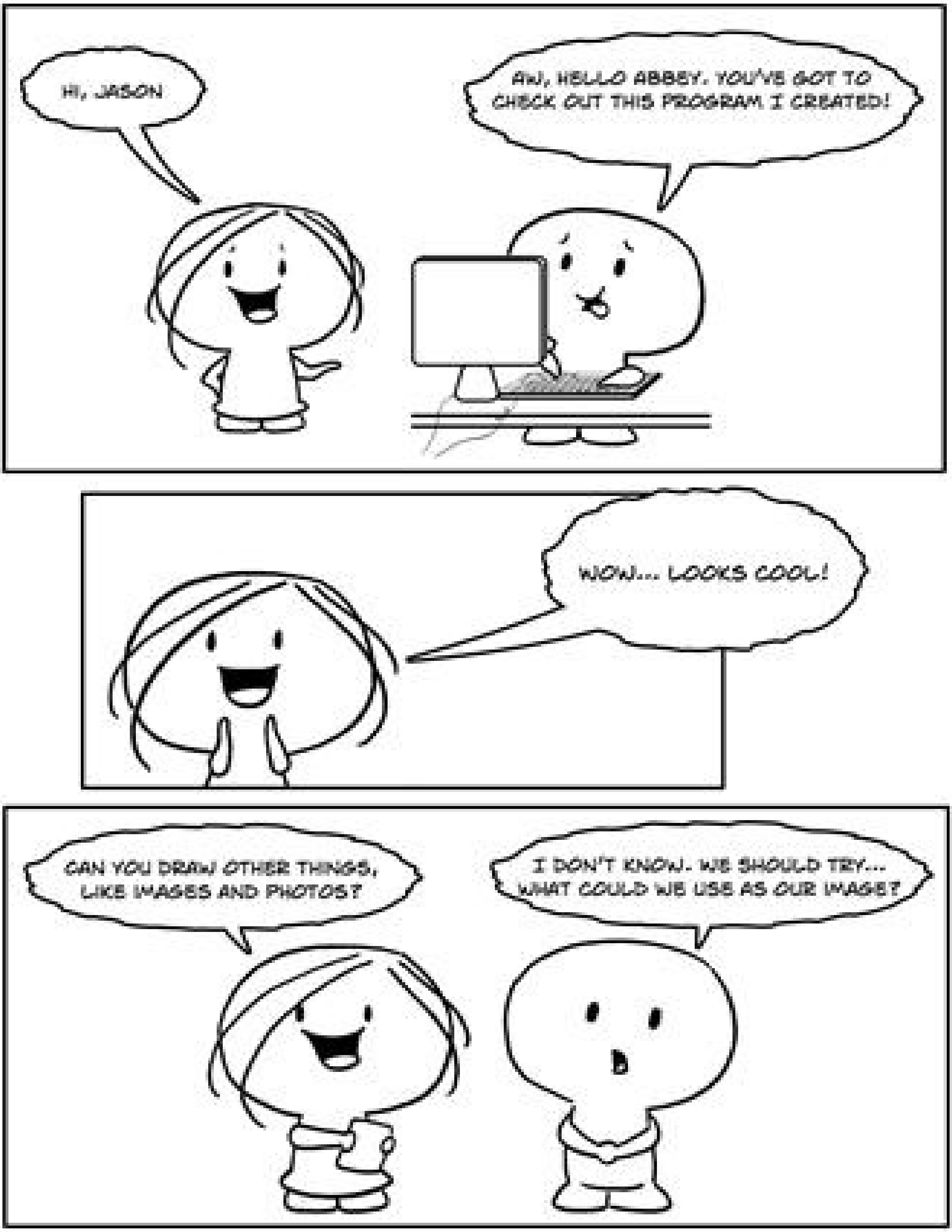
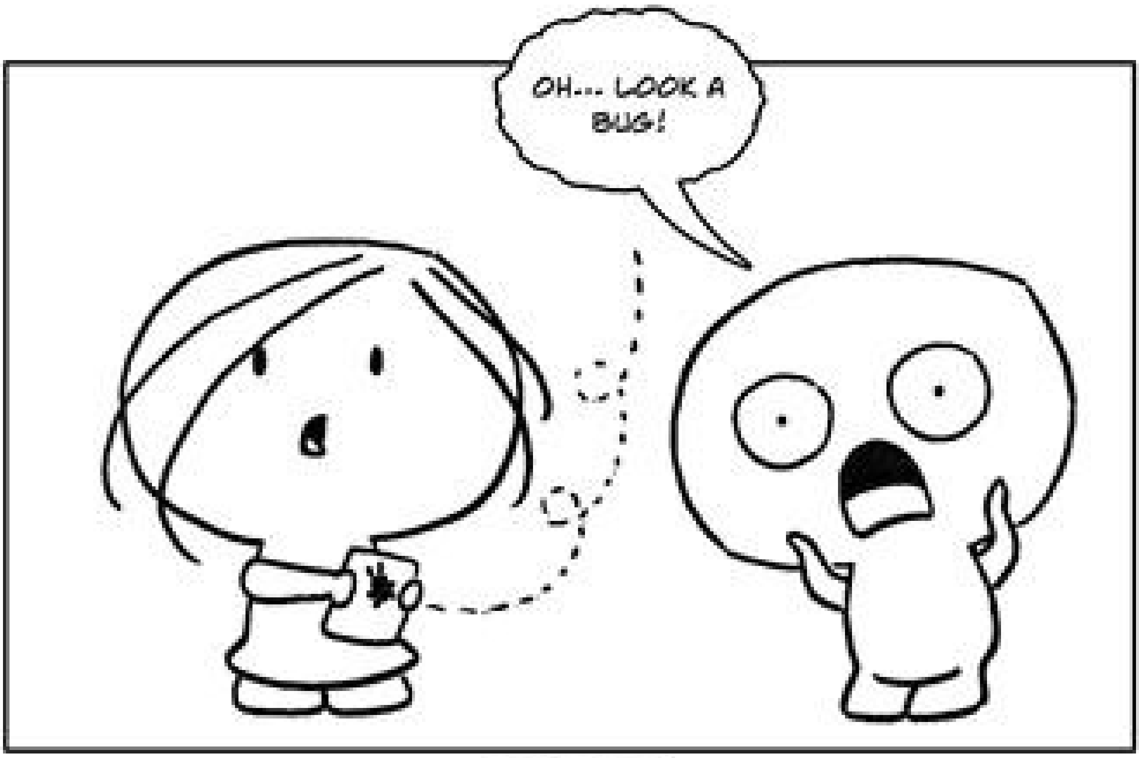
Chapter 2

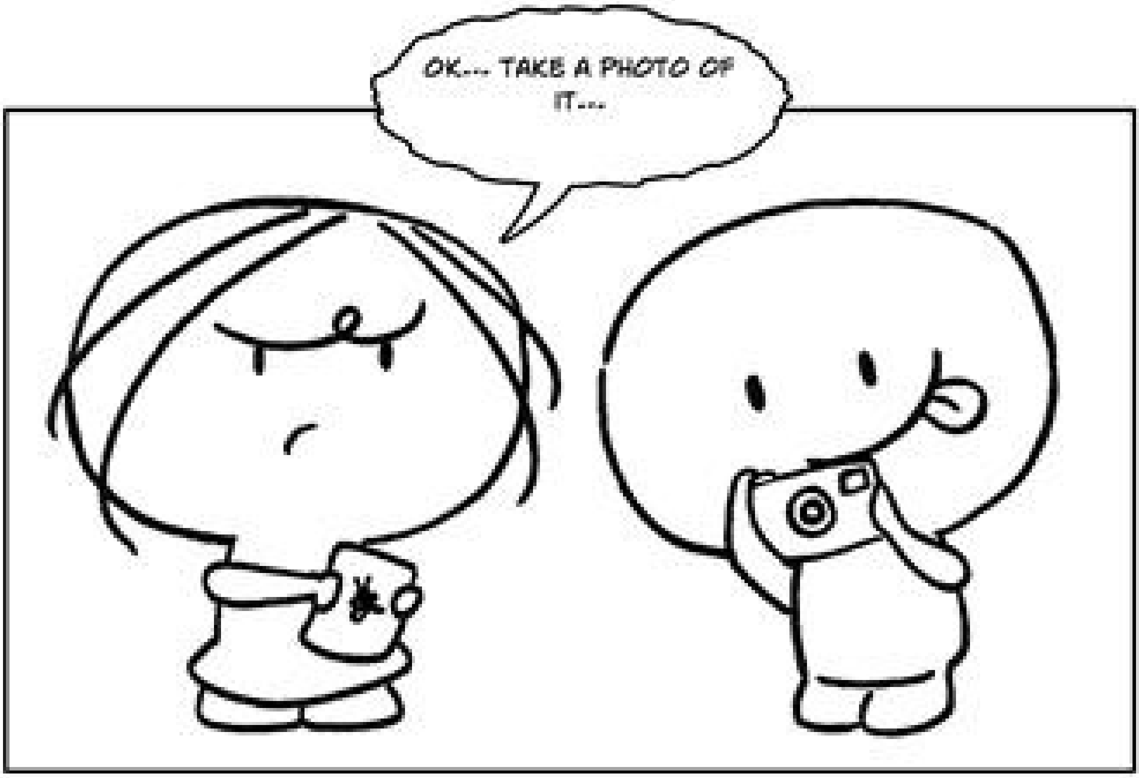
"Images, Fonts and Colors"

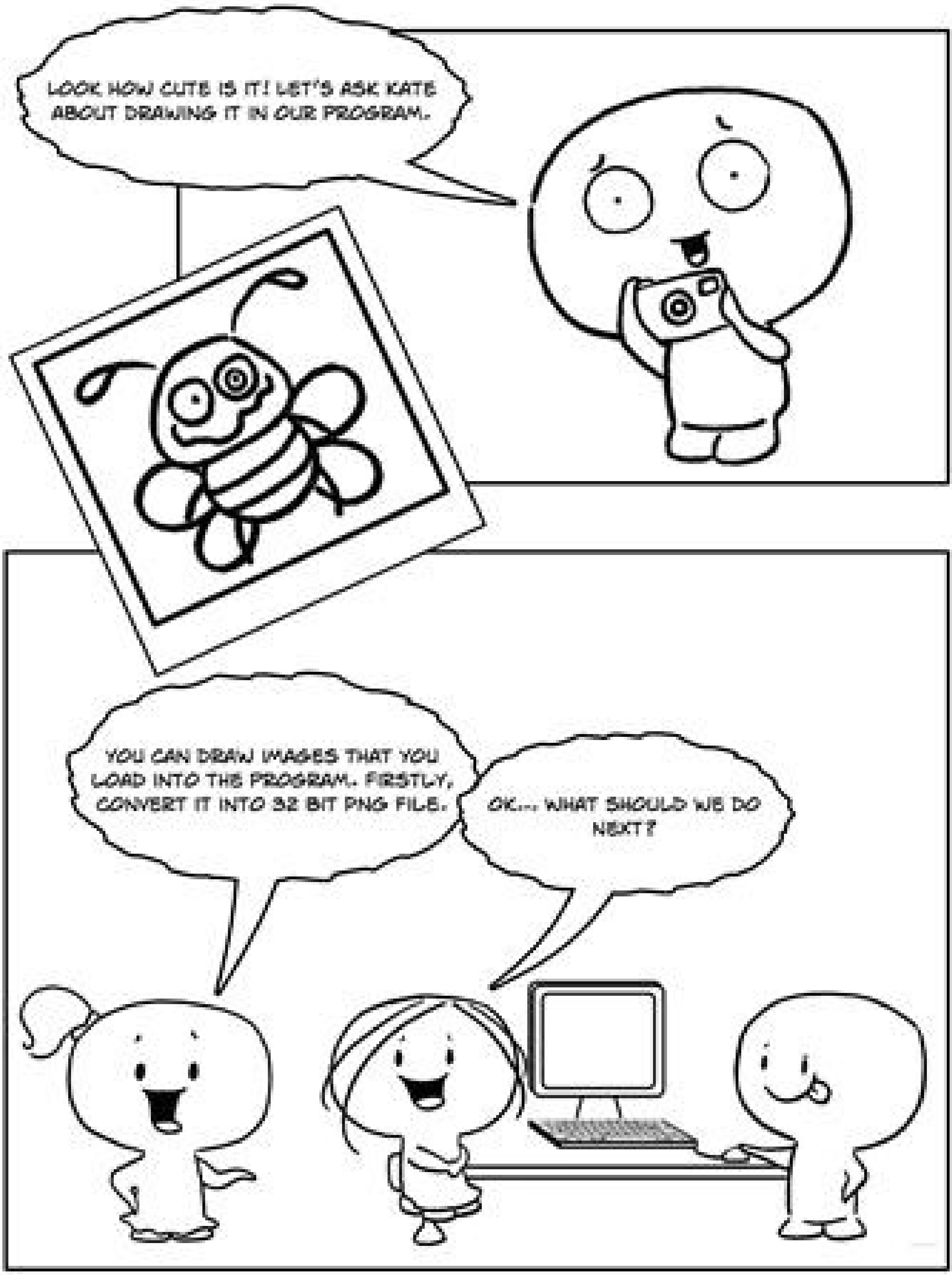
Summary:

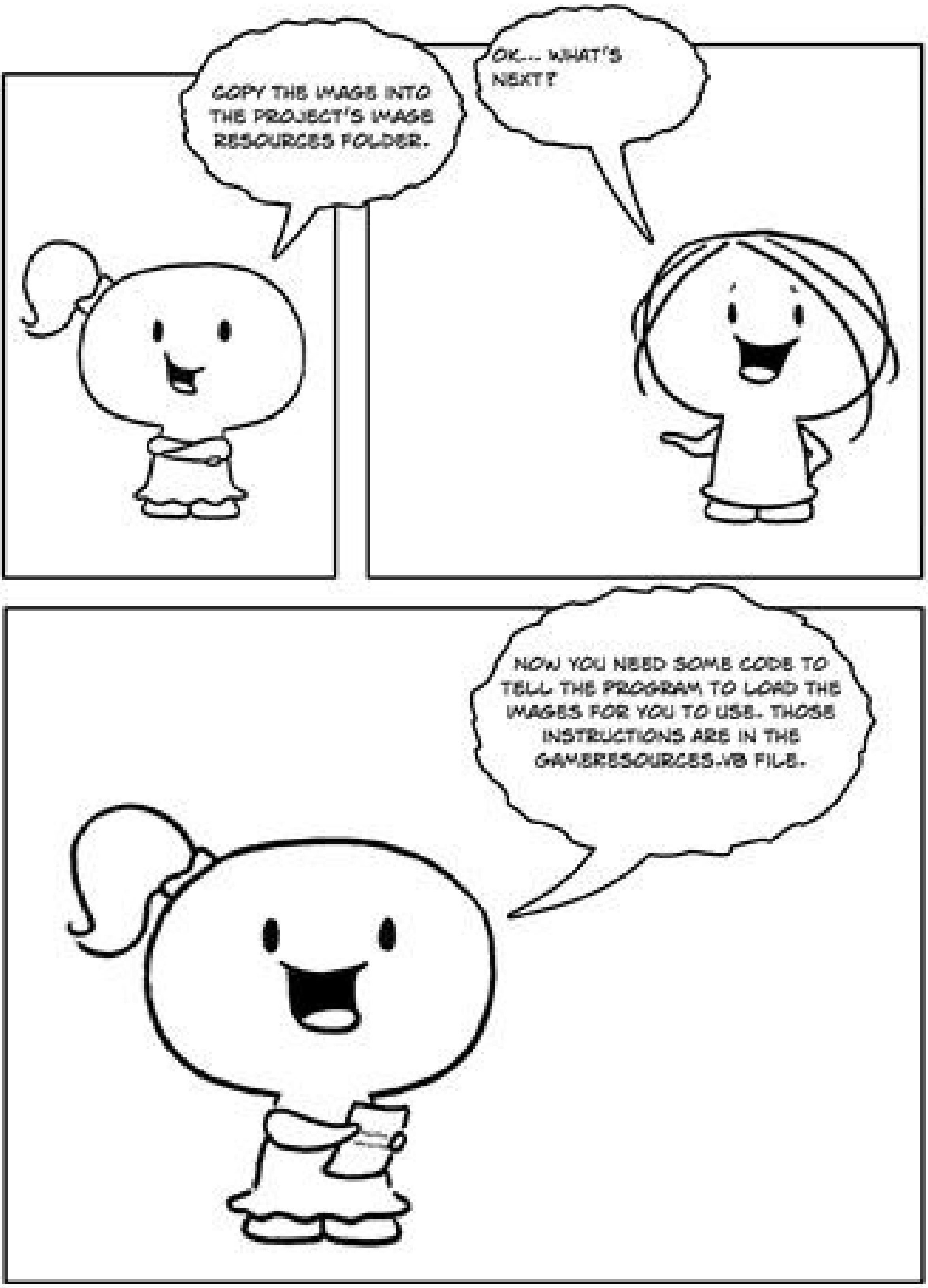
For this chapter you need to create a new project. You will need to have a .png images and true type font. All of necessary resources are provided. You also will be introduced to ARGB color representation.


HI, JASON
AW, HELLO ABBEY. YOU'VE GOT TO CHECK OUT THIS PROGRAM I CREATED!
WOW... LOOKS COOL!
CAN YOU DRAW OTHER THINGS, LIKE IMAGES AND PHOTOS?
I DON'T KNOW. WE SHOULD TRY... WHAT COULD WE USE AS OUR IMAGE?


OH... LOOK A BUG!


OK... TAKE A PHOTO OF IT...


LOOK HOW CUTE IS IT! LET'S ASK KATE ABOUT DRAWING IT IN OUR PROGRAM.
YOU CAN DRAW IMAGES THAT YOU LOAD INTO THE PROGRAM. FIRSTLY, CONVERT IT INTO 32 BIT PNG FILE.
OK... WHAT SHOULD WE DO NEXT?


COPY THE IMAGE INTO THE PROJECT'S IMAGE RESOURCES FOLDER.
OK... WHAT'S NEXT?
NOW YOU NEED SOME CODE TO TELL THE PROGRAM TO LOAD THE IMAGES FOR YOU TO USE. THOSE INSTRUCTIONS ARE IN THE GAMERESOURCES.VB FILE.

Part 1

Open Visual Studio 2008. Create a new SwinGame project and name it “HelloWorld”.

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

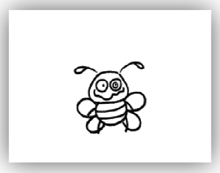
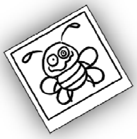


photo.png



2. 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.
3. Open your HelloWorld project in Visual Studio 2008.
   - Open Visual Studio 2008
   - Click File -> Open -> Project/Solution as shown in Figure 1.

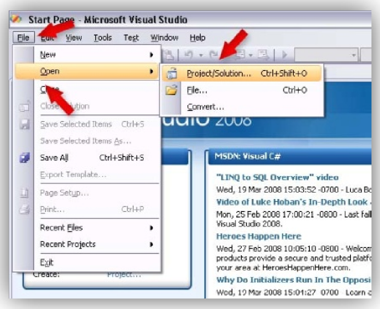


Figure 1

   - Click “HelloWorld” -> “HelloWorld” -> Open Microsoft Visual Studio Solution as shown in Figure 2.

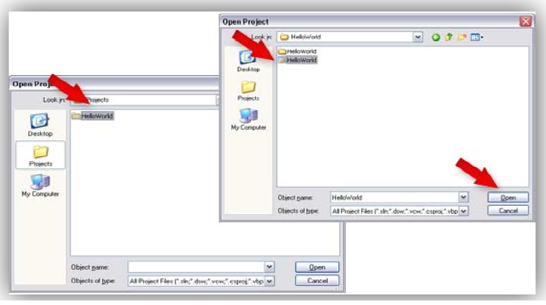


Figure 2

- 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
```

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:

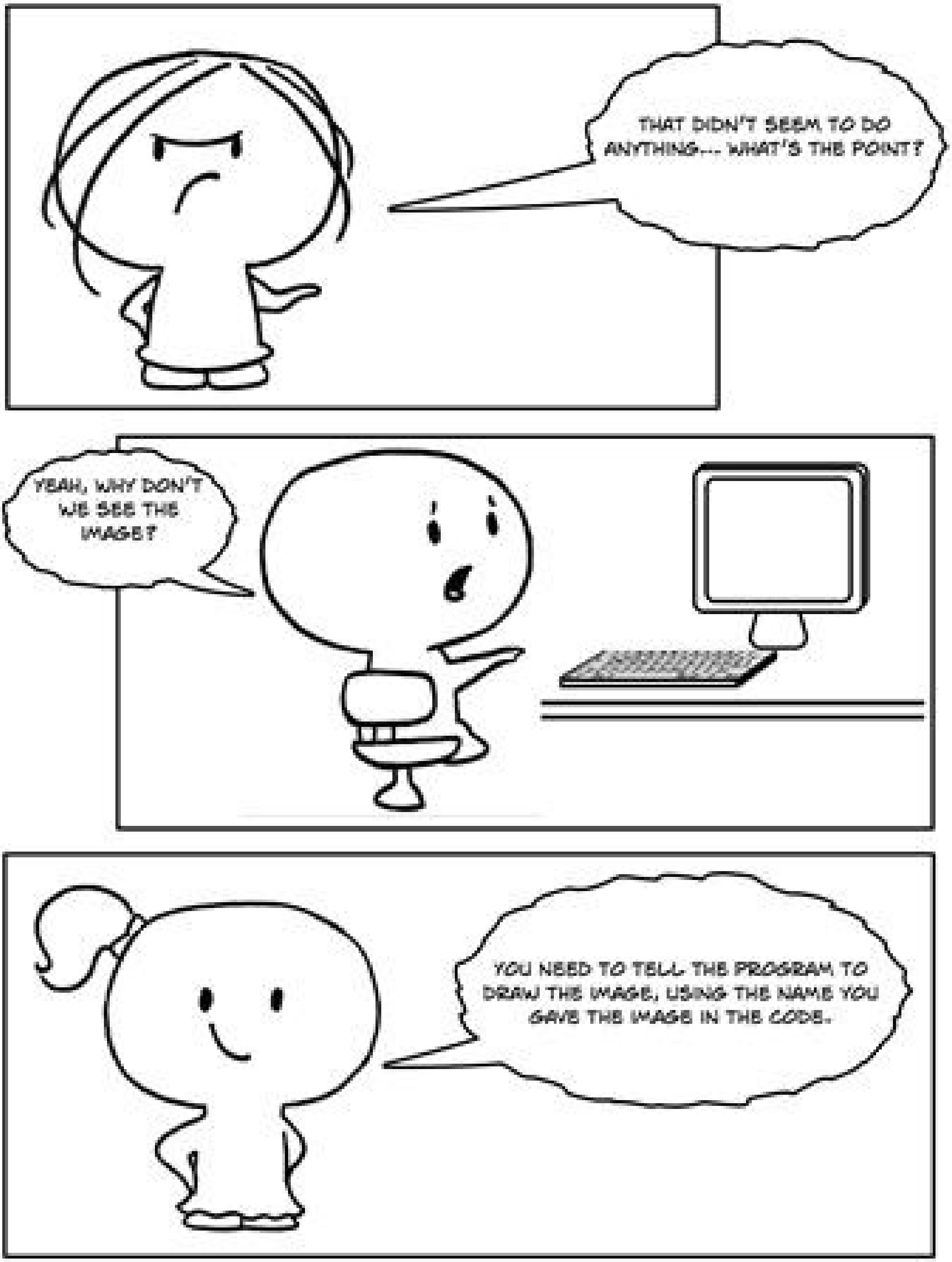
```
…

        NewImage("back", "back.png")
        NewImage("photo", "photo.png")
…
```

Exercise 1: *Loading images into your program*

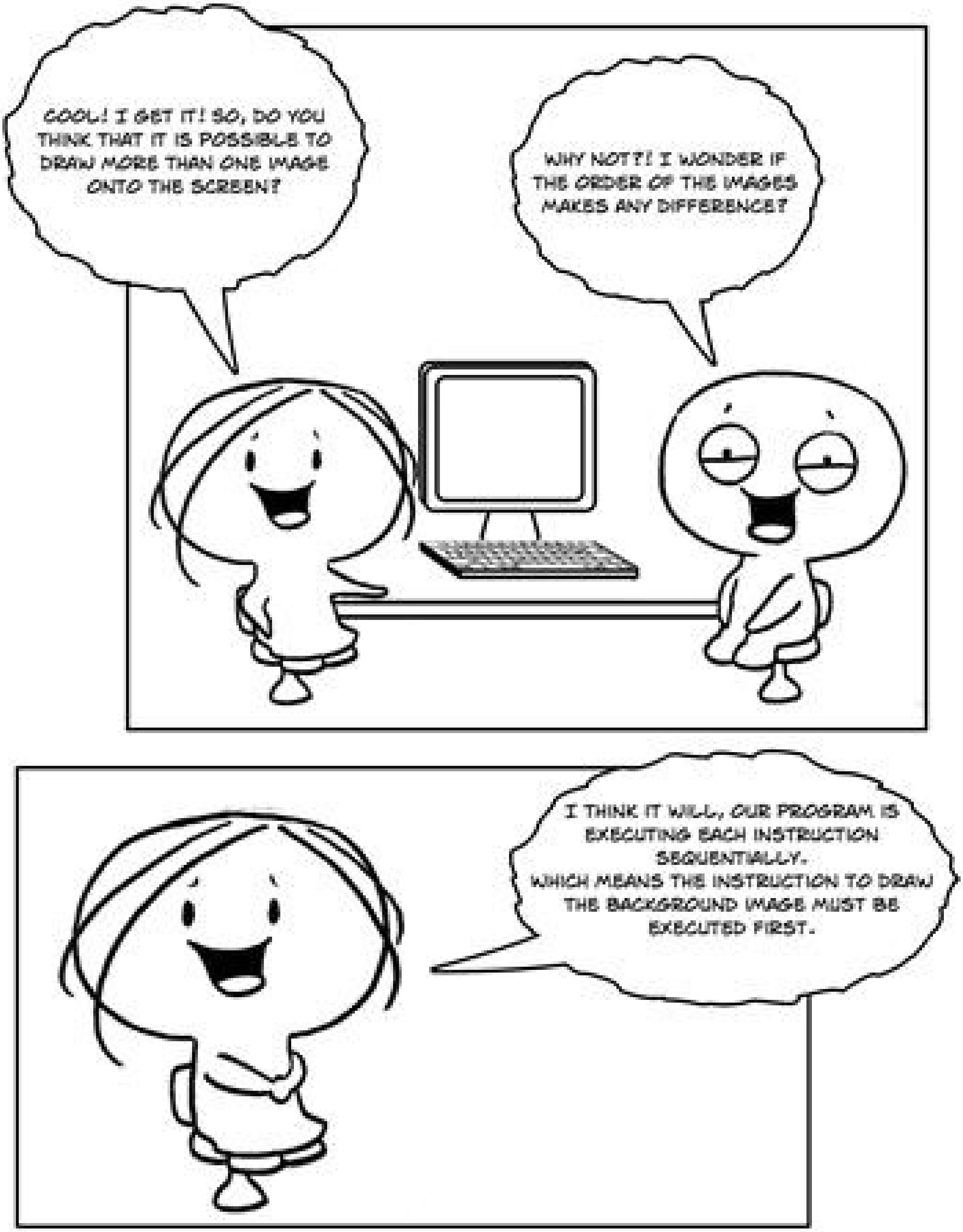
Answer the following questions on your worksheet:

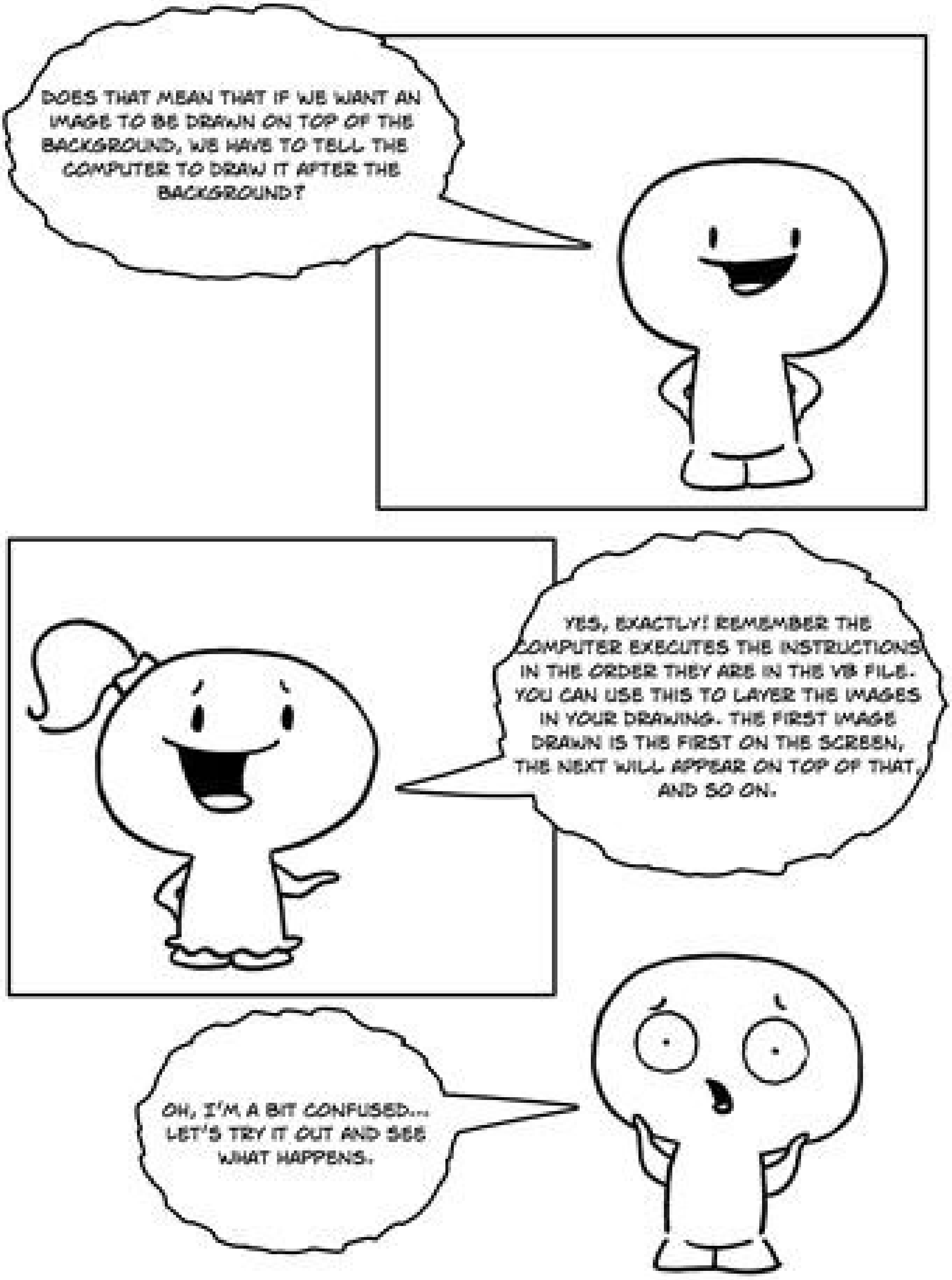
1. Write the complete code for LoadImages() sub into a worksheet.

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



THAT DIDN'T SEEM TO DO ANYTHING... WHAT'S THE POINT?
YEAH, WHY DON'T WE SEE THE IMAGE?
YOU NEED TO TELL THE PROGRAM TO DRAW THE IMAGE, USING THE NAME YOU GAVE THE IMAGE IN THE CODE.


COOL! I GET IT! SO, DO YOU THINK THAT IT IS POSSIBLE TO DRAW MORE THAN ONE IMAGE ONTO THE SCREEN?
WHY NOT?! I WONDER IF THE ORDER OF THE IMAGES MAKES ANY DIFFERENCE?
I THINK IT WILL, OUR PROGRAM IS EXECUTING EACH INSTRUCTION SEQUENTIALLY. WHICH MEANS THE INSTRUCTION TO DRAW THE BACKGROUND IMAGE MUST BE EXECUTED FIRST.


DOES THAT MEAN THAT IF WE WANT AN IMAGE TO BE DRAWN ON TOP OF THE BACKGROUND, WE HAVE TO TELL THE COMPUTER TO DRAW IT AFTER THE BACKGROUND?
YES, EXACTLY! REMEMBER THE COMPUTER EXECUTES THE INSTRUCTIONS IN THE ORDER THEY ARE IN THE VB FILE. YOU CAN USE THIS TO LAYER THE IMAGES IN YOUR DRAWING. THE FIRST IMAGE DRAWN IS THE FIRST ON THE SCREEN, THE NEXT WILL APPEAR ON TOP OF THAT, AND SO ON.
OH, I'M A BIT CONFUSED... LET'S TRY IT OUT AND SEE WHAT HAPPENS.

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)
…
```

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

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.

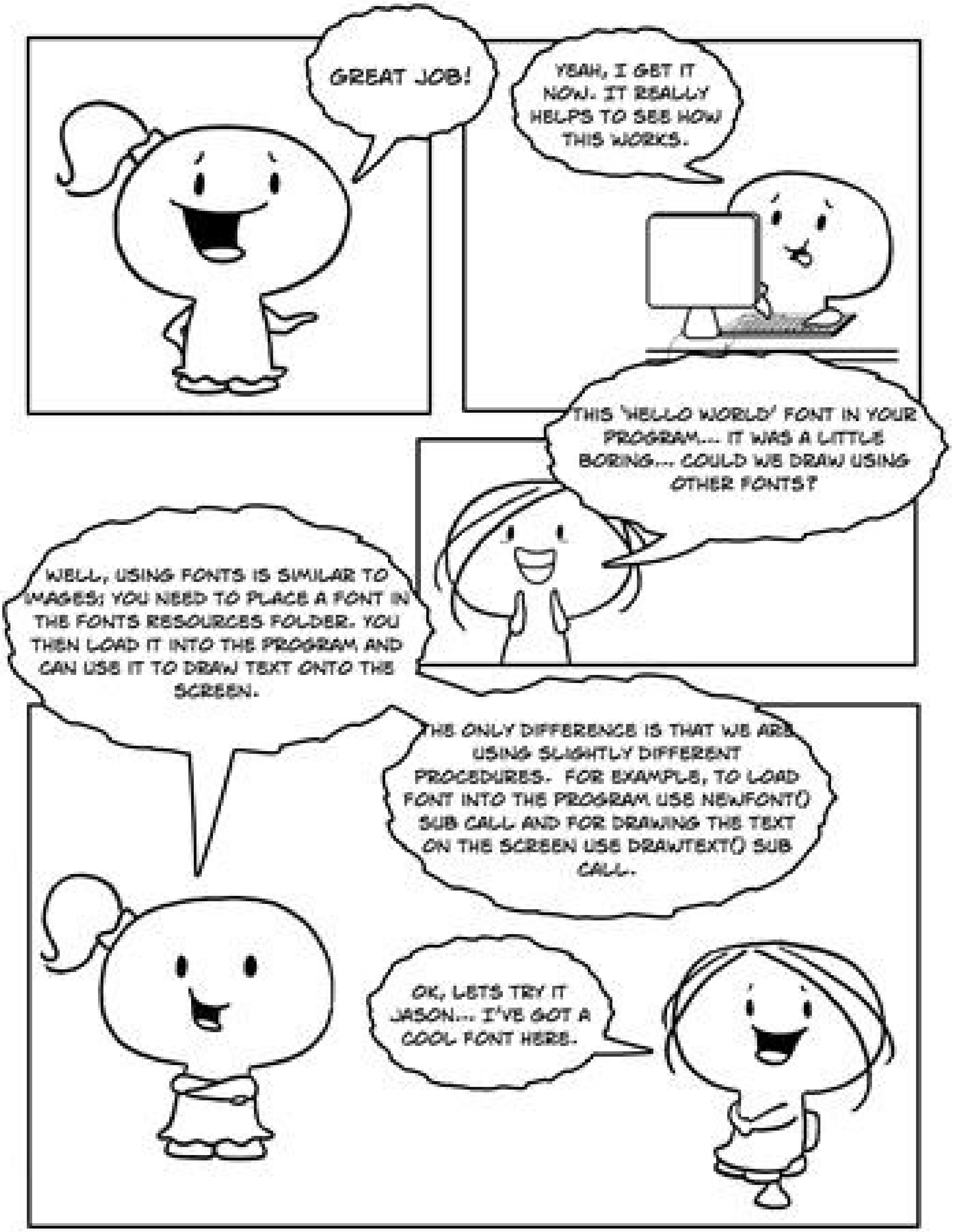
Exercise 1: *Drawing other images on the screen*

Make the following changes to your program and write your solutions onto the worksheet:

1. Tell the computer to draw photo.png on the screen. 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 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.


GREAT JOB!
YEAH, I GET IT NOW. IT REALLY HELPS TO SEE HOW THIS WORKS.
THIS 'HELLO WORLD' FONT IN YOUR PROGRAM... IT WAS A LITTLE BORING... COULD WE DRAW USING OTHER FONTS?
WELL, USING FONTS IS SIMILAR TO IMAGES: YOU NEED TO PLACE A FONT IN THE FONTS RESOURCES FOLDER. YOU THEN LOAD IT INTO THE PROGRAM AND CAN USE IT TO DRAW TEXT ONTO THE SCREEN.
THE ONLY DIFFERENCE IS THAT WE ARE USING SLIGHTLY DIFFERENT PROCEDURES. FOR EXAMPLE, TO LOAD FONT INTO THE PROGRAM USE NEWFONT() SUB CALL AND FOR DRAWING THE TEXT ON THE SCREEN USE DRAWTEXT() SUB CALL.
OK, LETS TRY IT JASON... I'VE GOT A COOL FONT HERE.

Part 3

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)
…
```

Write the complete code for the LoadFonts() sub into the worksheet.

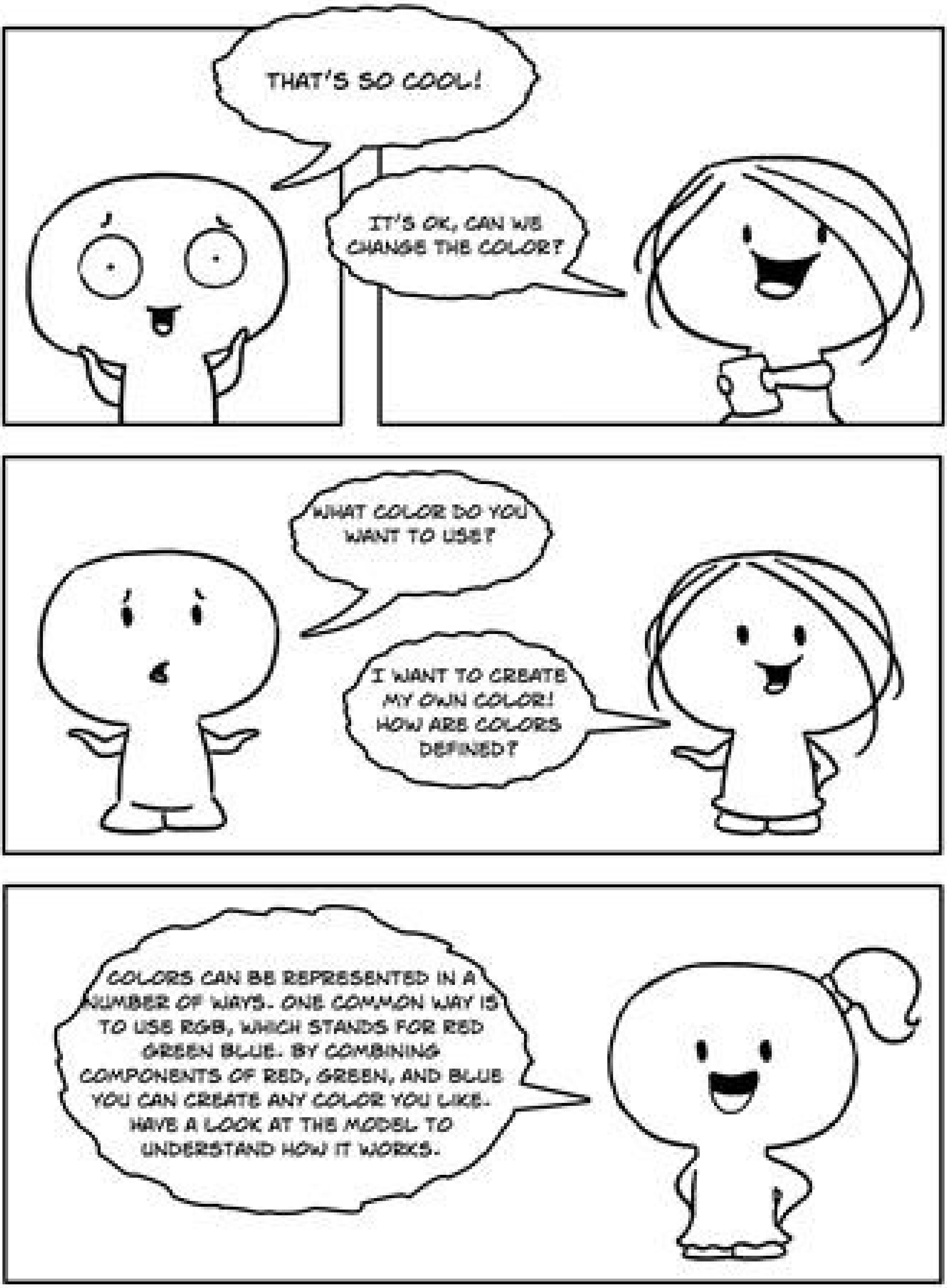
NOTE: We provide only one true type font for this exercise, if you want to use another one, you could download it from http://www.1001freefonts.com/ for free.

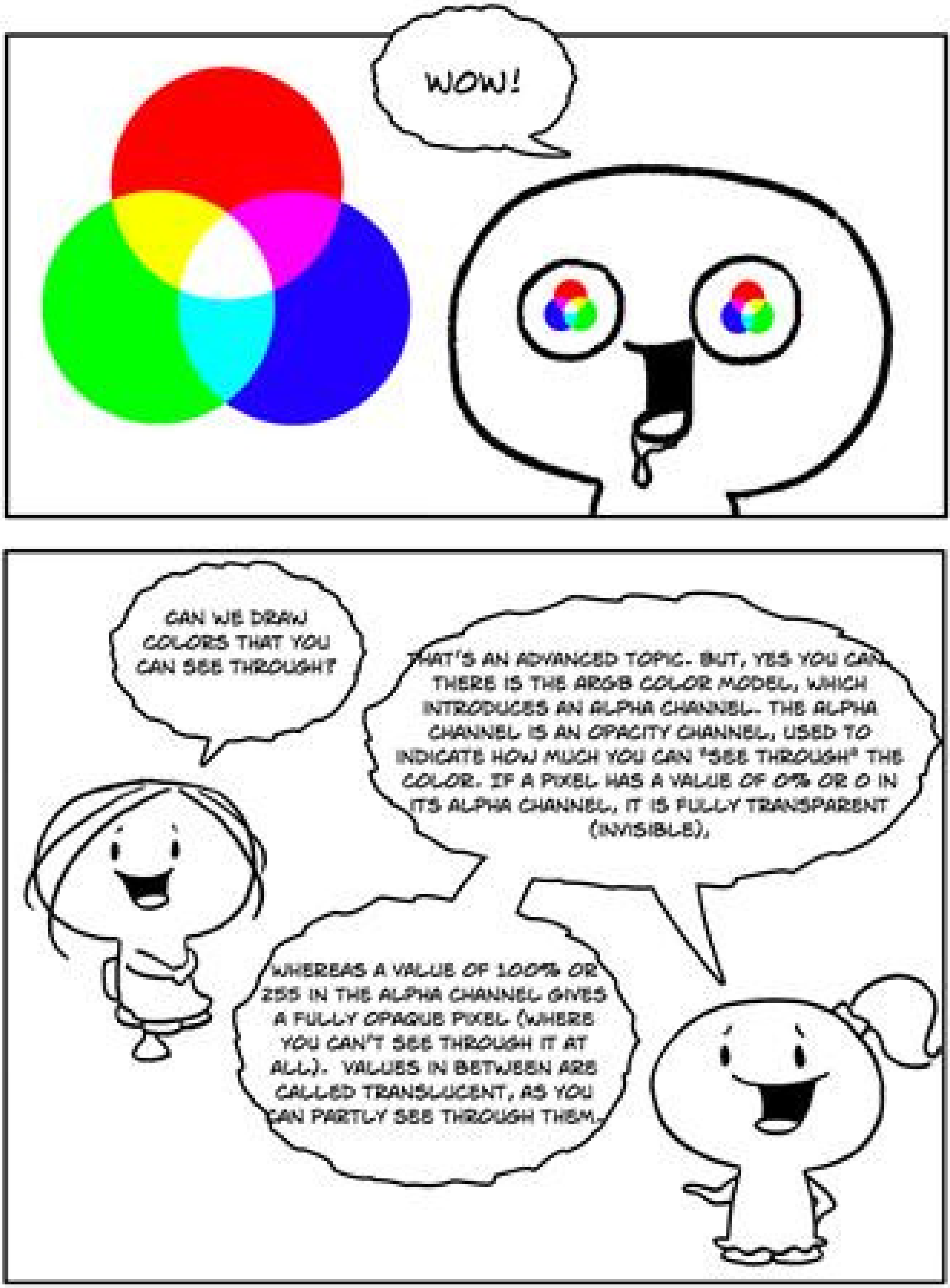
Exercise 1: *Drawing the text by using loaded true type font*

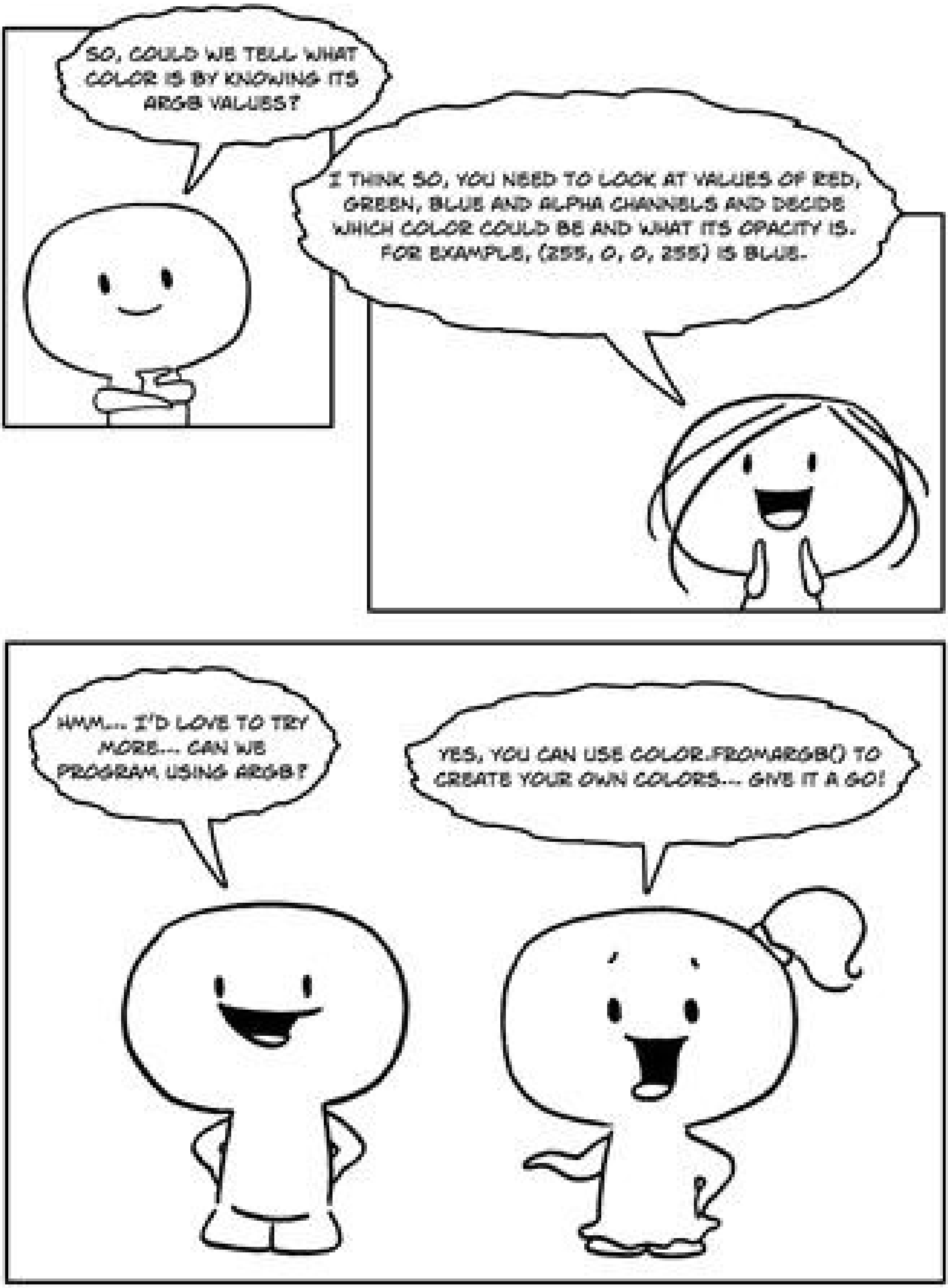
Make the following changes in your "*HelloWorld*" program and write your solution onto the worksheet:

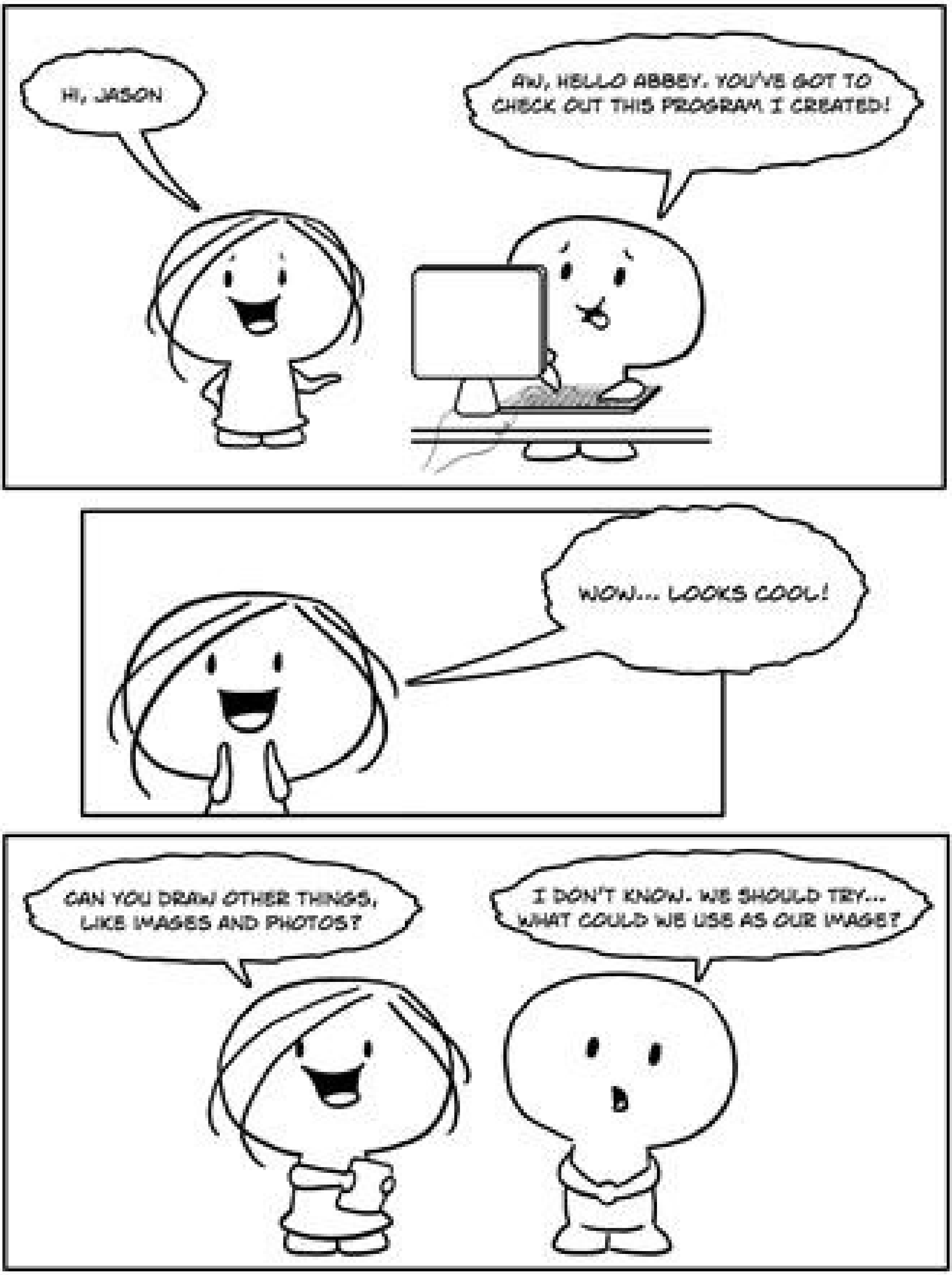
1. Open GameLogic.vb from the Solution Explorer. Use `Text.DrawText("Text to draw", Color.(Black), GameFont("BeanTown"), posX, posY)` to write the code that tells the computer to draw text with a new font to the worksheet.

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


THAT'S SO COOL!
IT'S OK, CAN WE CHANGE THE COLOR?
WHAT COLOR DO YOU WANT TO USE?
I WANT TO CREATE MY OWN COLOR! HOW ARE COLORS DEFINED?
COLORS CAN BE REPRESENTED IN A NUMBER OF WAYS. ONE COMMON WAY IS TO USE RGB, WHICH STANDS FOR RED GREEN BLUE. BY COMBINING COMPONENTS OF RED, GREEN, AND BLUE YOU CAN CREATE ANY COLOR YOU LIKE. HAVE A LOOK AT THE MODEL TO UNDERSTAND HOW IT WORKS.


WOW!
CAN WE DRAW COLORS THAT YOU CAN SEE THROUGH?
THAT'S AN ADVANCED TOPIC. BUT, YES YOU CAN. THERE IS THE ARGB COLOR MODEL, WHICH INTRODUCES AN ALPHA CHANNEL. THE ALPHA CHANNEL IS AN OPACITY CHANNEL, USED TO INDICATE HOW MUCH YOU CAN "SEE THROUGH" THE COLOR. IF A PIXEL HAS A VALUE OF 0% OR 0 IN ITS ALPHA CHANNEL, IT IS FULLY TRANSPARENT (INVISIBLE),
WHEREAS A VALUE OF 100% OR 255 IN THE ALPHA CHANNEL GIVES A FULLY OPAQUE PIXEL (WHERE YOU CAN'T SEE THROUGH IT AT ALL). VALUES IN BETWEEN ARE CALLED TRANSLUCENT, AS YOU CAN PARTLY SEE THROUGH THEM.


SO, COULD WE TELL WHAT COLOR IS BY KNOWING ITS ARGB VALUES?
I THINK SO, YOU NEED TO LOOK AT VALUES OF RED, GREEN, BLUE AND ALPHA CHANNELS AND DECIDE WHICH COLOR COULD BE AND WHAT ITS OPACITY IS. FOR EXAMPLE, (255, 0, 0, 255) IS BLUE.
HMM... I'D LOVE TO TRY MORE... CAN WE PROGRAM USING ARGB?
YES, YOU CAN USE COLOR.FROMARGB() TO CREATE YOUR OWN COLORS... GIVE IT A GO!


HI, JASON
AW, HELLO ABBEY. YOU'VE GOT TO CHECK OUT THIS PROGRAM I CREATED!
WOW... LOOKS COOL!
CAN YOU DRAW OTHER THINGS, LIKE IMAGES AND PHOTOS?
I DON'T KNOW. WE SHOULD TRY... WHAT COULD WE USE AS OUR IMAGE?

Part 4

Exercise 1: *Colors with RGB and RGBA*

Answer 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).
2. What is the color and how much could you see with the following ARGB values: (30, 0, 255, 0)
3. 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*

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. Use `Color.FromArgb(redVal, greenVal, blueVal, alphaVal)` as the color of rectangle.