# Introduction

In a previous assignment, we were tasked with creating a 2D array and dynamically filling it with a triangle and a glass shape. In our current assignment, we are to take that array and rotate it 90 degrees 3 times, printing it to the console at 90, 180, and 270 degrees, respectively. This program will accomplish that task by fulfilling the following functional requirements:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** |  | **Functional Requirement** |  | **Value** | **Stakeholder** |
| FID001 | **I want to** | create a 2D array | **so that** | I can use the 2D array to store data | student |
| FID002 | **I want to** | dynamically fill the array | **so that** | the program fulfills the requirements of the assignment | student |
| FID003 | **I want to** | fill the array with asterisks and spaces | **so that** | A 2D image will be created | student |
| FID004 | **I want to** | print the 2D array to the console | **so that** | the user can see that the array has been successfully created | instructor |
| FID005 | **I want to** | rotate the array 90 degrees clockwise | **so that** | the array may be printed at a 90 degree rotation | student |
| FID006 | **I want to** | rotate the array 180 degrees | **so that** | the array may be printed at a 180 degree rotation | student |
| FID007 | **I want to** | rotate the array 270 degrees | **so that** | the array may be printed at a 270 degree rotation | student |
| FID008 | **I want to** | print the rotated array at 90, 180, and 270 degrees | **so that** | the user can see that the array has been successfully rotated | instructor |

By fulfilling all of the above requirements, the problem presented will be successfully solved.

# Analysis

We had previously written a program that satisfied the first 4 functional requirements in the following ways:

“Since we knew that only single characters would make up each element of the array, the most efficient way to fulfill FID001 was to use *char[][]*to create the array, and set the size to 8 by 16. Since FID002 required that the array be filled dynamically, conditionals were used, specifically *for* and *if-else* loops. The rows of the array were iterated through, and logic was used to fill each column with asterisks at the appropriate indexes, as per FID003. Finally, *System.out.print()* was used in tandem with *System.out.println()* to print the image to the console, thus satisfying FID004.”

To fulfill FID005, a *transpose()* method was written to create a temporary array with 16 rows and 8 columns. This array was then iterated through, and each column was filled with the contents of the original array’s rows, effectively rotating the array 90 degrees clockwise. FID007 was accomplished by using the *transpose()* method and then iterating backward through the rows, creating a 270 degree rotation. The 180 degree rotation was achieved by iterating backward through the rows and columns of the original array, creating a flipped and mirrored image, thus satisfying FID006. Each of the methods created incorporated *System.out.print()* and *System.out.println()*, in order to print each version of the image to the screen and accomplish FID008.

# Conclusion

I’m sure that you will find that the program sufficiently and effectively solves the problem set forth. Upon running MainEntry.java, a crude image of a triangle and a wine glass made up of asterisks will be printed to the console, followed by that same image rotated by 90, 180, and 270 degrees, respectively.

I think I was about halfway to solving this problem before Monday’s class. I had incorporated the temporary array for transposing the original image, and my method looked almost identical to yours, but I was having trouble with some of the syntax. The idea of iterating backwards through the array had not crossed my mind, but in hindsight, it seems so obvious. Sorry to go off-topic, but I’d like to talk about the quizzes for a second. The last two I haven’t done well on (I haven’t received the most recent one back, of course, but I’m pretty sure I biffed it). Frustratingly, I end up figuring out the correct answers at some point during the class session. Despite what the quiz results may show, I do feel like I am starting to get it, and I’m becoming more comfortable with Java in general.

The repository for this assignment can be found at:

<https://github.com/cellson7170/ce_cs260_wk3_DataTwirl>