Project Summary: Blackjack - Group 11

Group Members: Alexander Mohler Homeschool, Phoenix
Chad Sweeney Oldham County, LaGrange

Blackjack is a popular card game which we adapted into Python using the pygame libraries. Blackjack involves repeatedly drawing cards in order to try and get as close to a combined value of 21 as possible without going over. If the player gets a better score than the computer controlled dealer, they will win the round. The player can also bet money on each round, winning a profit of that much if they win and losing that much if they lose. The ultimate goal of the game is to go from a starting amount of 100 dollars to a final amount of 5000 dollars.

We adapted Blackjack in Python. Python is a popular, high level, object oriented programming language. In order to write code, we used the integrated development environment PyCharm. PyCharm is a free integrated development environment which allows for easy editing and running of Python programs. The game itself heavily relies on one of Python's built in libraries known as pygame. Using pygame allows for the creation of a GUI, or Graphical User Interface, which allows the user to see and interact with the game in a visual and intuitive manner as opposed to typing in the console which, while functional, does not create a good user experience.

We developed our project with a Model View Controller, or MVC, style. Using MVC is preferable for games not high in action such as Blackjack. One major challenge we faced while implementing the game was counting the score of the player. The intended purpose of the faulty code segment was to add the value of a given card to the player's score whenever they drew one. Instead, it resulted in adding the value of every card the player had previously drawn to their score whenever they drew one which resulted in much higher scores than what was intended. We attempted to implement a fix, however, this resulted in the score being set to the value of a drawn

card, rather than being added. Eventually, we managed to get it working after much debugging. Another major challenge was dealing with a long running bug involving drawing a card from the deck that didn't exist. After many hours of trying to figure out why this happened, we determined that the drawCard() function was not quite functioning correctly in regards to shuffling the deck when it ran out of cards. When the game needs to draw a card and there are no cards left, it needs to reshuffle the deck and draw again. This was accomplished using recursion, however, there was not a return statement on the recursive call for the drawCard() function. As such, when the game needed to draw a card in order to add it to the list of cards drawn by the player or dealer, and the game needed to shuffle the deck, the game would not pass a card back to the list resulting in drawing a nonexistent card. In addition, implementing multiplayer was a large hurdle since our code at the time was not built with more than one player in mind. The biggest obstacle was implementing aces. In Blackjack, an ace is worth either 11 or 1 depending on the situation. This turned out to be extremely difficult to implement, however, we have a mostly working system in place meant to determine an aces' value.

In conclusion, we would consider our project a success as we successfully implemented a fully functional version of Blackjack in Python. Although there were a lot of struggles along the way, we managed to pull through in the end.