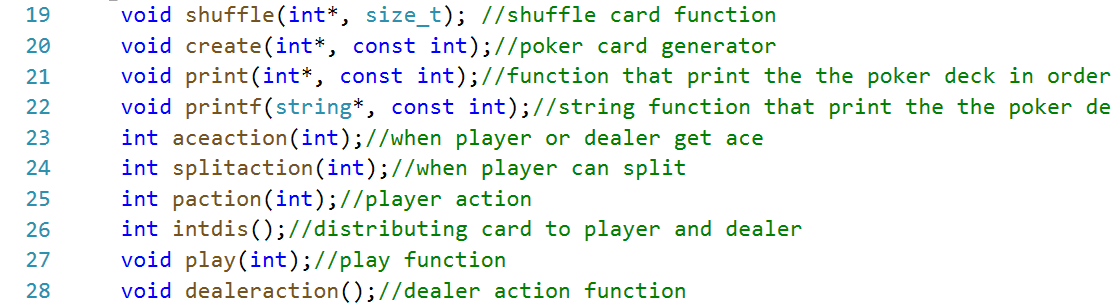
Personal Project

C++ Blackjack simulator

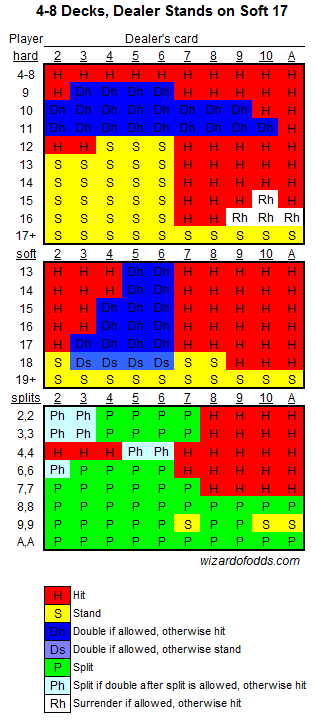
(computer vs computer) //we just sit and wait for the result



Objective: This program will stimulate the result for numerous games by using our adjustable strategy. By recording stats from computer vs computer while adjusting our bet, we can generate a must-win formula. This program will tell you, when you have more chance to win, and how you should adjust your bet!

\*\*The only reason I created this program was because I want to verify whether the must-win formula mentioned in Ben Mezrich’s book *Bring Down the House* exists. \*\*\*





**IMPORTANT!!!!**

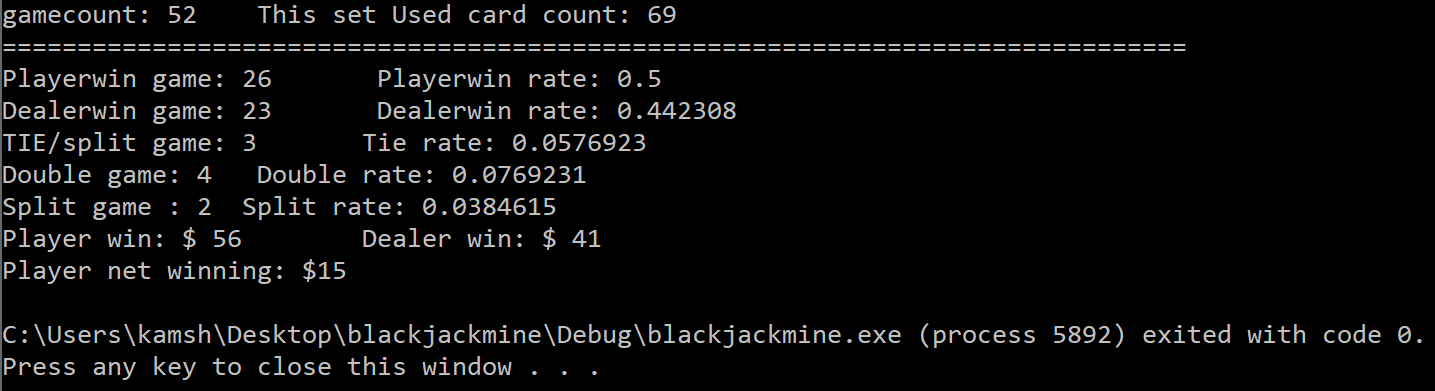
This is a strategy chart that we can found online. By following this chart, the players can maximize their win rate. However, **the win rate of the dealer would still be higher than the player** because that’s how blackjack designed.

However, we can still make money from the dealer if we can determine whether we are in an advantage situation.

In other words, you raise the bet when you know you have higher chance to win; lower the bet when you are more likely to lose.

This program will tell you, when you have more chance to win, and how you should adjust your bet!!!

Final result image



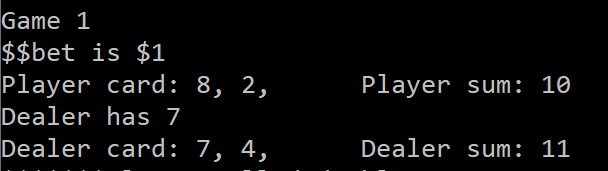
The final stats include:

* Numbers of game we played
* How many cards we used (important because this program is based on statistic)
* Win and tie rate of the player and the dealer.
* Split and double rate

Procedure overview:

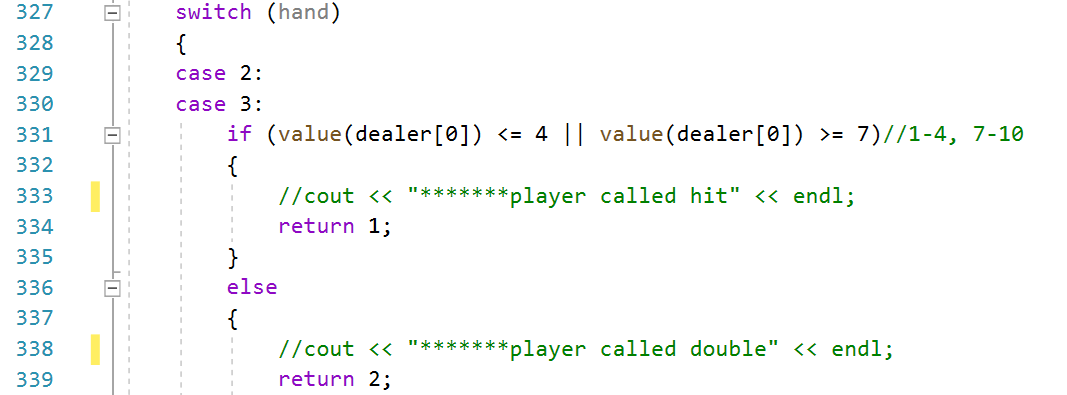
**#1 Write a simple 1v1 blackjack program, including card shuffling, card distributing, simple game rules.**

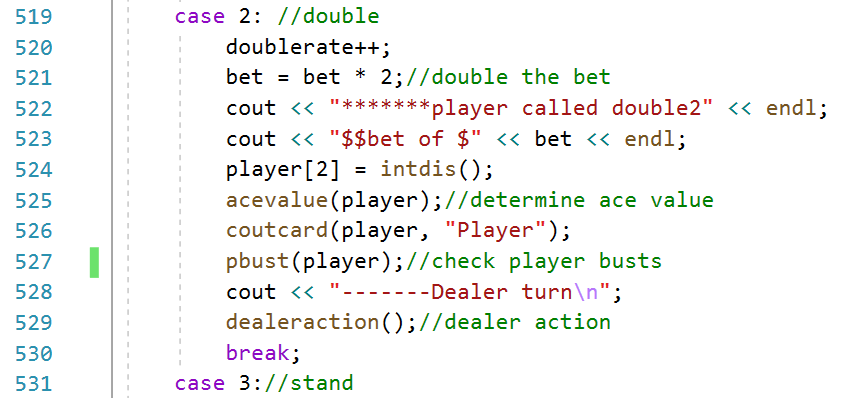


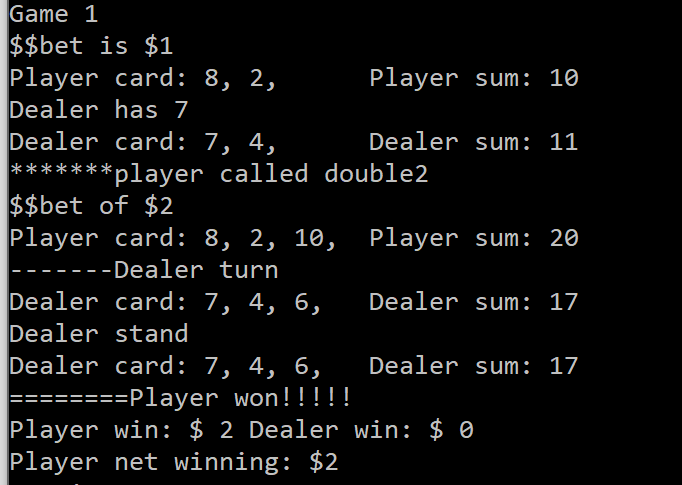


The program will display the player’s hand, the dealer’s exposed card and the dealer’s hidden card for research purposes.

**#2 Program the strategy chart to teach the computer how to deal with each situation.**

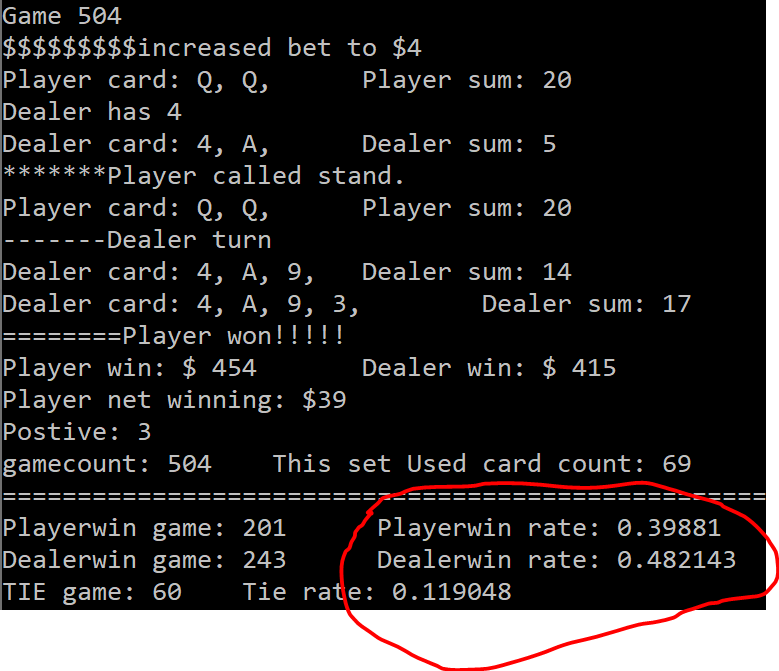






The program can decide what to do as a player for every game now.

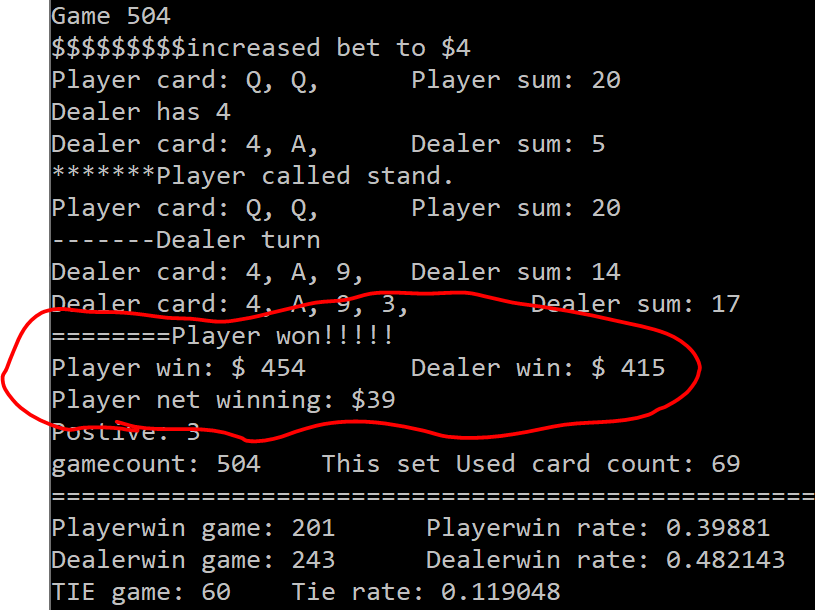
**#3 We can now calculate the win rate of the player after about 500 games**



The win rate of the player in 504 games is 40%.

PS. The maximum player win rate is about 40-43%. The strategy chart that we applied to this program already maximized the win rate. We will not be able to increase our win rate anymore.

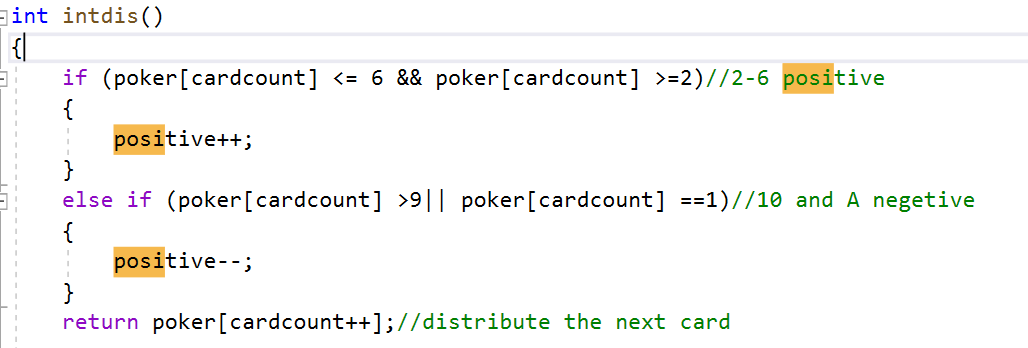
**#4 Even the player has a lower win rate, you can see the player is still winning.**



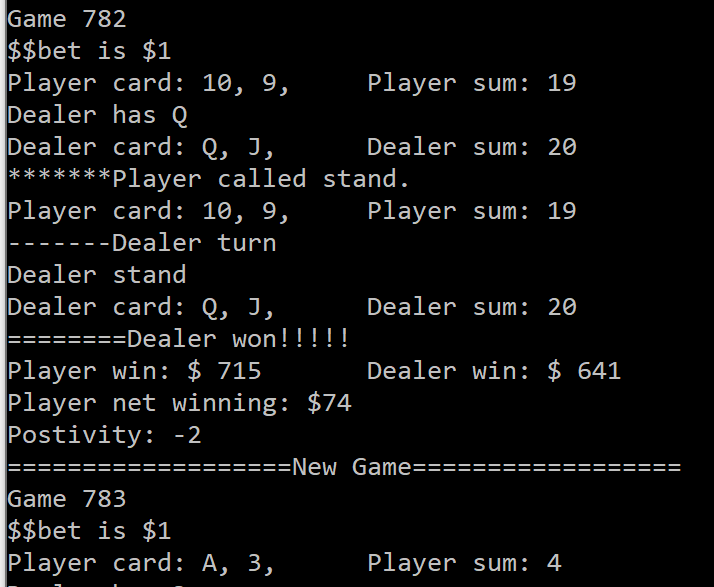
It is because we can change the bet when the situation is advantageous to us. For example, in Game 504, the system changed the bet to $4 automatically.

**#5 What defines advantage situation. Important!**

The more small value cards distributed, the more advantage we have. All cards that were distributed on the table must be shown to the players, so we can count them. We only need to identify whether smaller value cards distributed more than high value cards. In this program, the system will count for us.



Positivity = # of distributed small value card (2-6) (minus-) # of distributed high value card (10, J, Q, K).



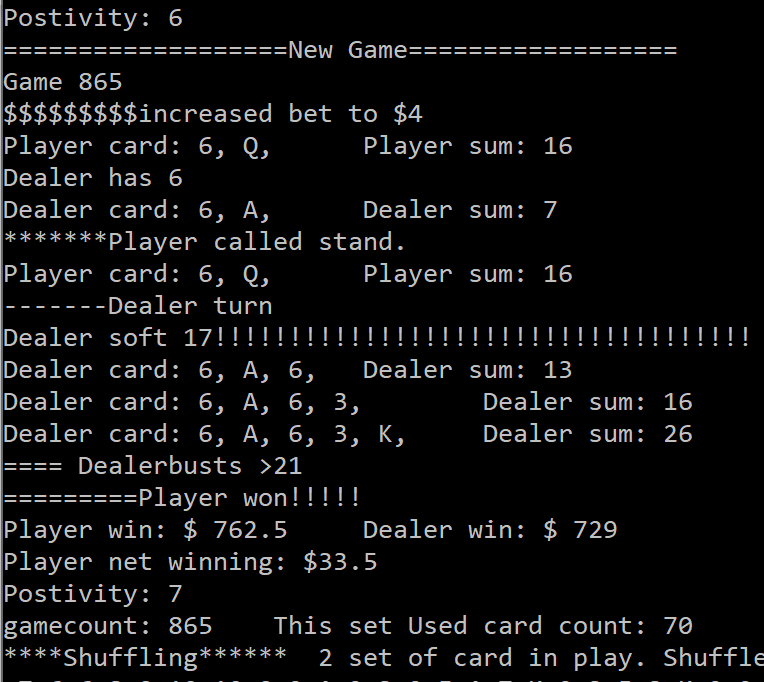


In game 782, the positivity is negative 2 (disadvantage) now, the program will only bet $1 for the next game.

Positivity will be shown after each game.

Positive=advantage

negative=disadvantage



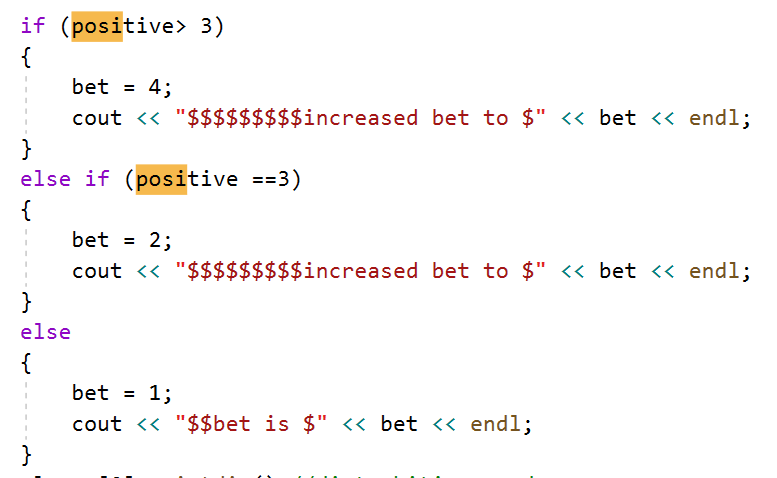


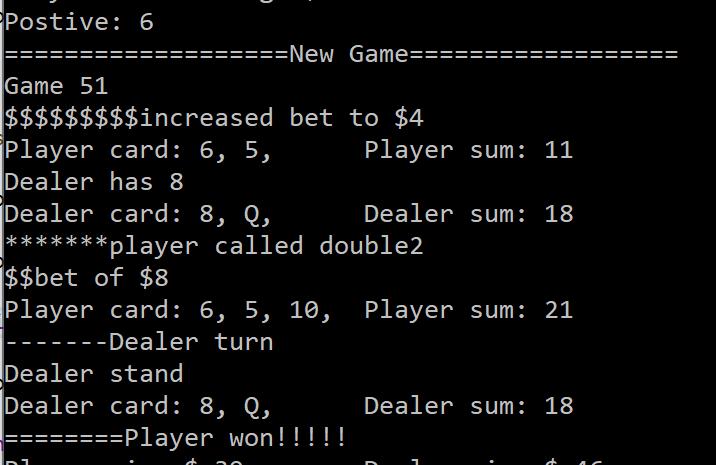
In game 865, the positivity was positive 6 by the end of last game. Theoretically, the player has a very high chance to win.



**#6. How do we adjust the bet?**

The positivity only indicates whether we have higher chance to win; it does not guarantee a win. Therefore, we need to develop our own bet strategy for how to adjust our bet. Changing the strategy is very easy; we can do it within 30 seconds. For example, we can double up the bet when positive = 3 and quadruple the bet when positive>3.



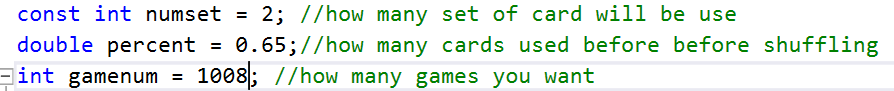


The system will automatically change the bet according to our bet strategy now.

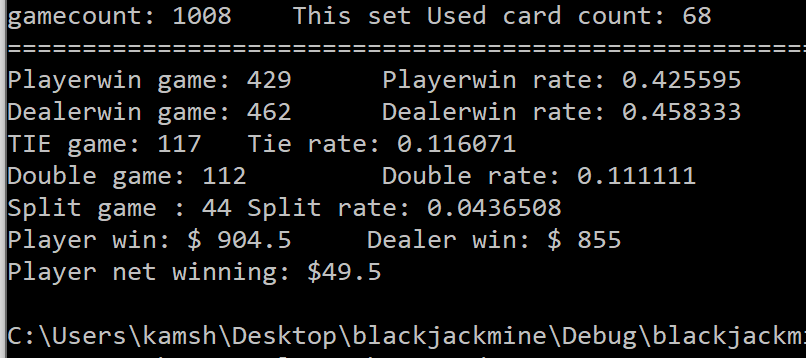


**Final step. Examine your strategy**

The stats are more accurate if more games were played. This is the whole point of my simulator; The program can simulate a hundred thousand games in 25 minutes. You can verify whether your strategy makes a profit using this simulator.



You can customize the card distribution because every casino has its own rule.



Conclusion: Using the example strategy in step six, the player earns $49.5 after 1008 games. Therefore, a must-win strategy does exist. According to Ben Mezrich’s book *Bring Down the House*, there were 6 MIT students who generated a must-win formula using this method (counting positivity) and won a billion dollars. Therefore, if you think the profit is too small, you can always go back to step 6 and adjust your strategy.

\*\*The reason I created this program only because I want to verify whether the must-win formula mentioned in Ben Mezrich’s book *Bring Down the House* exists. \*\*\*

I hope this program demonstrate my creativity and programming skills. I will keep develop more creative ideas, especially ideas that can benefit the community such as idea related to health and science. If you have any questions regarding this simulator, please do not hesitate to contact me.