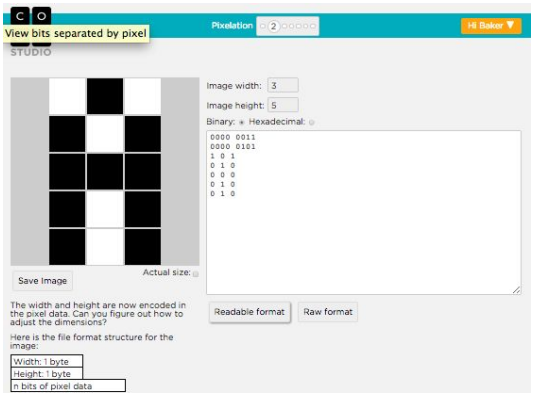
Name(s)_____ Period _____ Date _____

Activity Guide - B&W Pixelation Widget

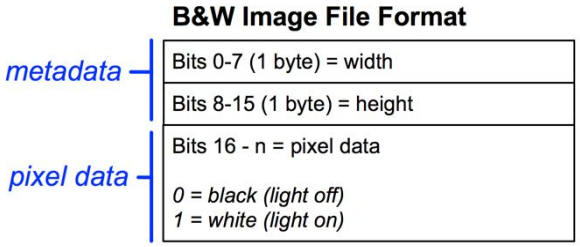
The B&W Pixelation Widget

The pixelation widget uses a file format as depicted below. For example, the 3x5 image of the letter “A,” shown at right within the Pixelation tool, would be encoded as a simple stream of these bits, organized like this (color added for emphasis):



We can break it up into pieces like so:

purpose	size	example
width	1 byte	0000 0011
height	1 byte	0000 0101
pixel data	varies	0000000110101000001 0010



Student Tasks

Log into Code Studio to get started with the pixelation widget. In code studio there are a few short tasks to help you get acquainted with the tool.

- **Create a small image:** Start by trying to recreate the 3x5 letter “A” depicted (shown above) using the pixelation widget.
- **Correct the error:** Oh no! An extra bit was inserted into an image during transmission! Track it down.
- **Make your own image of any size.**
  - Encode an image of anything you like.
  - Do not simply make an abstract pattern, like a checkerboard. It should depict something, perhaps your name written out, an icon or logo of some sort.
  - Optional: For fun, send your image bits to a friend using the Internet Simulator

When you are finished, complete the table below and answer the following questions:

Your image: Save your image and drag-and-drop or copy/paste it below.	**Your bits:** Copy/paste the bits that create your image below.

Questions

1. Using the B&W file format from the pixelation widget
    - What are the largest dimensions (width and height) of an image we can make with the pixelation widget?
    - How many total bits would there be in the largest possible image we could make with the pixelation widget?
    - How many bits would it take to represent the smallest possible image (i.e. an image with one pixel)?
2. What would happen if we didn't include width and height bits in our protocol? Assume your friend just sent you 32 bits of pixel data (just the 0s and 1s for black and white pixels). Could you recover the original image? If so, how?