Journal Report 7
10/14/19-10/25/19
Oliver Hayman
Computer Systems Research Lab
Period 4, White

Daily Log

Detail for each day about what you researched, coded, debug, designed, created, etc. Informal style is OK.

Monday October 28th

Tried packing objects again, looked at images of rotated of objects (which I create for every object just to check that it generate ok), and it turns out my method for modifying the scale of rotated objects was wrong. I changed this method slightly, and the rotations looked a little better.

Tuesday October 29th

Tried packing/debugging all of class, as my code took so long to run so it took a while.
I found a small error in the placing object method that I thought I had corrected months ago. This is part of the reason it took me so long to find, as I thought that method was essentially working just as it should and investigated other methods much more in depth for errors.
The packing algorithm was *fast* - much faster than any other algorithm I had tried before this point.

Thursday October 31st

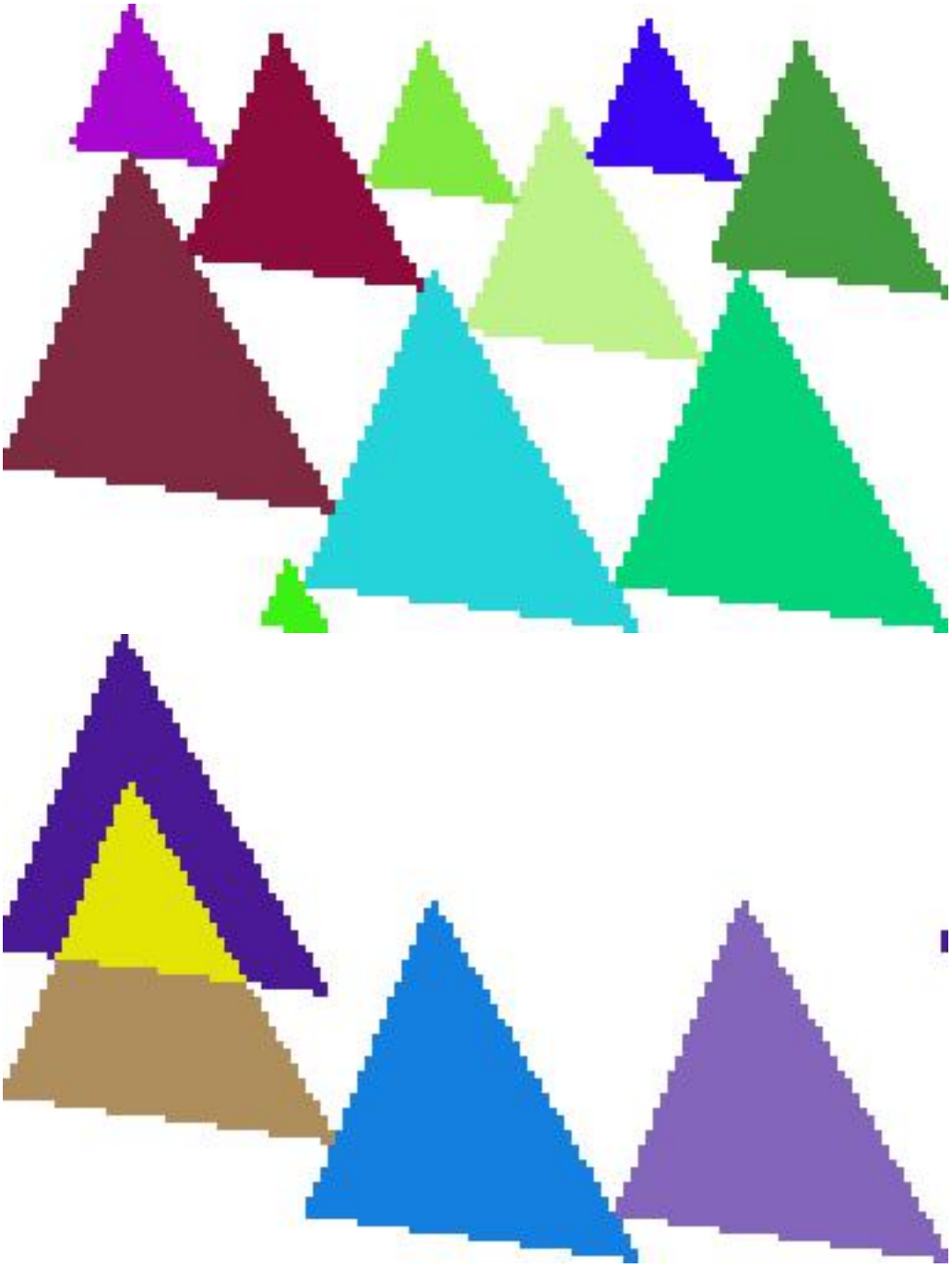
Most of class was presentations.
However, at the end of the class, I tried to pack several objects. It turns out that packings given for multiple objects were not correct - there are still some errors with the sizes of objects changing when you rotate them, and there was overlap of objects. However, due to the increased speed of the packing algorithm, these errors should be easier to fix.

Timeline

Date	Goal	Met
September 27th	I hope to figure out a method that efficiently finds the optimal rotation of an object by initially testing out several rotations of each object and then progressively gets closer to the optimal rotation through these tests, and begin work on it	I figured out that the best method for doing this is just by finding the optimal rotation for each object as you pack it, and any specific optimization algorithms would take too long
October 3rd	I hope to program the method found in the previous week	I did not program any method, but came up with several that would pack objects quickly and would be plausible
October 17th	I hope to be able to pack objects correctly and display an image of it	I haven't accomplished this yet
November 4th	I hope to be able to pack 10 objects in a single class period	I haven't been able to create correct packings yet, but given the current run time of my code this should be very achievable
November 18th	I hope to be able to pack objects correctly and have rotated objects be exactly the same size as the original object	
November 18th	I hope to be able to speed up my algorithm to the point that I can pack 40 objects correctly in a class period	

Reflection

I was finally able to figure out what was causing so many errors for me in the previous weeks - the packing method wasns't dealing with cases where it was impossible to pack an object correctly. I'm glad I could finally figure this out, and that my packing method is fast. Two packings are shown below (one clearly showcases that my code still has errors):



The rotation of the triangles is not the default rotation, so clearly the method is trying out each rotation. The fact that only one rotation is used seems to indicate there is a problem in how I'm scaling rotated objects - this rotation is likely much smaller than everything else.

The biggest challenge for me this week was the issue of rotating objects. When you rotate objects, since the width of the object in the image changes, you need to account for this since I use this width for scaling objects in the packing. I tried using trigonometry to find this scale before, but I did so incorrectly. I found out you need to consider two sets of diametrically opposite points in the rectangle, whereas I had previously been using one set.
One way to solve the rotation issue (which I'm surprised I hadn't thought of yet) is literally just

to use the Pillow rotate method and size finding method, and compare the width of the rotated image to the orignal image. This should be easier to implement and should also be more accurate than the trigonemtric method used.