James Palmer

Assignment 4

1) Created three files titled randomtestadventurer.c Randomtestcard1.c and randomtestcard2.c. Randomtestcard1.c is testing smithy card. Randomtestcard2.c tests for the village card.

2) For my improvements for randomtestadventurer I gave a random deck size and then pull a random hand size out of that given deck. I then randomly seeded copper, silver, and gold into the deck with random cards filling in the rest of the cards. The next step is to run the adventurer. After adventurer runs I check the amount of coins in the players hand now versus before the card was played and threw an error if a maximum of two coins were not added to the hand. I then checked each of copper, silver, and gold to see if any had been discarded with the other drawn cards. This is significant because if a coin card made it into the discard pile then, it was not properly source by the function.

   For randomtestcard1 I chose smithy. For this card I gave random deck and hand sizes again. But this time I tested hand size before and after testing that after smithy is discarded that the net gain for the hand should be +2 cards after the 3 cards are added to the hand. There was an issue with this where it turned out smithy was not initially placed in the discard pile but played card pile. A quick variable reassignment fixed that. Lastly I tested to ensure that the deck has decreased by the appropriate amount. When I created the function for smithy called smithyEffect(), I input a condition that the card would only be drawn. If the position in of the index in the loop was less than the handpos index that comes into the function as an argument. Visually this is something simple in code yet very complex from just looking at the game visually.

   For randomtestcard2 I choose the village card. However, this time I used to randomize the struct (as seen from one of our lectures) rather than state a default array and take its expected value from there. This way I can generate the hand count from the chosen random player instead of a default hand size. Allowing for variable generation when pulling from other player's card usage (i.e. lower our initial hand size). Lastly, I insured that our deck count was constantly updated with proper game state. This allows for more variable testing that is more in-depth rather than default testing and assuming default game mechanics are occurring. You can see in the code I added a broader function of player hand and actions updates. This can allow for a user to use more than one village card in a turn (leading to 3 possible actions rather than 1). In my past card test, this only accounted the default under the assumption that only one extra action can occur after a village card is played. This is important because it allowed for more layers of testing of the village card (as seen in the code). I am able to input more levels of expected output with different player card values.

3) Updated makefile with .out files.