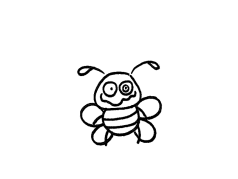
#### Part 1

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 on the CD. Copy the following images:



back.png

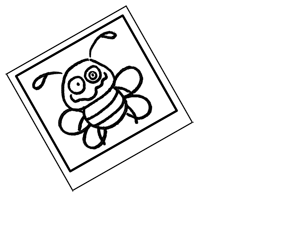


photo.png

1. Paste these images into your image resources folder for the Hello World program, which is in My Documents -> Visual Studio 2008 -> Projects -> HelloWorld -> HelloWorld -> Resources, i.e. C:\Documents and Settings\Name\My Documents\Visual Studio 2008\Projects\ HelloWorld\HelloWorld\Resources\Images.
2. Open your HelloWorld project in Visual Studio 2008.

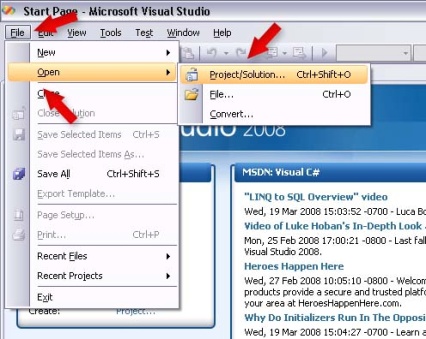
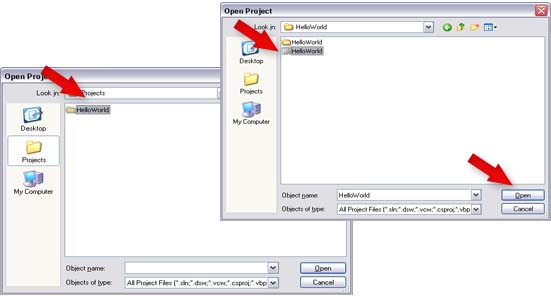
* Open Visual Studio 2008
* Click File -> Open -> Project/Solution as shown in Figure 1.
* Click “HelloWorld” -> “HelloWorld” -> Open Microsoft Visual Studio Solution as shown in Figure 2.

Figure 2

Figure 1

* Open GameLogic.vb and delete everything between SwinGame.Graphics.ClearScreen() and 'Refreshes the Screen and Processes Input Events (Figure 3).

|  |
| --- |
| 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 |

cha 2 - worksheet.png What is happening on your screen? Answer on the worksheet.

**NOTE:** We provide you with all necessary images for these exercises. If you want to create your own images, you must make them as 32 bit PNG images. Moreover, 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. Add the following code to LoadImages() sub:

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

Exercise 2: *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. Why do we need the first parameter in NewImage()? (write your answer on the worksheet)

#### Part 2

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 Do statement:

|  |
| --- |
| …  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 *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 spelled it correctly when you are referencing to it.

**Part 3**

Exercise 2: *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.

*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 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.

#### Part 4

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.

Follow the steps below to load the font:

1. Copy the BEANTOWN.ttf font from the Resources folder on the CD 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 completed LoadFonts() sub into the worksheet.*

**NOTE:** We provide only one true type font for this exercise, if you want to use another one, you could download it from <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 “*HelloWorld*” program and write your solution onto the worksheet:

1. Open GameLogic.vb from the Solution Explorer. Use Text.DrawText("Text to draw", Color.(Black), GameFont("BeanTown"), posX, posY) to write the code that tells the computer to draw text with a new font to the worksheet.

#### Part 4

Exercise 1: *Colors with RGB and RGBA*

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:

(0, 0, 0, 255)

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 above. Use Color.FromArgb(redVal, greenVal, blueVal, alphaVal)