CS 2110 Lab 18
Pixels

The TAs :)

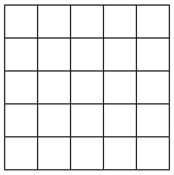
10/28/2018

1 Recitation Outline

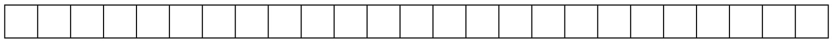
1. Pixels

2 Pixels

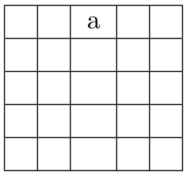
Our screen is divided up into small segments called pixels. When we think of a screen, we tend to think of the following image (suppose it's 5 pixels by 5 pixels):



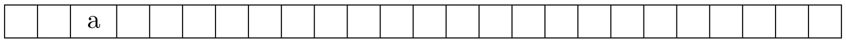
But on the Gameboy Advance, our pixels are actually stored in a 1-D array of length 25 (= 5 x 5) in memory like so:



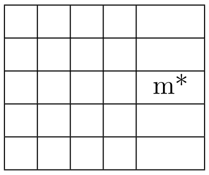
If we wanted to set the element at (0, 2) (row = 0, col = 2) to be 'a', our screen would end up looking like:



While our array in memory would reflect:



This begs the question, how do we set the 1D array in memory while being given the row and col of the specific position? From the last example, you may think that the answer is simply array[row + col]. But what if we wanted to set the element at (2, 4) to 'm*'.



Using our previous logic, we would think that this means we would set the 2 + 4 = 6th element in the array to 'm*':

					m									m*										

But this element in the array does not correspond exactly! 'm' is where we think the element should be placed, while 'm*' is the exact position it should be.

Notice that 'm*' is located at element 14, or at the element calculated by 2*5 + 4. In general, the correct position of the element in memory is:
`(row * (width of screen)) + col`