Abbey: Hi, Jason

Jason: Aww, hello Abbey. You’ve got to check out this program I created!

Abbey: Wow… looks cool! Can you draw other things, like images and photos?

Jason: I don’t know. We should try… what could we use as our image? Oh… look a bug!

Abbey: Ok… take a photo of it…

Jason: Look how cute is it! Let’s ask Kate about drawing it in our program.

Time passes…

Kate: Yes, you can draw images that you load into the program. Firstly, convert it into 32 bit PNG file.

Abbey: Ok… What should we do next?

Kate: Copy the image into the Project’s image resources folder.

Abbey: Ok… What’s next?

Kate: 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.

*Tutorial….part 1*

Abbey: That didn’t seem to do anything… what’s the point?

Jason: Yeah, why don’t we see the image?

Kate: You need to tell the program to draw the image, using the name you gave the image in the code.

Abbey: Cool! I get it! So, do you think that it is possible to draw more than one image onto the screen?

Jason: Why not?! I wonder if the order of the images makes any difference?

Abbey: I think it will, our program is executing each instruction sequentially. Which means the instruction to draw the background image must be executed first.

Jason: 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?

Kate: 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.

Jason: Oh, I’m a bit confused… Let’s try it out and see what happens.

*Tutorial… Part 2*

Kate: Great job!

Jason: Yeah, I get it now. It really helps to see how this works.

Abbey: This ‘Hello World’ font in your program was is a little boring… could we draw using other fonts?

Kate: 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.

Abbey: Ok, lets try it Jason… I’ve got a cool font here.

*Tutorial… Part 3*

Jason: That’s so cool!

Abbey: It’s ok, can we change the color?

Jason: What color do you want to use?

Abbey: I want to create my own color! How are colors defined?

Kate: 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.

Jason: WOW!

Abbey: Can we draw colors that you can see through?

Kate: 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.

Jason: So, could we tell what color is by knowing its ARGB values?

Abbey: 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.

Jason: Hmm… I’d love to try more… Can we program using ARGB?

Kate: Yes, you can use Color.FromARGB() to create your own colors… give it a go!

*Tutorial ….Part 4*