Peyton Kelly Progress-Report:

What data structures are you going to use in this project?

The data structures I plan to use will be a dictionary and lists this way I will be able to store the types of cards and be able to incorporate highs and low aces. The dictionary will hopefully be used as a storage for the cards used and then allow for resetting. The classes have been set up for the suits, ranking of the card number, and the value. Suits and ranks are set up as lists because they are homogeneous and the value is set up as a dictionary because it has strings and integers.

My class setups will probably be for setting the certain decks and generating the cards using the randoms one for the dealer and user. One class that I need to make will be for the dealer to whether to hit or stand. This will be with the ints because the dealer must stand at 17 or higher. One of the problems I have encountered is withing my chips and betting standard. I would like to have players be able to win as much as they can and be able to leave the table at any time for now I have just put in a max bet rule of 2500. This makes it possible to go all-in from the start.

From the beginning of this project, I believed that the code and implementation would be challenging. Once the starter code was figured out like my class set up and inheritance it began to be clearer on what came next. This was one of the first things that I really had to map out on paper and realize the outcomes I wanted to achieve before starting to build the code. My original outline for the code was going to be very vague and simple. I was just going to have 2 random numbers from 1-11 generated as the player’s cards and have a similar implementation for the dealer’s cards.

I have completed most of my code in my text editor. Right now it is just coming to tinkering with the code to make things better and create the best possible outcome and making sure everything works properly. Right now I have the game set up as playing a hand and then asking if you would like to play again if yes, you will continue but the overall chips of the player is seeming to give me problems. I am pondering creating a new class that sets the player's chips after the game and replacing the new hand with the new class so that way it will be able to keep adding the chips and keep hold of the total number of chips. I want players to be able to know how much they would be leaving with. As I am writing this I will be toying with the betting class and all the methods because I want to have additional payouts luck double queens ("Lucky Ladies") pays 20 to 1.

Creating the hand class was one of the more difficult things in this project. This is because this is where the problem of aces high or low at values for 11. This was particularly challenging because you must adjust the original value of the ace depending on the total the player has. Aces high at 1-10 and aces low when 11-20.

Then it came to one of the strategy parts of the game of blackjack. To hit or stand is an important question that carries a heavy weight. This is to add another card to the hand at play or stay at the current number one is at with cards in hand. Hitting on the cards at hand can make or break the game for you as the player because it is impossible to bust with the first two cards. I have implemented this part of the code by creating inputs that the user must answer unless blackjack is achieved. All of these parts of code are put and inherit their parents the deck and hand because you cannot have repeating cards within the same hand. Most of this game is depending on if

statements and them being able to take care of all possible answers like the answers expected and value errors or just the user messing up their input to not ruin the hand for a player and cause a crash.

I have been testing the product many times over to make sure that the game does not crash if something happens. I have been trying to crash the game by doing things like messing up inputs and the new hand function. Now for the opponent, the dealer, or in our case the computer and code. This game is not impossible to win. The dealer does not win in all cases. The dealer has a specific rule set they must follow to play the game correctly just like the user must follow. All the win scenarios are built out and including the push meaning a tie is conducted to play a new hand directly after with the same bet because a winner must be declared for each game/round.

One of the more satisfying things to map of the game was easily the display. I enjoyed having the freedom to how my game would turn out such as the spacing and all the choices I could make such as displaying the hidden card for the dealer's second card. The flipped card carries an important choice because it is hoping that the dealer has low enough where one feels safe to hit freely about them having blackjack or put the player in a terrible situation to hit.

This game is mostly up to pure luck with what the computer dealer decides to give the player but this how the advantage of the strategy of the game is able to be played. Although most casinos look out heavily for people doing things like counting cards this is one deck that is shuffled for each round and never be able to guess what card is next with certainty because the cards will never be physical it is just numbered and letters attached to values. I feel that I am in a good position to be able to present

this project because there are multiple endings and you can always come out of drought but your choice to stay or leave.