CSI 370 Computer Architecture
Research 1 - Project

Jack Hannan

1 Blackjack

The project I decided to work was to create a Blackjack game that you could play by your self. I chose blackjack because I enjoy playing casino card games with my friends. I did not think that blackjack was too complicated a game to make and you can also play it by yourself, so there is some practicality, as I do not think I could create a game with multiple users. My goal at the start of this project was to create something similar to the picture below, it might not be as colorful. Over the semester I enjoyed coding in assembly so I wanted to do a project that would let me do a lot of that. The only code that that is not in assembly is getting user input, printing to the console, and shuffling the deck. Now there are some features of blackjack missing from the version I created and also the way it works is not the same as it would in a casino.

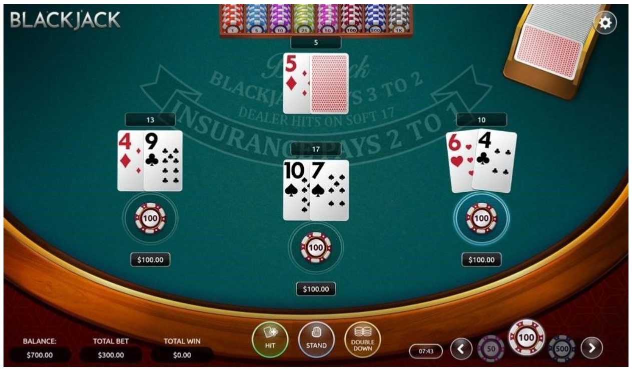


Figure 1: Example of what the game could look like

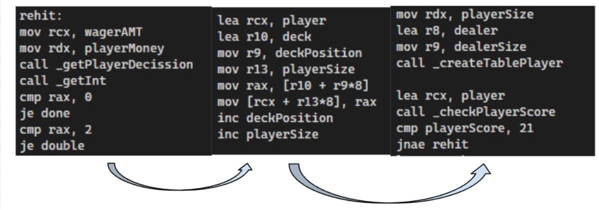
2 Blackjack Rules

Blackjack is a game of you verse the dealer even if there are other people people playing at the same table. The goal of blackjack to get to a combined card value of twenty one. If you go over twenty one you lose but if the dealer goes over 21 but you did not you win. If neither you or the dealer got to twenty one and did not go over twenty one then who ever is closest to twenty one wins. If you and the dealer have the same card value then its a draw and you do not lose any money. Unless you go twenty one with your first two cards in that case you would win and be get paid out with three to two odds instead of one to one odds. When it comes to the value associated with each card its not to difficult except maybe for the ace. The numbered cards are the value of what ever number is on them. For jack, queen, and king all those cards are worth ten. For an ace card it can have two values being eleven or one. So if you get an ace you can decided for it to be an eleven or a one and you can change its value when ever you want to. Blackjack is a fairly simile game though there are a lot of interesting complexnesses when it comes to probabilities. It game that seems like there would not be much strategy in but there is if you are able to keep track of the cards that used recently. If you are able to do that it can give you better insight on how you should go about playing your next if you ever get in a situation were you are contemplating getting another card.

The game play of blackjack is also fairly simple the games stares out with the dealers deals a card to everyone playing then to themselves faced down and then deals a second card to everyone face up including the dealer. Then after everyone has a card the player to the left of the dealer goes first in deciding what actions they want to make. The most basic actions are hit and stand where hit is where you ask for another card to help you to get to twenty, and stand is where signal you do not want anymore cards. Another option you could make is you can double down which is where you double your bet and you get only one extra card to help you get to twenty one. You can split your hand into two if you are dealt the same type of cards, like if you got dealt two kings, you also get dealt a new card for each hand and you have to make a new bet for your new hand as well. There is also something called insurance for when the dealer is showing that they may have a natural blackjack and if the dealer has a natural blackjack it pays two to one. Another thing to add is it standard for a dealer to stand once the get anything of value seventeen or higher. So another reason I decided to do my project on blackjack is I like all the complexities in an otherwise simple game.

3 Process

I used MASM and C++ as the language for this and I linked an .asm file to a .cpp file so that I could use both of those languages and it was done in 64 bit. The way I went about coding all of the different needs for this project was I would build on top of everything. So I would start with getting the initial user input, the getting the deck to shuffle, then deal the cards out and play the game of blackjack. One thing that took up some of my time was figuring out how I wanted to do things and trying to do as much as possible with techniques learned this semester and not relying to much on c++. Another thing that took up some of my time was trying to go for my more ambitions goals and not just get blackjack to work and worry about that later. I even changed the way the decked worked towards the end as I felt like the change made it easier to implement. A big part of my project is calling functions and passing variables to those functions especially the linked functions between MASM and C++. Excluding the main function in MASM code I have six functions but two of them are basically the same function one calculates score for the player the other for the dealer. The the two function dealing with player actions are similar as the player has more options and needs to get user input where the dealer has pretty set rules on what they do. Then I had a function that did an initial deal for the start of the game and a function that dealt with determine that outcome of the game. Then I had a function in the c++ file to shuffle the array, print to console the state of the table, get user input for variety of things, and also print messages to the user via the console.



```
rehit:
mov rcx, wagerAMT
mov rdx, playerMoney
call _getPlayerDecission
call _getInt
cmp rax, 0
je done
cmp rax, 2
je double
```

```
lea rcx, player
lea r10, deck
mov r9, deckPosition
mov r13, playerSize
mov rax, [r10 + r9*8]
mov [rcx + r13*8], rax
inc deckPosition
inc playerSize
```

```
mov rdx, playerSize
lea r8, dealer
mov r9, dealerSize
call _createTablePlayer

lea rcx, player
call _checkPlayerScore
cmp playerScore, 21
jnae rehit
```

Figure 2: Player Hits

3.1 Limitations

There are somethings that are in the normal game of blackjack that I did not include in my game of blackjack. I did not include the ability to make an insurance bet as I think it is a dumb bet so I did not want to put it in my version. I also did not include the ability to split your hand as I did not have enough time to do that though I would have wanted to do it. I did implement the ability to double down however there is no way of stopping a person from doubling down a second time. I wanted to only use one deck for my version as I fell like I am making a version of blackjack that you would play with your friends at someone's house and not at a casino where they use six to eight decks at each table they have of blackjack.

3.1.1 Arrays

A big part of the project was arrays. Which can cause some problems because making sure you are using the right array or the right counter to go through the array can be difficult to keep track of. I had three arrays one for the deck of cards, one for the cards the player has, and one for the cards the dealer has. Another problem I ran into is making sure I am working with player data and dealer data in the right spots since the code sections for them look very similar to each other. Which is annoying especially when coding in MASM where those issue might not be as easy to compared to higher level languages where colors allowing them to pop out and be more visible.

3.2 Dealing Cards

When it comes to dealing cards to the player and the dealer you have the initial deal and then the in game deals. The initial deal is pretty straight forward you go through the deck array and you alternate between the player and dealer twice. Then when it comes to deal cards to the player and dealer in the game it is a little more complicated as there are more things that need to be kept track of. You have to keep track of the position you are in dealing cards from the deck and also the position to where you are going to the payer's or dealer's next card to because otherwise could could replace a card dealt instead of deal a new card. One of the most import things when it comes to keeping track of those positions is to reset them when you start a new game otherwise everything will messed up.

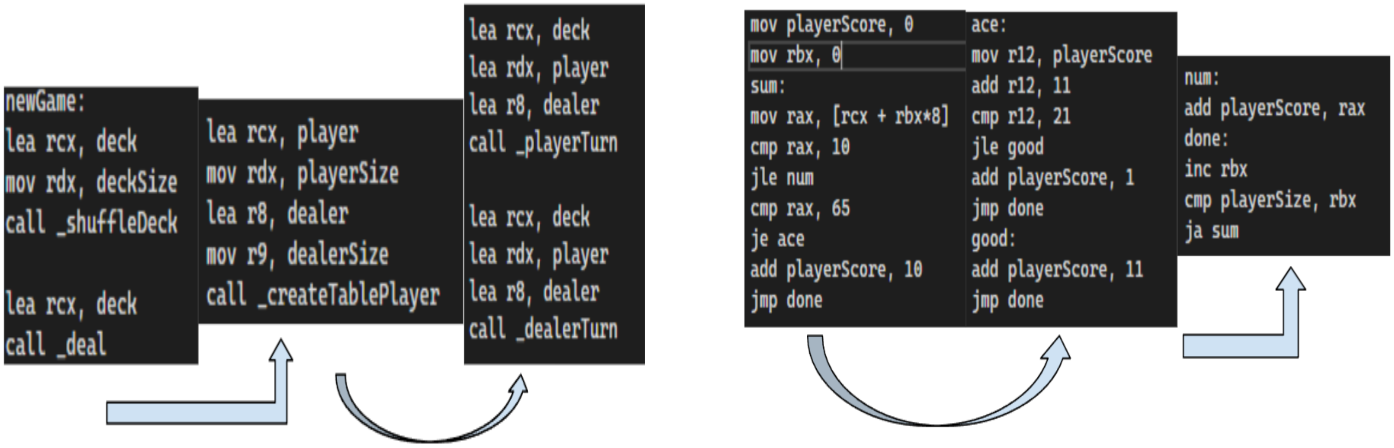
3.3 Checking Total Card Score

This part was a bit tricky for me at first because initial I had all of the card values as string so figuring out how to convert the ASCII values into numerical values was tricky. Then I realized that since I am just outputting the the card values to console I do not need to store the numbered cards as strings and then just check for if the an ace comes up because the rest jack, queen, king are all worth ten. Dealing with the ace was interesting because I did not want to include any more arrays into the program so dealing with the fact that the ace is two different values at once was a bit annoying. I stored the value of card score in an unused register, as I did not want to alter the card score before determining what value I wanted to give the ace, and added eleven to it and seen if it cause the score to bust. If adding eleven caused a bust I gave the ace a value of one other the ace's value was eleven. Also to deal with the fact that one turn you may wont the ace to be an eleven and the next it do be an one I reset the card score every time I went to check it.

4 Challenges and Solutions

4.1 Shuffling

One of the challenges I faced was shuffling the array acting as a deck of cards. At first I was trying to do it using intrinsics but I didn't really understand how to do it because I was shuffling strings and the shuffle was for floating numbers. I realize now that it probably would have worked because the the array gets passed over as ints. Though another thing that confused me was you were only able to shuffle a small portion of an array at a time. So I then went to try to figure how to write a shuffle function in MASM and realized that I did not have enough time for this so I decided to just have a built in function in c++ do the shuffling for me to solve my shuffling problem.



```
newGame:
lea rcx, deck
mov rdx, deckSize
call _shuffleDeck

lea rcx, deck
call _deal
```

```
lea rcx, player
mov rdx, playerSize
lea r8, dealer
mov r9, dealerSize
call _createTablePlayer
```

```
lea rcx, deck
lea rdx, player
lea r8, dealer
call _playerTurn

lea rcx, deck
lea rdx, player
lea r8, dealer
call _dealerTurn
```

(a) Flow of the Game

```
mov playerScore, 0
mov rbx, 0
sum:
mov rax, [rcx + rbx*8]
cmp rax, 10
jle num
cmp rax, 65
je ace
add playerScore, 10
jmp done
```

```
ace:
mov r12, playerScore
add r12, 11
cmp r12, 21
jle good
add playerScore, 1
jmp done
good:
add playerScore, 11
jmp done
```

```
num:
add playerScore, rax
done:
inc rbx
cmp playerSize, rbx
ja sum
```

(b) Calculating Player Card Value

Figure 3: Two Examples From Code

4.2 Displaying the Table State

Initially I wanted to have some sort of graphical competent to my project but I learned I did not have enough time to research all that was need for this. As I could use the message box but I think it would look weird clicking buttons to have you do something in blackjack but it not be label that in the message box. Also it took me a bit to realize this and I was trying to create a big string to serve a the message in the message box and I kept running into trouble there of converting the ints passed to the function into characters. To solve this issue I decided I was going to just print the table state to console and just cout everything. Another issue I had related to the table state was that I wanted to show how much money the player had and how much they were betting. I kept running into problems with this as I was already passing four variables to that function so those two variables had to be passed to the function from the stack. I kept having problems passing variables through the stack. So I just make a new print to the console function and passed those two variables through by themselves.

4.3 Functions Not working

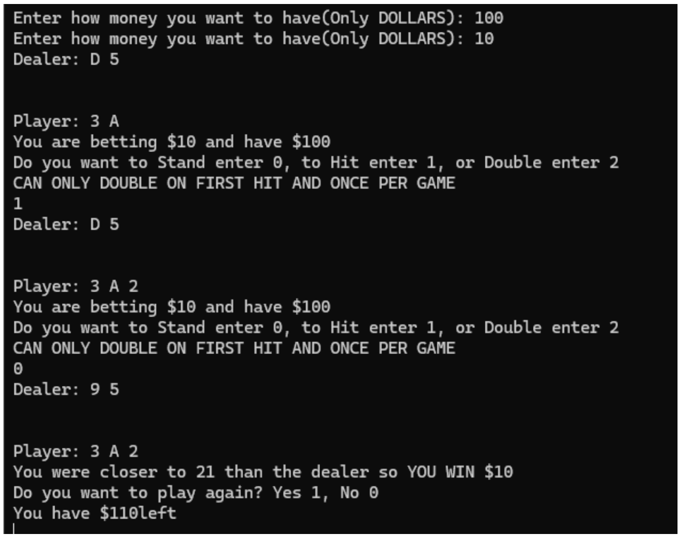
I started working on the project with the deck on being thirteen cards one of each card can my functions seemed to be working. Then I attempted to have the deck be a complete deck of cards but I got an error saying that the deck array was too big. So then I decided to use half a deck and it ran but after that everything with my code has been messed up. Especially when it comes to the functions pertaining to the dealer. I have not be able to figure out why and how to fix it. So I have the deck back to thirteen cards on of each card but the dealer functions still do not seem to work.

5 Things I Would Have Liked to Have Done

If I had more time to work on this assignment I would have liked to polish it up a bit more and understand exactly why I am getting so of the problems I am having. Also I think it does not print out ten properly it just prints out 0. So if I had more time I would have liked to do a deep dive into debugging all of the functions I use and getting all of the code to work properly. I also would have liked to implement more things into the program like be able to split you hand when given the opportunity to do so. I would have liked to properly implement the double down feature as the way it is right now its not properly implemented. The possible out comes from doubling down are you bust and lose double your wager, you beat the dealer and just earn your wager, or you lose to the dealer and lose your wager. Also there is nothing stopping some one from doubling down after they hit which is not allowed in blackjack. I also would have liked to fix the dealer because for some reason it does not seem like the dealer want to hit ever. This makes it not feel like you are playing against anyone if the dealer never hits as the stakes of hitting when you have fifth teen or sixteen are lower because you know the dealer will stay with whatever they got. I also would have liked to incorporate some sort of graphical competent into the game. Instead of the user typing in their input of what they want to do in the game have them press buttons of what they want to do in the game. I also would have liked to make the way the output of the program looks better. I think a big one is not spending so much time on researching and learning things that I ultimately did not even use because if I could of had that time to fix the problems in my code now I think my program would run a lot smother.

6 Visualizations

Here are some results of the current state of the code



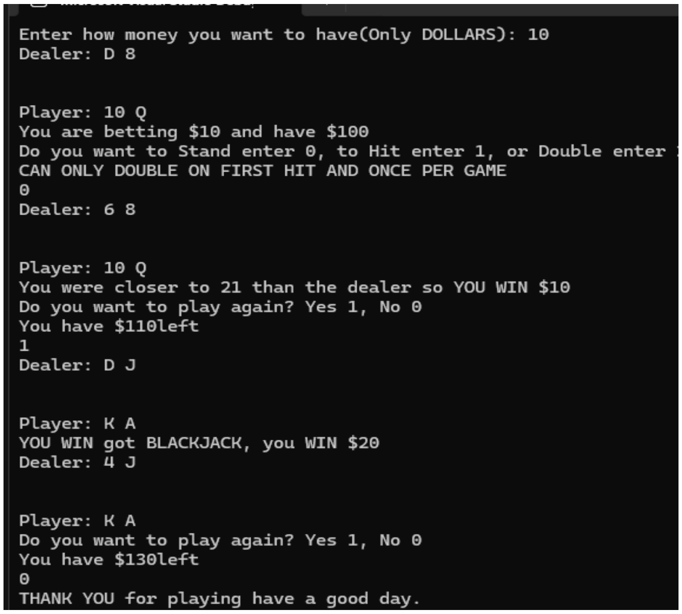
```
Enter how money you want to have(Only DOLLARS): 100
Enter how money you want to have(Only DOLLARS): 10
Dealer: D 5


Player: 3 A
You are betting $10 and have $100
Do you want to Stand enter 0, to Hit enter 1, or Double enter 2
CAN ONLY DOUBLE ON FIRST HIT AND ONCE PER GAME
1
Dealer: D 5


Player: 3 A 2
You are betting $10 and have $100
Do you want to Stand enter 0, to Hit enter 1, or Double enter 2
CAN ONLY DOUBLE ON FIRST HIT AND ONCE PER GAME
0
Dealer: 9 5


Player: 3 A 2
You were closer to 21 than the dealer so YOU WIN $10
Do you want to play again? Yes 1, No 0
You have $110left
```

(a) Result of One Game



```
Enter how money you want to have(Only DOLLARS): 10
Dealer: D 8


Player: 10 Q
You are betting $10 and have $100
Do you want to Stand enter 0, to Hit enter 1, or Double enter
CAN ONLY DOUBLE ON FIRST HIT AND ONCE PER GAME
0
Dealer: 6 8


Player: 10 Q
You were closer to 21 than the dealer so YOU WIN $10
Do you want to play again? Yes 1, No 0
You have $110left
1
Dealer: D J


Player: K A
YOU WIN got BLACKJACK, you WIN $20
Dealer: 4 J


Player: K A
Do you want to play again? Yes 1, No 0
You have $130left
0
THANK YOU for playing have a good day.
```

(b) Result of Two Games

Figure 4: Output From Playing Blackjack

7 Sources

- Google AI when you search shuffle vector c++ on Google