Reflection:

@author: Christina He (worked with Audrey Dawson)

@version: 19 September, 2017

It’s a really difficult task for me. And I think the hardest part in it is to find the relationship between the MapGrid and MapViewer. At first we think that the array in the MapGrid should be extended to fill the whole window with magnification 1. So we tried to calculate the pixels in every array and divided x coordinate of offset point by the pixels in one array to calculate which array the offset point was in. But then it turned out to be there is only one pixel in every array. At this way we can calculate the offset point directly.

Though we reversed the process for drawing the map in a more confusing way, this provides me a different perspective to see the question, and I think having many pixels in one array can be related to the rectangle. By thinking this way, maybe we can come up with the idea of how to manage the color square by square, instead of pixel by pixel.

And when we wrote our code to determine whether the offset is inside or outside of the map at first, there was always an OutOfBound error. We used about one hour to debug it. And finally we found that we only need one “=” in our “if” statement. So we should really pay attention to those kind of subtle logic, and when you can’t find your bug, use “System.out.println” to print the way the code runs. That’s really helpful.

Also, when I review our code after today’s class, I have a better understanding about the frame, JComponent, and pane. We first create a frame (window), and then we have a pane. The JComponent can be applied to the pane to draw pictures. What’s more, I’m more familiar with the Graphics class, and have a basic idea of its method.