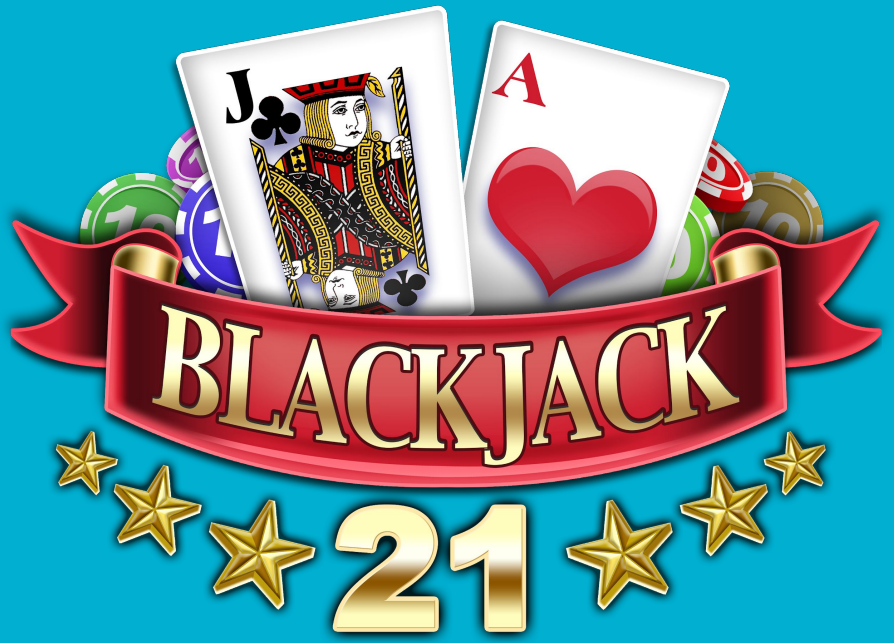
Blackjack

—

By: Ghasif Syed, Tong Trinh, Yehua Chen



Introduction

- Create a server and multiple clients network program to allow multiple players to join the game.
- Each of the players is going to play their own game, they are not playing the same game together.
- Set up all the corresponding values for cards 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, and Ace. The total number of the cards is 52.

Challenge for us

- Make sure all the clients are able to connect/communicate with the server.
- Think about how to determined the winner of the game.
- Allow players to restart a new game without disconnecting and join the server again.
- Make sure the values are correctly added together.
- Ace is a special card in this game. Ace can be 11 or 1. We have think about all the possible situations when a player/dealer has an Ace or Aces in the hand.

Description of the program

- The relationship between Server/Client, TCP, and Socket Programming.
- Use multithreading to allow multiplayers to play in one game.
- Server functions:
  - Wait for connections from the client.
  - Receive messages from client.
  - Send the messages back to all clients.
- Client functions:
  - Connect to the server.
  - Receive messages from the server.
  - Send messages back to the server.

Possible results for the program

```
DEALER: 3 of ♣♣♣; (HIDDEN CARD)
PLAYER: King of ♠♠♠; 2 of ♣♣♣
Dealer Hand Value is: 14
Player Hand Value is: 12
Do you want to hit or stand?
Give your input: hit
DEALER: 3 of ♣♣♣; Ace of ♠♠♠
PLAYER: King of ♠♠♠; 2 of ♣♣♣
Dealer Hand Value is: 17
Player Hand Value is: 19

Dealer Hand:
=========
|3      |
|   ♣   |
|      3|
=========
=========
|A      |
|   ♣   |
|      A|
=========
=========
|A      |
|   ♣   |
|      A|
=========
=========
|2      |
|   ♣   |
|      2|
=========

Player Hand:
=========
|K      |
|   ♠   |
|      K|
=========
=========
|2      |
|   ♣   |
|      2|
=========
=========
|7      |
|   ♣   |
|      7|
=========


YOU WIN. Your score is higher than the Dealer
Press "ENTER" to start a new game!
Give your input:
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                NEW GAME
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
DEALER: Ace of ♥♥♥; (HIDDEN CARD)
PLAYER: 8 of ♦♦♦; 6 of ♥♥♥
Dealer Hand Value is: 21
Player Hand Value is: 14
Do you want to hit or stand?
Give your input: hit
DEALER: Ace of ♥♥♥; 10 of ♥♥♥
PLAYER: 8 of ♦♦♦; 6 of ♥♥♥
Dealer Hand Value is: 21
Player Hand Value is: 24

Dealer Hand:
=========
|A      |
|   ♥   |
|      A|
=========
=========
|1      |
|   ♥   |
|      1|
=========

Player Hand:
=========
|8      |
|   ♦   |
|      8|
=========
=========
|6      |
|   ♥   |
|      6|
=========
=========
|1      |
|   ♠   |
|      1|
=========


YOU LOSE. Dealer got a Blackjack
Press "ENTER" to start a new game!
Give your input:
```

```
DEALER: 8 of ♦♦♦; (HIDDEN CARD)
PLAYER: 7 of ♦♦♦; Queen of ♦♦♦
Dealer Hand Value is: 10
Player Hand Value is: 17
Do you want to hit or stand?
Give your input: stand
DEALER: 8 of ♦♦♦; 2 of ♥♥♥
PLAYER: 7 of ♦♦♦; Queen of ♦♦♦
Dealer Hand Value is: 17
Player Hand Value is: 17

Dealer Hand:
=========
|8      |
|   ♦   |
|      8|
=========
=========
|2      |
|   ♥   |
|      2|
=========
=========
|7      |
|   ♠   |
|      7|
=========

Player Hand:
=========
|7      |
|   ♦   |
|      7|
=========
=========
|Q      |
|   ♦   |
|      Q|
=========


ITS A TIE. Push!
Press "ENTER" to start a new game!
Give your input:
```

Python Code: Server

```
@author: Ghasif, Yehua, Tong
"""

# Imports
import socket
import random
from _thread import *

# Server setup
ServerSocket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
port = 1233
ThreadCount = 0
ServerSocket.bind(('', port))

# Wait for a Connection
print('Waiting for a Connection..')
ServerSocket.listen(10)

# Thread function
def threaded_client(connection, num):

    # Declare the suits/ranks of cards along with the values associated with them
    suits = ["♠♠♠", "♦♦♦", "♥♥♥", "♣♣♣"]
    ranks = ["2", "3", "4", "5", "6",
                  "7", "8", "9", "10", "Jack", "Queen", "King","Ace"]
    values = {"2":2,"3":3,"4":4, "5":5, "6":6,
                  "7":7, "8":8, "9":9, "10":10, "Jack":10, "Queen":10, "King":10,"Ace":1}

    # Array that represents the Dealer's Hand
    dealer_Hand = []

    # Array that represents the Player's Hand
    player_Hand = []

    # Boolean to represent if the game is currently being played or not
    playing = True
```

```
# Function that sums up the player/dealer hand (array) passed in and returns the total value
def get_Value(hand):
    total = 0
    aces = 0
    for x in hand:
        val = values[x[0]]
        total += val
        if (x[0]=="Ace"):
            aces+=1
    while total <= 11 and aces:
        total += 10
        aces -= 1
    return total

# Function that formats a specific card from the player/dealer hand (array) and index (int) passed in and ret
def show_Card(hand,index):
    ret = hand[index][0]+" of "+hand[index][1]
    return ret

# Function that formats the player/dealer hand (array) passed in and returns string value of ASCII art cards
def print_Deck(hand):
    if (hand == player_Hand):
        ret = "Player Hand: \n"
    if (hand == dealer_Hand):
        ret = "Dealer Hand: \n"
    card = ""
    maxLen = len(hand)
    for x in range (maxLen):
        rank = str(hand[x][0])
        suit = str(hand[x][1])
        card = "=========\n|"+rank[0]+"      |\n|   "+suit[0]+"   |\n|      "+rank[0]+"|\n========="
        ret += card+"\n"
    return ret

# Function that resets the player & dealer hand and adds 2 cards to each hand
def starting_Hand ():
    player_Hand.clear()
    dealer_Hand.clear()
    add_Cards(dealer_Hand)
    add_Cards(dealer_Hand)
    add_Cards(player_Hand)
    add_Cards(player_Hand)
```

Python Code: Server (cont.)

```
    # While loop used to play the game, fucntions while the game is being played, playing (boolean) = True
    while (playing==True):
        # Receives the input from the player on whether they want to hit or stand
        data = connection.recv(1024).decode('utf-8')
        # If statement that checks to see if the player choose to hit
        if (data == "hit"):
            hit()
            # If statement that checks if the game has ended; sets playing (boolean) to False
            if (checkStatusHit()!=""):
                playing=False
            # Sends message of the current game
            connection.send(str.encode(oldGame()+"\n"+checkStatusHit()))
        # If statement that checks to see if the player choose to stand
        elif (data == "stand"):
            stand()
            # If statement that checks if the game has ended; sets playing (boolean) to False
            if (checkStatusStand()!=""):
                playing=False
            # Sends message of the current game
            connection.send(str.encode(oldGame()+"\n"+checkStatusStand()))
        # If statement that checks if the game has ended, playing (boolean) = False; resets the game, and sets playing (boolean) to True
        if(playing==False):
            starting_Hand()
            connection.send(str.encode("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n
            playing=True
    connection.close()

# Main while loop
while True:
    Client, address = ServerSocket.accept() #accept new client
    print('Connected to: ' + address[0] + ':' + str(address[1])) #print client address
    start_new_thread(threaded_client, (Client, ThreadCount, )) #create new thread with client connection and id
    ThreadCount += 1
    print('Thread Number: ' + str(ThreadCount))
ServerSocket.close()
```

Python Code: Client

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Mar 15 19:18:17 2021

@author: Ghasif, Yehua, Tong
"""

# Imports
import socket

# Establish connection with server
ClientSocket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
port = 1233
ClientSocket.connect(('127.0.0.1', port))

# Receive initial response from server
response = ClientSocket.recv(1024)
print(response.decode('utf-8'))

# While loop that maintains communication between the client and server back and forth
while True:
    userInput = input('Give your input: ')
    ClientSocket.send(str.encode(userInput))
    response = ClientSocket.recv(1024)
    print(response.decode('utf-8'))
ClientSocket.close()
```

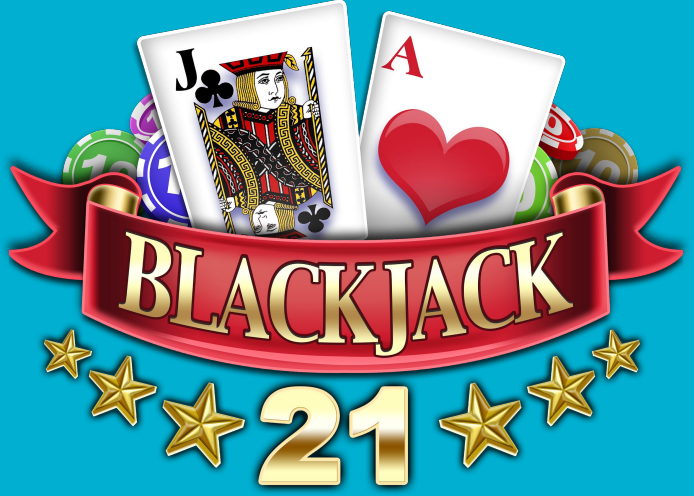
DEMO

Conclusion and Future Work

- Creating blackjack in python taught us a lot, not just the coding and socket programming part, but the whole process of it. We got to learn and experience ourselves from this project.
- If we have more time on this project, we will do the betting coins.
- Another thing we want to do if we have more time is the implementation of GUI (Graphical User Interface) for this project. It takes too much time, but it is very fun.

References

- https://bicyclecards.com/how-to-play/blackjack/
- https://www.askpython.com/python/examples/blackjack-game-using-python
- https://www.youtube.com/watch?v=IPMcV_IXtX4
- https://games.washingtonpost.com/games/blackjack
- https://www.geeksforgeeks.org/socket-programming-multi-threading-python/
- https://docs.python.org/3/library/threading.html



Thank You
Any questions?