

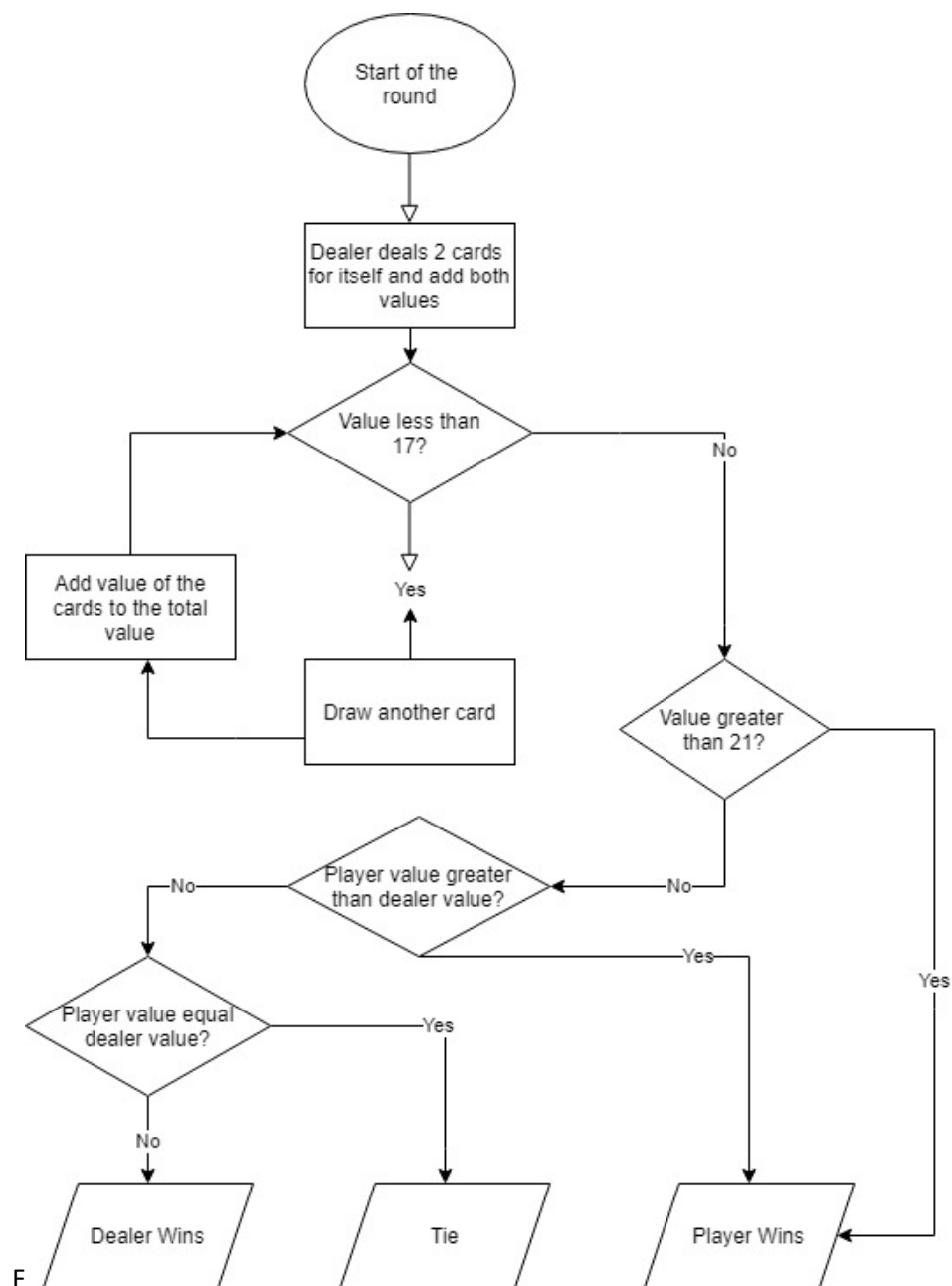
Mini Project – BlackJack

Parth Lohia(C108)

Siddh Gandhi(C122)

Aryan Mashru(C123)

Flowchart/ Algorithm



Code

```
from tkinter import *

root = Tk()

root.title("Blackjack")

root.geometry("1280x720")

root.resizable(width=False,height=False)

root["bg"] = "red4"

score_dealer = 0

score_player = 0
```

```
import tkcalendar

from tkinter import messagebox

from datetime import date

import random

import re
```

```
def switch_frames_gametologin():

    Frame3.place_forget()

    Frame2.place_forget()

    Frame5.place_forget()

    dealer_card_frame.place_forget()

    player_card_frame.place_forget()
```

```
def switch_frames_logintogame():

    login_frame.place_forget()

    registration_frame.place_forget()

    dealer_card_frame.place(x=0, y=0)

    Frame2.place(x=0, y=240)

    Frame3.place(x=320, y=240)

    Frame5.place(x= 960, y=240)

    player_card_frame.place(x=0, y=480)
```

```
regex = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
```

```
def register_user():
```

```
    email_info = emailID_ET_r.get()
```

```
    username_info = username_ET_r.get()
```

```
    password_info = password_ET_r.get()
```

```
    password_info_2 = confirm_password_ET_r.get()
```

```
    userday = int(spinbox_dob_1.get())
```

```
    usermonth = int(spinbox_mob_1.get())
```

```
    useryear = int(spinbox_yob_1.get())
```

```
    today = date.today()
```

```
    date1 = date(useryear, usermonth, userday)
```

```
    if email_info == "" or username_info == "" or password_info == "":
```

```
        messagebox.showerror("", "One or more fields have been left empty")
```

```
    elif len(username_info) <= 3 :
```

```
        messagebox.showerror("", "Username must have more than 3 characters")
```

```
    elif len(password_info) <= 5 :
```

```
        messagebox.showerror("", "Password must have more than 5 characters")
```

```
    elif int((today - date1).days) < 5840:
```

```
        messagebox.showerror("", "You must be at least 16 years of age to continue")
```

```
    elif password_info == username_info:
```

```
        messagebox.showerror("", "Username and Password cannot be same")
```

else:

```
if password_info == password_info_2 and (re.fullmatch(regex, email_info)):
```

```
    # Open file in write mode
```

```
    file = open("myfile.rtf", "a")
```

```
    # write username and password information into file
```

```
    file.write("\n")
```

```
    file.write(username_info)
```

```
    file.write(",")
```

```
    file.write(password_info)
```

```
    file.write("\n")
```

```
    file.close()
```

```
    emailID_ET_r.delete(0, END)
```

```
    username_ET_r.delete(0, END)
```

```
    password_ET_r.delete(0, END)
```

```
    confirm_password_ET_r.delete(0, END)
```

```
    registration_frame.place_forget()
```

```
    login_frame.place(relx=.5, rely=.5, anchor=CENTER)
```

```
elif password_info != password_info_2:
```

```
    messagebox.showerror("", "PASSWORDS DO NOT MATCH!")
```

```
    password_ET_r.delete(0, END)
```

```
    confirm_password_ET_r.delete(0, END)
```

```
else:
```

```
    messagebox.showerror("", "Enter Valid E-mail ID")
```

```
def show_register():  
    login_frame.place_forget()  
    registration_frame.place(relx=.5, rely=.5, anchor=CENTER)
```

```
def register():  
    register_user()
```

```
# date1 = cal.get_date()  
# d = date.today()  
# user_day = d.day  
# user_month = d.month  
# user_year = d.year  
# date_format = '%m-%d-%y'  
# date_string = user_month + '/' + user_day + '/' + user_year  
# d.strptime(date_string, date_format)  
# print(d)  
# new_date1 = date(user_year, user_month, user_day)
```

```
def Login():  
    username1 = username_ET.get()  
    password1 = ("," + password_ET.get())  
  
    if username1 == "" or password_ET.get() == "":  
        messagebox.showerror("", "Enter Username and Password")  
    else:  
  
        file = open("myfile.rtf", "r")
```

```
for row in file:
```

```
    if username1 in row and password1 in row:
```

```
        status_login = 1
```

```
        switch_frames_logintogame()
```

```
        break
```

```
    else:
```

```
        status_login = 0
```

```
if status_login == 0:
```

```
    messagebox.showerror("", "User not found, Register")
```

```
login_frame = Frame(root, width=128, height=720, bg="red4", highlightthickness=1, highlightcolor="#000000")
```

```
login_frame.place(relx=.5, rely=.5, anchor=CENTER)
```

```
heading_label = Label(login_frame, text="Login to BlackJack ♠", bg="red4", font=("Konstler", 50), pady=20)
```

```
heading_label.grid(row=0, column=0, columnspan=2)
```

```
username_label = Label(login_frame, text="Enter Username: ", bg="red4", padx=20, fg="gold")
```

```
username_label.grid(row=1, column=0)
```

```
username_ET = Entry(login_frame)
```

```
username_ET.grid(row=1, column=1, padx=20)
```

```
password_label = Label(login_frame, bg="red4", text="Enter Password:", padx=20, fg="gold")
```

```
password_label.grid(row=2, column=0)
```

```
password_ET = Entry(login_frame, show="*")
```

```
password_ET.grid(row=2, column=1, pady=30, padx=20)
```

```
register_button = Button(login_frame, text="Register", command=show_register, bg="goldenrod")
```

```
register_button.grid(row=3, column=0, padx=20, pady=20)
```

```
login_button = Button(login_frame, text="Login", command=Login, bg="goldenrod")
```

```
login_button.grid(row=3, column=1, padx=20, pady=20)
```

```
registration_frame = Frame(root, width=128, height=720, bg="red4", highlightthickness=1,
highlightcolor="#000000")

registration_frame.place_forget()

heading_label_r = Label(registration_frame, text="Register to BlackJack ♠", bg="red4", font=("Al Bayan", 30),
pady=20)

heading_label_r.grid(row=0, column=0, columnspan=2)


emailID_label_r = Label(registration_frame, text="Enter Email ID: ", bg="red4", padx=20, fg="gold")
emailID_label_r.grid(row=1, column=0)

emailID_ET_r = Entry(registration_frame)
emailID_ET_r.grid(row=1, column=1, padx=20, pady=20)


username_label_r = Label(registration_frame, text="Enter Username: ", bg="red4", padx=20, fg="gold")
username_label_r.grid(row=2, column=0)

username_ET_r = Entry(registration_frame)
username_ET_r.grid(row=2, column=1, padx=20, pady=20)


password_label_r = Label(registration_frame, bg="red4", text="Enter Password:", padx=20, fg="gold")
password_label_r.grid(row=3, column=0)

password_ET_r = Entry(registration_frame)
password_ET_r.grid(row=3, column=1, padx=20, pady=20)


confirm_password_label_r = Label(registration_frame, bg="red4", text="Confirm Password:", padx=20,
fg="gold")
confirm_password_label_r.grid(row=4, column=0)

confirm_password_ET_r = Entry(registration_frame)
confirm_password_ET_r.grid(row=4, column=1, padx=20, pady=20)


birthdate_label_r = Label(registration_frame, bg="red4", text="Day:", padx=20, fg="gold")
birthdate_label_r.grid(row=5, column=0)

dob_1 = IntVar()

spinbox_dob_1 = Spinbox(registration_frame, from_=1, to=30, width=5)
spinbox_dob_1.grid(row=5, column=1)
```

```

birthmonth_label_r = Label(registration_frame, bg="red4", text="Month:", padx=20, fg="gold")
birthmonth_label_r.grid(row=6, column=0)

mob_1 = IntVar()
spinbox_mob_1 = Spinbox(registration_frame, from_=1, to=12, width=5)
spinbox_mob_1.grid(row=6, column=1)


birthyear_label_r = Label(registration_frame, bg="red4", text="Year:", padx=20, fg="gold")
birthyear_label_r.grid(row=7, column=0)

yob_1 = IntVar()
spinbox_yob_1 = Spinbox(registration_frame, from_=1920, to=2021, width=5)
spinbox_yob_1.grid(row=7, column=1)


confirm_registration_button = Button(registration_frame, text="Confirm Registration", command=register)
confirm_registration_button.grid(row=8, column=0, columnspan=2, padx=20, pady=20)


def load_images(card_images):
    suits = ['heart', 'club', 'diamond', 'spade']
    face_cards = ['jack', 'queen', 'king']
    extension = 'png'

    # for each suit, retrieve the image for the cards
    for suit in suits:
        # first the number cards 1 to 10
        for card in range(1, 11):
            name = 'cards/{}/{_}.{}'.format(str(card), suit, extension)
            image = PhotoImage(file=name)
            card_images.append((card, image, ))

        # next the face cards

```



```

    for card in face_cards:

        name = 'cards/{}/{_}.{}'.format(str(card), suit, extension)

        image = PhotoImage(file=name)

        card_images.append((10, image, ))

def _deal_card(frame):

    # pop the next card off the top of the deck

    next_card = deck.pop(0) # Use 0 to take card from top of deck

    # and add it back to the deck

    deck.append(next_card)

    # add the image to a label and display the label

    Label(frame, image=next_card[1], relief="raised").pack(side="left")

    # now return the card's face value

    return next_card

def score_hand(hand):

    # Calculate the total score of all cards in the list

    # Only one ace can have the value 11 and this will be reduce to 1 if the hand would bust

    score = 0

    ace = False

    for next_card in hand:

        card_value = next_card[0]

        if card_value == 1 and not ace:

            ace = True

            card_value = 11

        score += card_value

        # if we would bust, check if there is an ace and subtract 10

    if score > 21 and ace:

        score -= 10

        ace = False

    return score

def deal_dealer():

    dealer_score = score_hand(dealer_hand)

```

```
while 0 < dealer_score < 17:  
    dealer_hand.append(_deal_card(dealer_card_frame))  
    dealer_score = score_hand(dealer_hand)  
    score_dealer = score_hand(dealer_hand)  
    var_dealer_score.set(f"Dealer Score: {dealer_score}")
```

```
player_score = score_hand(player_hand)  
if player_score > 21:  
    result_text.set("Dealer wins!")  
    hit_button["state"] = "disabled"  
    stand_button["state"] = "disabled"
```

```
elif dealer_score > 21 or dealer_score < player_score:  
    result_text.set("Player wins!")  
    hit_button["state"] = "disabled"  
    stand_button["state"] = "disabled"
```

```
elif dealer_score > player_score:  
    result_text.set("Dealer wins!")  
    hit_button["state"] = "disabled"  
    stand_button["state"] = "disabled"
```

```
else:  
    result_text.set("Draw!")  
    hit_button["state"] = "disabled"  
    stand_button["state"] = "disabled"
```

```
def deal_player():  
    player_hand.append(_deal_card(player_card_frame))  
    player_score = score_hand(player_hand)  
    score_player = score_hand(player_hand)  
    var_player_score.set(f"Player Score: {score_player}")
```

```

if player_score > 21:
    result_text.set("Dealer Wins!")
    hit_button["state"] = "disabled"
    stand_button["state"] = "disabled"
if player_score == 21:
    result_text.set("Player wins!")
    hit_button["state"] = "disabled"
    stand_button["state"] = "disabled"

def initial_deal():
    deal_player()
    dealer_hand.append(_deal_card(dealer_card_frame))
    score_dealer = score_hand(dealer_hand)
    deal_player()
    dealer_score = score_hand(dealer_hand)
    var_dealer_score.set(f"Dealer Score: {dealer_score}")

def new_game():
    global dealer_card_frame
    global player_card_frame
    global dealer_hand
    global player_hand

    # embedded frame to hold the card images
    dealer_card_frame.destroy()

    dealer_card_frame = Frame(root, width = 1280, height = 240, bg = "red4", highlightthickness = 1,
highlightcolor = "#000000", padx = 5, pady = 5)
    dealer_card_frame.pack_propagate(0)
    dealer_card_frame.place(x=0, y=0)

    # embedded frame to hold the card images
    player_card_frame.destroy()

    player_card_frame = Frame(root, width = 1280, height = 240, bg = "red4", highlightthickness = 1,
highlightcolor = "#000000", padx = 5, pady = 5)

```

```

player_card_frame.pack_propagate(0)

player_card_frame.place(x=0, y=480)


result_text.set("")

hit_button["state"]= "normal"

stand_button["state"]="normal"


# Create the list to store the dealer's and player's hands

dealer_hand = []

player_hand = []

initial_deal()


def shuffle():

    random.shuffle(deck)


def play():

    initial_deal()


#REGISTRATIONFRAME

#registration_frame = Frame(root, width = 1280 , height = 720, bg ="blue",highlightthickness = 1, highlightcolor
="#000000")

#registration_frame.pack_propagate(0)

#registration_frame.place(x=0,y=0)

#DEALERFRAME ***

dealer_card_frame = Frame(root, width = 1280, height = 240, bg ="red4", highlightthickness = 1, highlightcolor
="#000000", padx = 5, pady = 5)

dealer_card_frame.pack_propagate(0)

dealer_card_frame.place(x=0, y=0)

#ACTIONSFRAME ***

Frame2 = Frame(root, width = 320, height = 240, bg = "red4",highlightthickness = 1, highlightcolor =
"#000000")

```

```

Frame2.pack_propagate(0)

Frame2.place(x=0,y=240)

#BUTTONS   **

hit_button = Button(Frame2,text = "HIT",width = 30,fg = "red4",bg = "gold", command = deal_player)
hit_button.pack(side = TOP, pady = 10)


stand_button = Button(Frame2,text = "STAND",width = 30,fg = "red4",bg = "gold", command = deal_dealer)
stand_button.pack(side = TOP, pady = 10)


new_game_button = Button(Frame2,text = "NEW GAME",width = 30,fg = "red4",bg = "gold", command =
new_game)

new_game_button.pack(side = TOP, pady = 10)


#RESULTFRAME   ***

Frame3 = Frame(root, width = 960, height = 240, bg = "red4",highlightthickness = 1, highlightcolor =
"#000000")

Frame3.pack_propagate(0)

Frame3.place(x=320,y=240)

result_text = StringVar()

result = Label(Frame3, textvariable=result_text,bg="red4", font = ("Arial", 50),fg = "white")

result.pack(side = LEFT,padx = 20)


#SCOREFRAME   ***

Frame5 = Frame(root, width = 320, height = 240, bg = "red4",highlightthickness = 1, highlightcolor =
"#000000")

Frame5.pack_propagate(0)

Frame5.place(x = 960 , y = 240)

var_player_score = IntVar()

var_player_score.set(f"Player Score: {score_player}")

player_score_label = Label(Frame5, textvariable = var_player_score, fg = "white", bg = "red4")

player_score_label.pack(side = BOTTOM, fill = BOTH)

var_dealer_score = IntVar()

var_dealer_score.set(f"Dealer Score: {score_dealer}")

```

```
dealer_score_label = Label(Frame5, textvariable = var_dealer_score, fg = "white", bg = "red4")
```

```
dealer_score_label.pack(side = TOP, fill = BOTH)
```

```
#PLAYERFRAME ***
```

```
player_card_frame = Frame(root, width = 1280, height = 240, bg = "red4", highlightthickness = 1, highlightcolor = "#000000", padx = 5, pady = 5)
```

```
player_card_frame.pack_propagate(0)
```

```
player_card_frame.place(x=0, y=480)
```

```
# load cards
```

```
cards = []
```

```
load_images(cards)
```

```
print(cards)
```

```
# Create a new deck of cards and shuffle them
```

```
deck = list(cards) + list(cards) + list(cards)
```

```
shuffle()
```

```
dealer_hand = [] # to store dealers hand
```

```
player_hand = [] # to store players hand
```

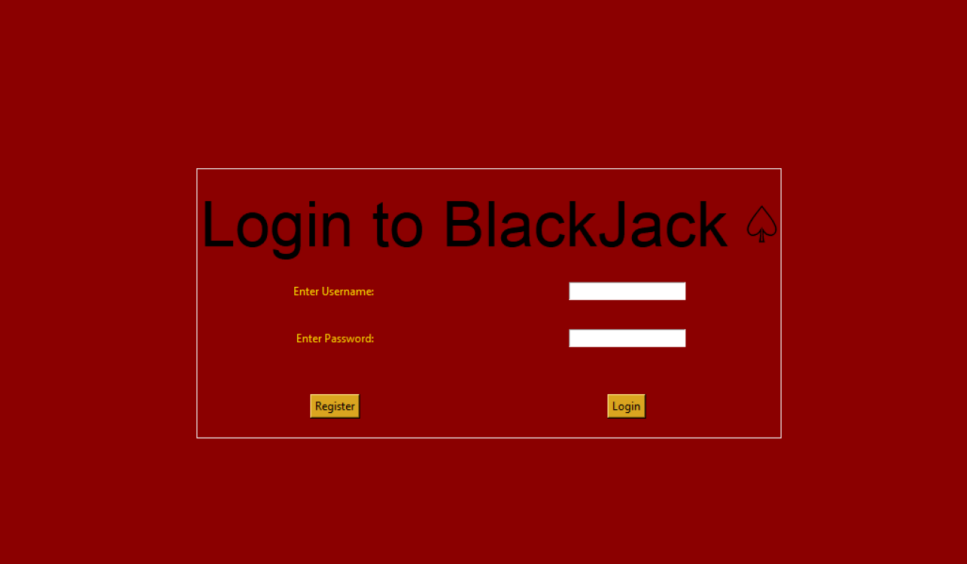
```
play()
```

```
switch_frames_gametologin()
```

```
root.mainloop()
```

Output

Start Menu



BlackJack

Login to BlackJack ♠

Enter Username:

Enter Password:

[Register](#) [Login](#)

Registration Menu

Blackjack

Register to BlackJack

Enter Email ID:

sid@gmail.com

Enter Username:

sidgandhi

Enter Password:

sid2004

Confirm Password:

sid2004

Day:

17

Month:

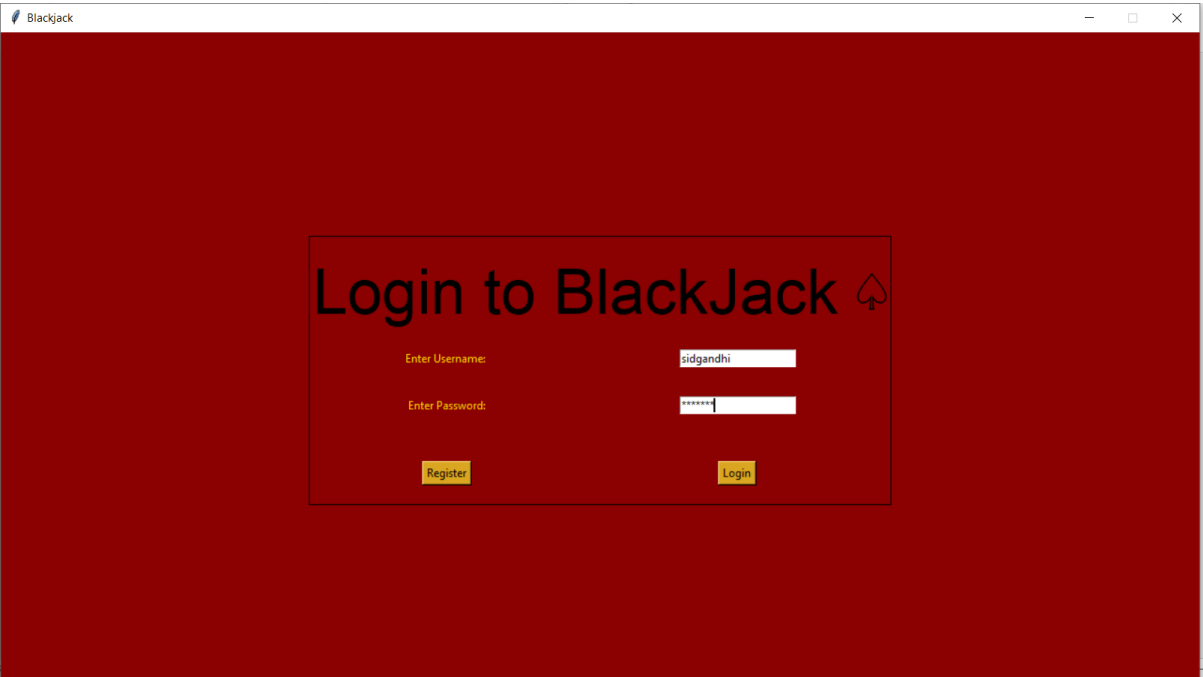
12

Year:

2004

Confirm Registration

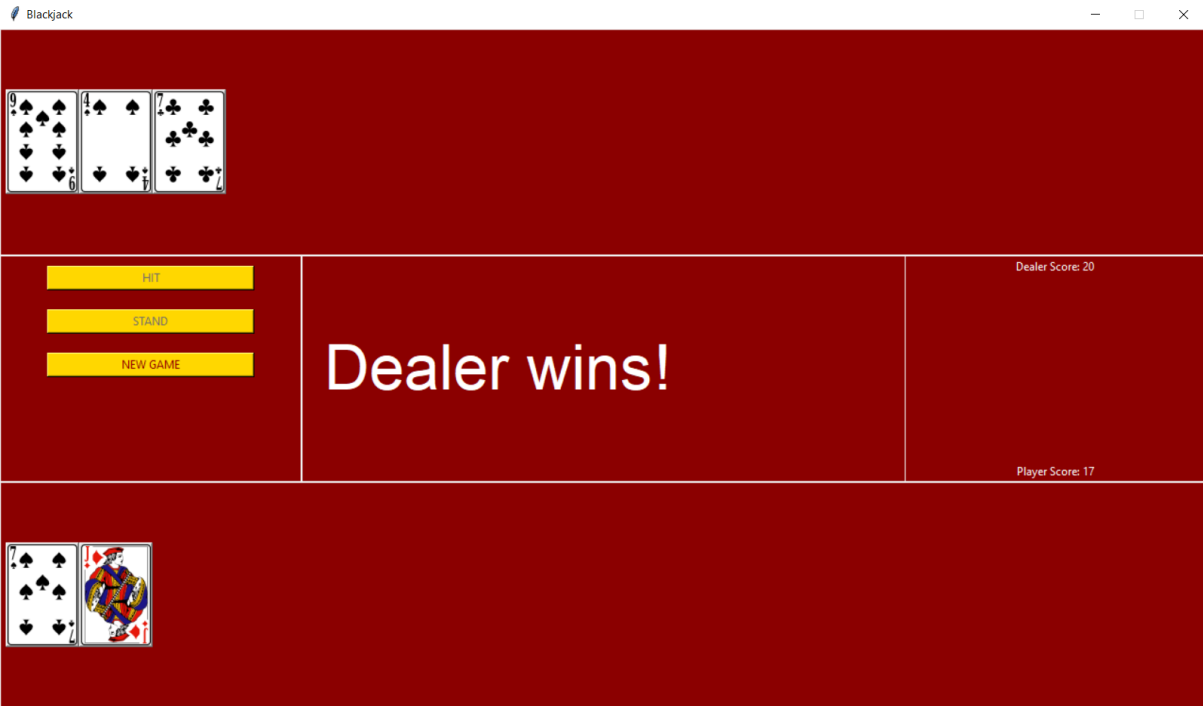
Login Menu



The Game Area



Scenario – Dealer wins



Scenario- Player Wins

