I've always been interested by blackjack because it's the easiest of the gambling card games and I can actually learn it. When Mr. Kiang was giving examples for the project and said that blackjack was an automatic go I got excited because I knew that's what I wanted to do. It was, however, a really big challenge to start. I had no idea where to start and when I did research to figure out how other people had done it I found that I didn't like or really understand their designs. I ended up deciding I'd figure out how to write the game as I went along and did some preliminary research on how to beat the game and then wrote a standard deck class. None of the examples I found online for the deck class appealed to me; they used arrays instead of arraylists and had many methods for shuffling and rearranging the deck that seemed excessive and unneeded.

I started the deck with arraylists and then instead of having a shuffle method I just had the card returned picked randomly (used random to pick a number between 1 and however many cards were in the deck and picked that slot in the array). At first the deck was very complicated because I had no idea how to use arraylists. After a lot of research and help from my new best friend stackoverflow I was able to make a guide for all the basic commands with arraylists and I now feel relatively comfortable using them. After mastering arraylists the deck class was easy to fix. But unfortunately I still didn't know where to go with the game when I was finished with the first class. I decided the next basic thing I'd need to know how to do would be to get numeric values for the cards because the deck class returned a string.

CardValue was by far the easiest class to write but I had the biggest issues with it. Of course, when I finally figured out that the main error was that I hadn't made the checking in CardValue case sensitive, I felt really stupid but it was an easy fix when I finally figured it out.

From that point I decided to just start writing the game in BeatTheDealer, the driver class. Looking back I would have made several more class files because the driver file became very big. But at the time I figured it would be relatively short: all I had to do was deal cards and compare scores under or at 21. I set a layout with writing and comments and started to fill it in. The code got messy, fast. At one point I made more classes and tried to separate a dealer and player class but I just couldn't make it work and ended up keeping them all together in one. One of the hardest problems to fix was making it possible for an ace to be worth 1 or 11. It took me a while to figure it out but my solution was actually only a few lines of code. For the dealer, everytime he hit I had it check for an ace. If there was one, I checked to see if his score plus 10 (aces were counted as 1 in the code) would be ideal. I did the same basic thing for the player except I only counted it at the end when they said stand.

After the bulk of the code was done I started to try to organize it. The driver file was a complete mess so I tried to pull things out into more classes but the variables were so commonly used between everything that I wasn't able to do it. In order to try to clean it up from there I just segregated everything into methods to try to organize it and later organized the methods to a point where I'm happy enough with how it works.

Adding the AutomatedGame classes was a lot harder than I thought it would be. I assumed it would be short and easy and was terribly mistaken. I was able to use some of the code I had already written but there were enough differences that I had to rewrite it and couldn't just call the old methods. I ended up completely rewriting the Auto classes because they didn't work right and I figured it would be easier and cleaner than trying to fix what I had.

My game isn't complete but it's close. There are some really weird bugs that cause it to break under interesting circumstances and I'm still trying to figure out how, where, and why it does that. It gets the job done, just not as well as I would like. I just need more time with it.

My biggest obstacle overall was starting the code. If I had been able to plan it out better or had really been able to understand what I was trying to do before I did it the code would be so much cleaner and much more organized and I would probably be having less trouble with these last minute picky bugs.

The most help I received was probably from Aliya and Wyatt. I went to them when I had questions and sometimes they could help me and sometimes they couldn't but I would definitely be further behind if I hadn't had their help. Other than asking vague help-me-fix-this-error questions I didn't really collaborate with others. Makana Valdez was doing a similar project but I was always further ahead than him and he was writing his code very differently. It was interesting to compare mine to his but I didn't change things depending on it.

**Fun tidbit: had a little bit of inspiration when I realized I needed a word count on this piece and just wrote a quick little class file that would do that for me. Practical uses everywhere