Michael and Breeze’s Blackjack Course Project

Breeze Tucker   
*Computer Science   
Ursinus College*Collegeville, USA  
brtucker@ursinus.edu

Michael Connors  
*Computer Science   
Ursinus College*Collegeville, USA  
miconnors@ursinus.edu

# Introduction

For our second course project, my group was tasked with creating a game of Blackjack going against a computer opponent. Blackjack is a card game where you bet money against an opponent before each of you are dealt two cards. If the sum of your hand ever exceeds twenty-one, then you must hand your bet money over to the opponent. However, you need to get as close as possible to twenty-one because whoever has the smallest sum in their hand must give up their bet money. Knowing this, you can choose to keep your hand as it is, add another card to try to get closer to twenty-one, or forfeit the turn to your opponent. This process will repeat until a player is out of available funds or gives up, declaring the other the victor.

Getting started is always the most difficult part of coding, and that certainly was the case for the second course project. When there are so many things that you must implement, it is difficult to know what to start with. My group decided to tackle this problem by breaking the project into many shorter files that we would combine at the end. The individual programming assignments helped us to achieve this by setting smaller goals for ourselves as we worked.

# Software design decisions

Our group decided to split our workload into five main groups based on the individual programming assignments. Michael was assigned to tasks 3.1 and 3.4, Breeze was assigned to tasks 3.2 and 3.3, and we both collaborated on 3.5. Michael ended up combining all of our code, debugging it, and filling in anything we missed.

I, Michael, decided to start with task 3.1, which was to program visualizations of the card values on screen. I also had to visualize bets and wins onscreen, but decided to hold off on it since I was assigned to task 3.4 which includes that anyways. I did not want to be overwhelmed by too much at once for my starting point. Task 3.4 was to continue with the visual aspects of the project and create the text user interface. I never ended up creating a specific file for task 3.4, but I worked it into the project as I moved along. I chose to take on combining our code and creating many other parts that were not in the individual programming assignments. Though, I only did that because the due date crept up on us too quickly. It ended up skewing the distribution of labor immensely.

# Implimentation

. The coding process was streamlined through the use of reusable procedures. By putting frequently used lines of code into a procedure which can be called from anywhere, it allowed us to continue coding without worrying about the same problem repeatedly. The ability to return to the place the procedure was called allows them to be used without worrying about messing up the order of the code. However, you cannot call a procedure based on a condition, so it was very inconvenient to create a new label that only calls a procedure every time a condition had to be met. A few examples of procedures that were used many times throughout the project are input\_int, which takes a word-sized number from user input, int\_to\_ascii, which converts an integer into a string of ascii values, rand\_num, which generates a random number, and string\_print, which prints a string from memory.

Since the game of Blackjack repeatedly goes through the players turn, the game order was implemented using a large loop. After the starting values are configured, the program goes through the steps of Blackjack, starting by initializing player bets and dealing cards. Then, the player inputs a number based on whether they want to keep their current hand, add another card, or forfeit the turn. After the player is done, the computer opponent decides on one of those choices as well. Next, each player’s hand is evaluated to determine the winner of the turn. Finally, the turn requirements are checked, and if someone is out of money or gives up, the game loop is over, and a winner is declared.

# Discussion and Conclusions

I, Michael, am very proud of this project. Since I ended up creating the finished product, I ended up becoming very attached to it. This assignment really helped me to understand assembly code when I was struggling greatly beforehand. Professor Núñez-Corrales was also a great contributor to my understanding through individual meetings.

However, we took too long to start this project. Since the end of the semester is filled with many other finals, we did not have much free time to work on the course project. We only ever got a chance to meet up once to split up the responsibilities. This led to I, Michael, working for fifteen hours straight on coding and writing parts of this essay on the last day before it was due. Some of the code was even committed a few hours past the due date at midnight. In the future, we need to work on starting earlier and managing our time better through schedules.

##### References

No outside references were used.