleted the following line of code:

|  |
| --- |
| …  Text.DrawText("Hello World!", Color.Aqua, GameFont("ArialLarge"), 50, 50)  … |

We want to use something similar to test our new font. Change the above code so that we use the new font and it says a message of your choosing a different position.

*Hint:* Also you need to consider the order of the elements in your program as you did for images.

#### GetAttachment8.jpg

#### GetAttachment9.jpg

#### GetAttachment10.jpg

## Part 4

### Manipulating our Colors

Now let’s practice our colors in the program.

Exercise 1: *Colors with RGB and ARGB*

cha 2 - worksheet.pngAnswer the following questions on your worksheet:

1. What is the color and how much could you see it with the following ARGB values:

(255, 255, 0, 0).

1. What is the color and how much could you see with the following ARGB values:

(30, 0, 255, 0)

1. What is the color and how much could you see with the following values of ARGB:

(127, 0, 0, 255)

1. What is the color and how much could you see with the following values of ARGB:

(255, 255, 255, 0)

1. What is the color and how much could you see with the following values of ARGB:

(0, 0, 200, 0)

Exercise 2: *Drawing rectangle with the customized color*

cha 2 - worksheet.png Make the following changes to your program and write your solutions onto the worksheet:

1. Draw one rectangle of each of the colors from the previous exercise, put the rectangles at different points on the screen.
2. In GameLogic.vb you will remember earlier we deleted the following line of code:

|  |
| --- |
| …  Graphics.FillRectangle(Color.Red, 20, 150, 500, 50)  … |

Modify this code to use ARGB, instead of Color.Red use Color.FromArgb(alphaVal, redVal, greenVal, blueVal).

saveicon.png Remember to save your project (File – Save All). Once you have finished then you can close Visual Studio or move on to the next chapter.