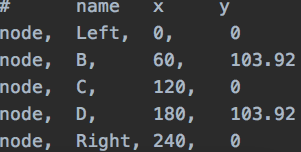
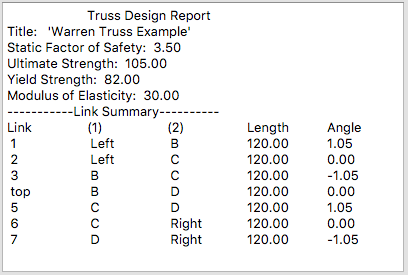
Last week we discussed remedial English in a way. This week, we are discussing remedial art class. We want to draw a figure on our GUI using lines and shapes. We imagine having a set of **(colored) pencils or pens (for drawing lines)**, **paint brushes (for filling in shapes)**, **paint (i.e., colors, polka dots, etc.)**, and a **piece of paper or canvas** on which to create our art. Some other tools in our engineering art kit might include a straight edge/ruler for drawing lines of particular length, graph paper with a grid of fixed size for easily locating points on the 2-D paper, a protractor for measuring angles, and a compass for drawing arcs. Animation on the computer is essentially like a flip cartoon book you might have made as a child, where each frame can be drawn with a slight difference from the previous frame. When you flip the pages, you see the animation happen. This is effectively what a computer does, but with much more intricate images and very fast flipping of the pages.

Let’s do a quick example of sketching a static truss:

***Example***: A Truss (from HW8 SP20)



From out previous homework, I might want to sketch this truss so I can see what it looks like. On engineering graph paper, I would sketch it as follows:

Left (0,0)

B (60,103.92)

Right (240,0)

C (120,0)

D (180,103.92)

My steps to create this graphic in Microsoft Word were to:

1. From ribbon, Select - *View→Gridlines* (checked on Show panel) This helped me draw things to scale. I just decided that each grid line represented 20 units.
2. From ribbon, Select – *Insert→Shapes→Drawing Canvas* to create a rectangular region on my document where I can draw the truss. I set the outline to a blue, thin line.
3. From ribbon, Select – *Insert→Shapes→Oval* (draw circles at node points assuming an origin at node ‘Left’ with *y*-axis pointed up, *x*-axis pointed right.) Note: the circles are drawn with a darkish blue outline with a lighter blue fill. The width of the outline can be specified as can the colors of the line and the fill.
4. From the ribbon, *Select – Insert→Shapes→Line* (draw the line segments connecting the nodes according to the truss report). The line segments are blue in color and I can select the width.
5. Finally, I labeled my nodes by *Insert→Shapes→Text Box*.

No doubt, if you were to draw the same truss by hand using a box of colored pencils, you could reproduce a reasonable sketch with the nodes represented by shaded blue circles and blue lines and black text. If your colored pencils were mechanical, you might want a range of lead sizes to draw lines of different width to make your art look better.

**LED displays**:

*pixels*

When a computer draws graphics (we will restrict our attention to 2-D drawings on your screen), it is really illuminating (or dimming) a pattern of small light patches (pixels) arranged in a 2-D grid. Granted, the pixels are very small and our human eyes often can’t see this underlying structure of a LED (light emitting diode) display. It turns out that there are actually 3, side-by-side grids of LED’s, one is Red, one is Green, and one is Blue. By controlling brightness of three adjacent Red, Green and Blue LED’s, one may obtain any color for the pixel (remember the color wheel from art class). Our eyes and brain actually do the color blending and we see a continuous spectrum of colors. So, to draw a line on a computer screen, we need to tell the graphics card of the computer exactly which pixels to illuminate and how much.

*coordinates*

Since the array of pixels is inherently 2-dimensional and the physical monitor has a certain number of pixels wide × pixels tall, we can specify a pixel location with two coordinates (*nx*, *ny*). Our intuition (and previous experience) as engineers and students of mathematics is that a logical reference pixel (*origin*) would be the bottom left of the display screen and that we should use a right-handed coordinate system. Of course, we could have a global origin/coordinate system (associated with the physical computer screen) and a floating/local coordinate system (associated with a window, button, group box, etc.) to make it easier to specify points.

**A HUGE POINT OF CONFUSION** for most of us is that the origin for drawing with computers is actually at the top-left of the screen/canvas and the orientation of the *y*-axis is pointed downward while the *x*-axis still points to the right. The reason for this is historical and related to the way original displays (called cathode ray tubes, or CRTs) scanned an electron beam from top to bottom of a screen to create an image. Like much in computer science, this original coordinate system has persisted and can be a source of frustration for beginners.

The FACT that the display is inherently digital (coordinates are integers) with an upside-down *y*-axis is not insurmountable, because we can estimate floating point coordinates with integers and get good results if the pixels are small. Likewise, a simple transformation of coordinates can be done to ‘flip’ the *y*-axis to specify points on the screen after you have done your calculations.