**Final Exam Write-Up**

1. In my mind, the first step is removing the noise present in the image. Given that the diameter of the largest noise is 3 pixels, this can be accomplished by eroding the image and then dilating it, two morphology operations that equate to the close operation. This will not change the center of any discs present in the image, but it will remove the noise.
2. My next step would be to isolate the lines. To do this, I would clamp the image in a very specific way. The green used for the lines is different than the green used for the circles. By clamping the image with upper and lower bounds of the green that is used for the lines, we can remove the circles from a copy of the image.
3. I would take the image with isolated lines and then run the Hough Transform for lines on it. I cannot remember if Hough Transform for lines will return the line segments as well as lines. If it does return a lot more results than lines I am looking for, I would use K-Means Clustering on the results. I know I need lines that are traveling at a 90-degree angle and a 0-degree angle. If I alter the parameters of the clustering to retrieve groups of those lines, I can reliably derive information about the grid I’m trying to fill and store the bounds (ie the upper left corner and the bottom right corner). I will create a data structure that stores each of the grid spaces.
4. Returning to my image after the close operation, I now want to start the process of isolating the circles. This entails a set of routines that I need to execute 5 times, once for each type of disk starting with the largest.
   1. Create a binarized copy of the current image (the first time through this is the image with noise removed). This will remove all color and leave me with just circular shapes. This binarization does not require otsu but it will be customized. Black or the green of the lines is black, anything else is white.
   2. Next, I will use canny edge detection for this image. I need to remove the filling of the circles for the next algorithm.
   3. Now I can run Hough Transform for circles on this binarized edge-only image. I believe Hough Transform entails parameters for radius, in which case I will set it to the radius of the current disk for this iteration of the loop. This should return a giant list of circles that I can filter through.
   4. Regardless of Hough Transform for circle capabilities, I will perform my own filtering. Looping through I will discard any circles that have a radius/diameter less than that of the current disk for this iteration of the loop.
   5. I will then perform K-Means clustering with groups equal to the number of grids that my line detection sub-routine detected and initially populated with circles in the center of those grids. This should reduce the detected circles to just the disks matching the current disk for this iteration of the loop.
   6. After the K-Means clustering, I will iterate over each of the resulting circles and increase the point value of the grid space which the center of the circle falls into by the point amount corresponding to the disk for this iteration of the loop.
   7. Finally, to prevent the detection of greater-sized disks, I will construct an entirely black image, populated only by white circles in the exact same locations as the detected circles. I will combine this image with the image from 4a (before binarization) via a customized exclusive or. Black and Black will leave the background untouched, any color and black will be maintained, but any color (not black) and white) will result in black. This will erase all the circles matching the disk for this iteration from the image. This resultant image will become the current image for the next iteration of the loop.
5. After the loop has run through each of the disk sizes, each of the grid’s squares will have updated based on the disks centered in them.