

Blackjack Project Reflection

1. Make a copy of this document and make it editable
2. Save it as Blackjack Project Reflection - YourLastName
3. Complete the reflection
5. Download a .pdf of your reflection and include it with your project submission
6. Print a copy of your reflection and submit it on Monday March 11 in class

Category	Points
Reflection	/5
Submission Requirements -readMe -Works cited -includes .java source files -includes .java tester files -UML -Reflection	/1
Source Code -javaDocs -organized -commented when appropriate	/1
Program -Game play: Follows the correct rules and flow of the game Output is organized and easy to read and follow -Does it work? -Extension(s) -Creativity / Stretch previous skills	/23
Total	/30
Notes:	

- I. Compare and contrast your original UML to your final code. Discuss your findings and what led you to keep things the same and what led you to make changes.

My card, deck, shoe, and hand classes are all almost the same, but the dealer and player classes are very different than I originally had intended. The player class in the UML had a few methods that were not added to the player class or even the game at all. In their stead were other methods that may have performed different functions or similar functions to them. For example the won and lost methods were used to manipulate the number of chips which was not a previous function. In the dealer class has to be the most major differences because besides the most of the instance variables and the constructor everything else is different. The way I had originally written dealer in the UML was not taking into account the concept that the dealer would literally run the entire game upon creation. Which is why that class only has a few methods, but each method is considerably longer than the rest of the methods in any class due to their immense number of functions. I felt that the 10+ methods originally planned for dealer were not necessary and only 5 or 6 was much more ideal. However, the most interesting and cool thing is how it is kind of like a chain reaction from when the constructor is called all the way to the end because one method calls another until the player loses or is done playing. Lastly the dealer is also different because I was not originally planning to make the game have a multiplayer function which in turn created two extra instance variables, an extra constructor, and two extra methods. I find it really intriguing how different methods are used to run the single player game compared to the two player game up until the dealer's turn method that then runs identically.

- II. A list of all known bugs along with a description of either the error message generated or the logically error caused. Comment on why you were unable to correct the bug.

The only bugs I know of involve the scanner class for the number of chips and when you bet. These values I made to be ints only because you can not bet part of a chip. This sometimes might cause an error if the user enters in a double because doubles are not integers so it will cause the game to crash as a result. I was unable to correct this bug because I did not know how to continuously ask the user for a value until the intended type of variable was given. The methods I tried would not work because the game would instantly crash upon entering the wrong primitive type.

This is not a bug, but a comment that my game ends upon either player reaching 0 chips even if one player is able to continue playing.

- III. An overall reflection in which you discuss your program. Please reflect on:

- a. Particular parts of your project of which you are proud.
- b. Highlight any of the component(s) you added to the program that went above and beyond the basic requirements.
- c. Specific things that you found to be challenging to think through and code, things you learned from the project, things you probably would have done differently, and so on.

I am very proud of the two player mode that was implemented into my blackjack game because it took a lot of testing in order to remove over 30 bugs in the original code. That class took a lot of hard work and time because of all those errors and I am extremely happy that I was able to get the mode to work by the due date. Also I am proud of the overall project because I never would have thought how basic most of the code that is used in the project would be and that the project could have been done at a much sooner date as a result. I know that some people used either graphics or other self-taught concepts so they probably could not have done it sooner, but I did like how wide spread each and every individual's project was completed. The only two additional components I added to my game were the two player mode that I touched on earlier and five card charlie. Five card Charlie I do not really consider to be too much of a stretch because it only takes an extra method or two in the hand class and 5 or 6 additional lines in the dealer class in order to be used. However, the multiplayer mode is much more difficult and harder to complete because it largely depends on how the

single player original mode of your game was designed. If I knew I was going to make it multiplayer at the beginning I think that a lot of my dealer class would be different because I had no original intention of making it that way. As a result it took a lot more time to create because of the way I had originally designed it to be solely a one player game. Additionally I thought getting the aces to change from 11 to 1 was difficult and I am thankful to Zach for showing me a way to do that without having to manipulate other classes. His way of solving that problem was extremely helpful to me because his way completely different and honesty way easier than the four or five ways I tried to do it initially. Also I had trouble with clearing the hand in both the dealer and hand testers because I did not realize that if the hand was not cleared it would continue to build up cards forever. This was a funny bug that I had in dealer were I was on the third round, but my hands were starting in the 50's and 60's and not were the game was meant to start. Lastly, I learned how important the basics are because my entire project was done with basic java concepts besides the arraylist and scanner. Which as said earlier, I was shocked at how important those fundamental skills are to creating a game or project because you would expect the more complex concepts to be of more use when creating a large scale game.