'Boyle Family Casino'

Virtual Gaming Application

(Python Program)

By

Nate Boyle

12/24/2023

MAIN MENU

Users can:

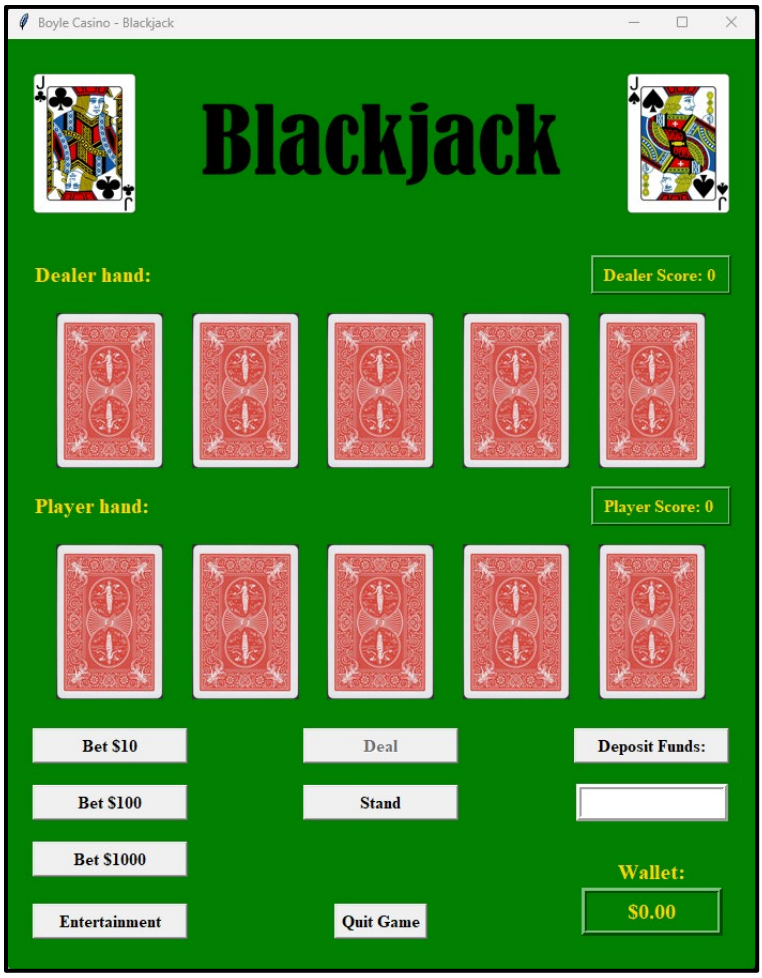
- Select to play Blackjack, Video Poker, or SLOT WARS
- They can exit the app via the Exit Casino button or just closing out the window
    - Note: closing the main menu closes the app entirely. It must remain open (but can me minimized) while playing games.
- Pause or play the casino lounge music that initiates when the app is opened.

Note: The playing cards above are a custom design based on the Boyle family crest.

BLACKJACK

Game Details:

- Standard blackjack rules.
- Players can deposit unlimited imaginary funds into the wallet.
- Bets can be placed until the wallet runs out of funds.
- In the bottom left corner is an 'Entertainment' button that opens YouTube to the *Roast of Frank Sinatra*.
- Pop-up messages will display when the player loses, wins, draws, or runs out of funds.



Selected Blackjack Screenshots

Player draws 21 and beats the dealer:

Dealer's hand beats the player:

Player's hand is a bust:

Player draws Blackjack!

VIDEO POKER

Game Details:

- Standard jacks-or-better poker rules.
- Players can deposit unlimited imaginary funds into the wallet.
- Bets can be placed until the wallet runs out of funds.
- When the game initiates the *Best of Frank Sinatra* audio track begins to play. Below the bottom right corner of the payout table corner is a button that allows players to pause/play the track.
- Pop-up messages will display when the player loses, wins, draws, or runs out of funds.

Selected Video Poker Screenshots

Jacks or better:

Losing hand:

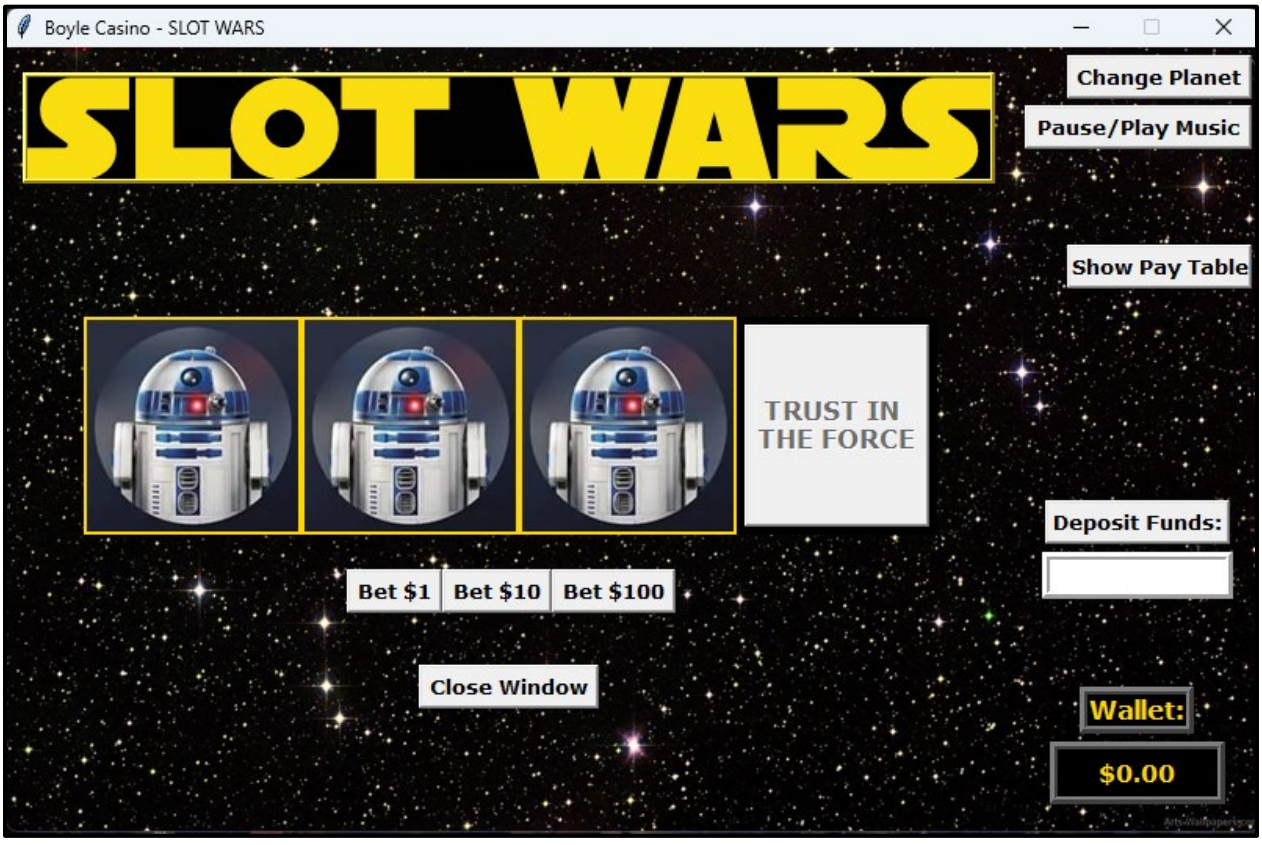
Two pair:

Flush!

SLOT WARS

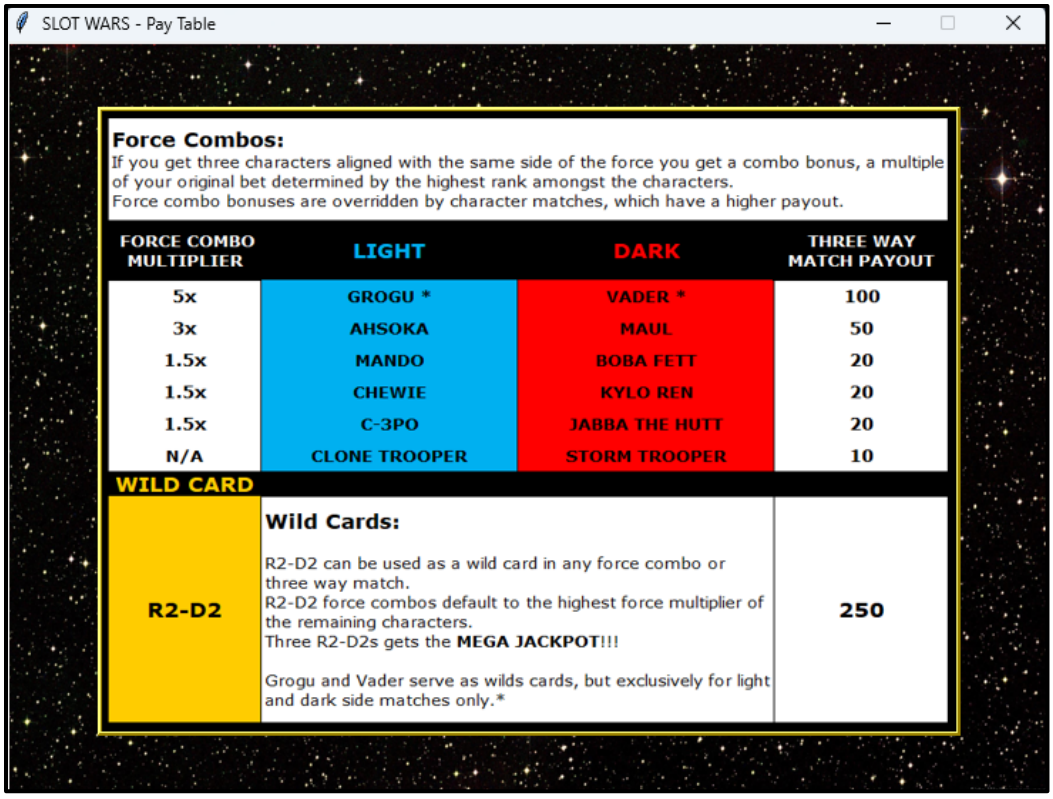
Game Details:

- Standard 3-reel slot machine game.
- Players can deposit unlimited imaginary funds into the wallet.
- Bets can be placed until the wallet runs out of funds.
- When the game initiates a *Star Wars* audio track begins to play based on what planet (background) a player is on. In the top right corner of the window are buttons that allow players to change their planet or pause/play the track.
- Below the buttons mentioned above is a 'Show Pay Table' button that shows all of the possible winning combos and their payouts.
- Pop-up messages will display when the player loses, wins, or runs out of funds.

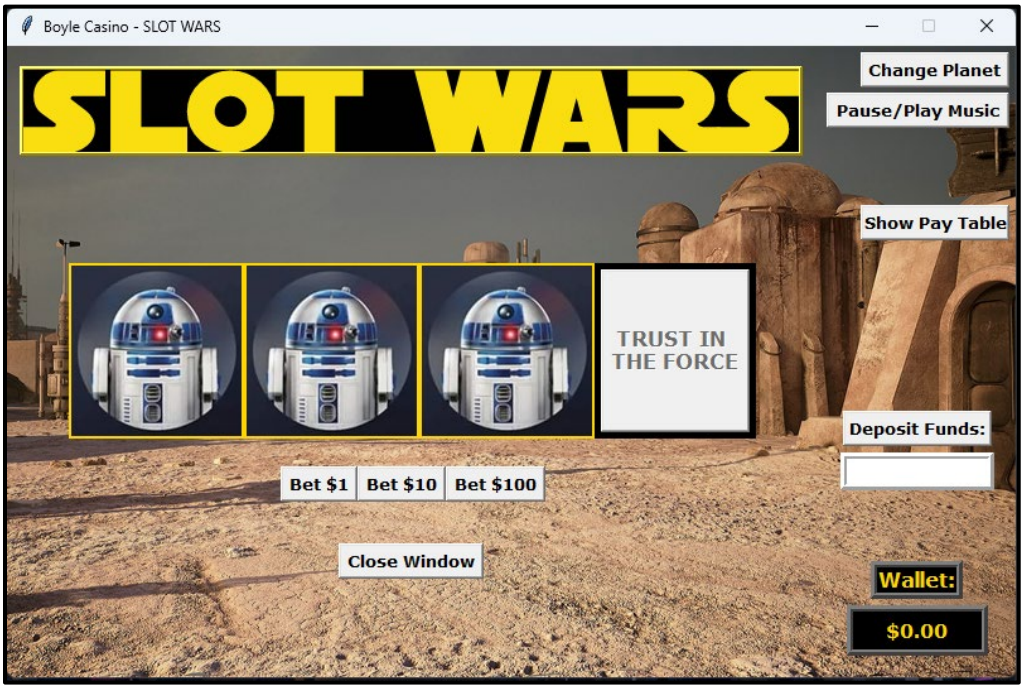


Selected SLOT WARS Screenshots

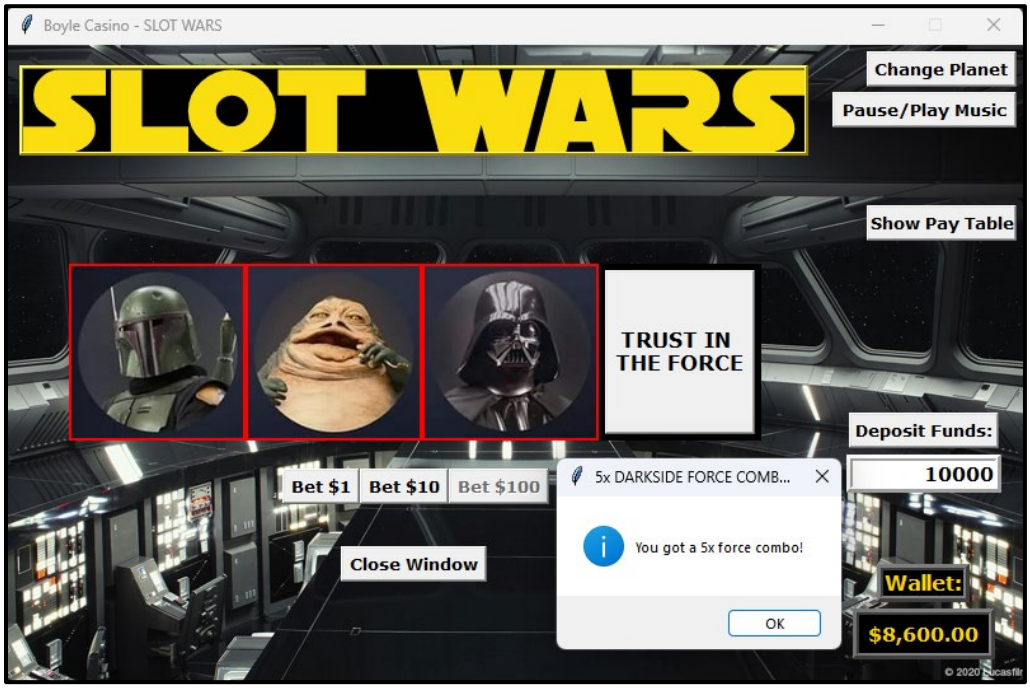
Payout Table:



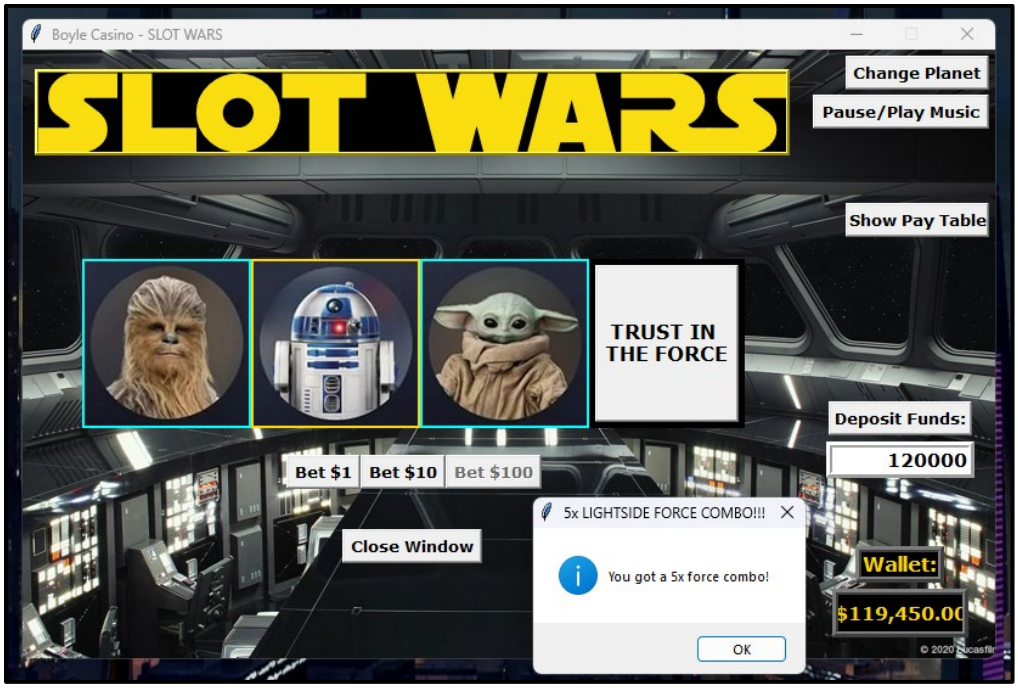
A game being played on Tatooine:



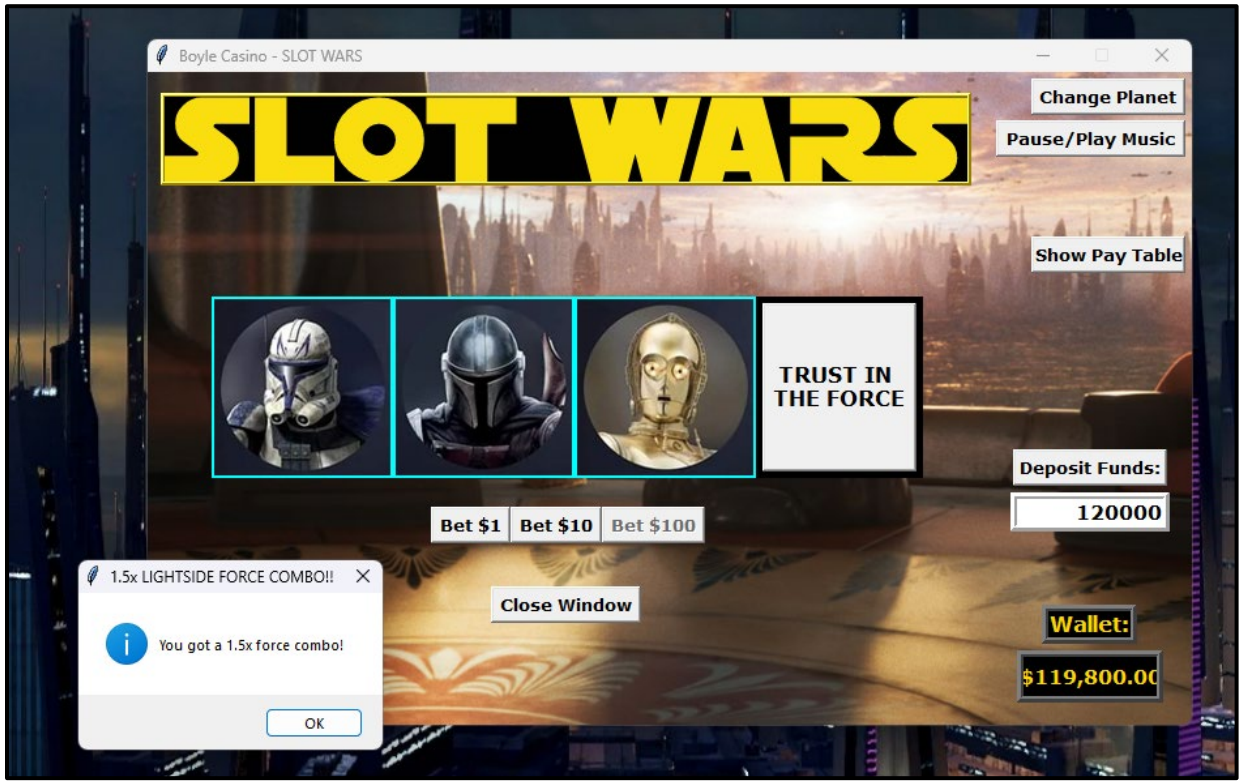
A 5x Darkside force combo on the bridge of an Imperial Star Destroyer:



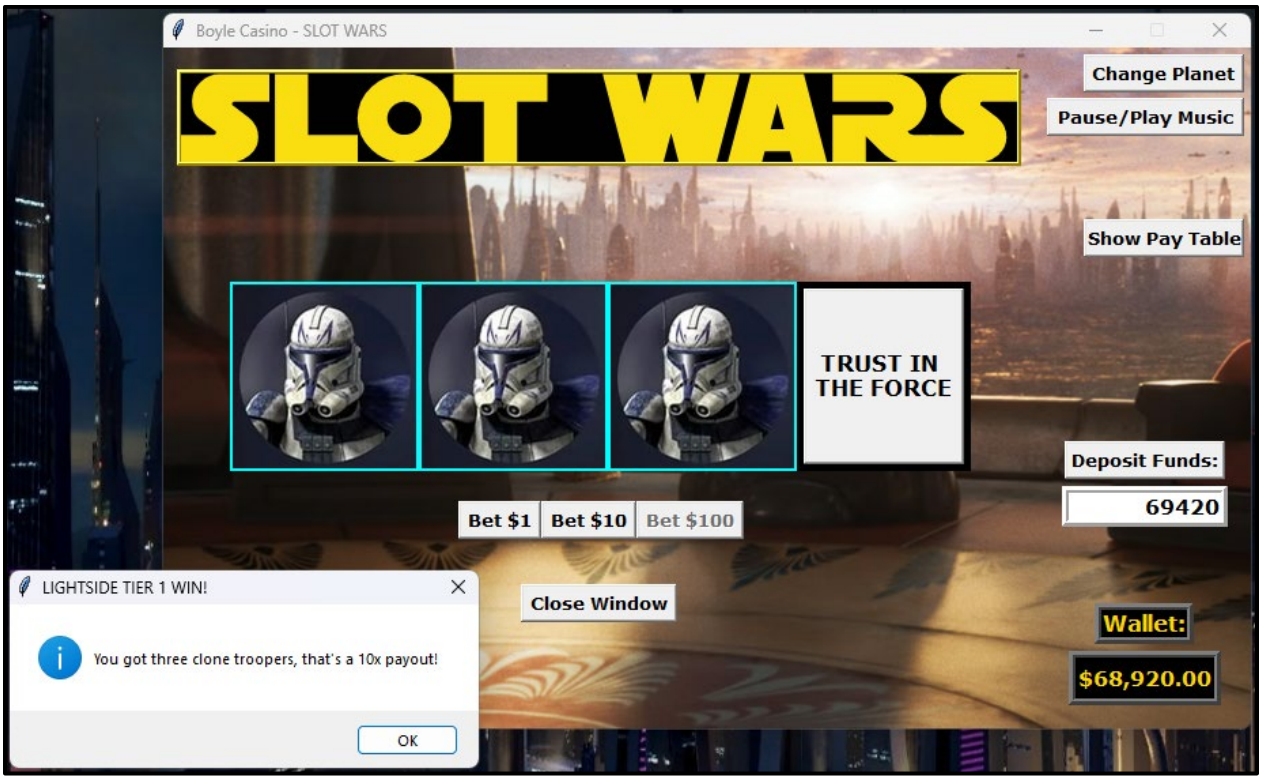
A 5x Lightside force combo on the bridge of an Imperial Star Destroyer:



A 1.5x Lightside force combo inside the masters' chamber of the Jedi Temple:



Three of a kind!



Please find the program code on the remaining pages.

```
#!/usr/bin/env python
# coding: utf-8

# ## Table of contents
#
# ###### ctrl+f a line of text below to navigate to the associated area
#
# #### 1* create card image variables
# #### 2* create card dictionaries
# #### 3* create deck dictionary
# #### 4* load slot reel images
# #### 5* create slot reel symbol dictionaries
# #### 6* initialize slot reel symbol list and add dictionaries
# #### 7* blackjack gui
# #### 8* poker gui
# #### 9* slots gui
# #### 10* welcome menu gui
# #### 11* open program

# In[13]:


# import os (operating system) library
import os

# import sys (system) library
import sys

# import pandas library as pd
import pandas as pd

# import numpy library as np
import numpy as np

# import these modules from datetime to store and compare dates
from datetime import datetime, date, timedelta

# import time for delay
import time
import pygame

# from tkinter import all standard modules with * (this is for the gui)
from tkinter import *
# from tkinter 'specifically' import messagebox, ttk as they are not standard
modules uploaded with *
from tkinter import messagebox, ttk

# import imageTk and Image for picture use in gui
from PIL import ImageTk as itk, Image
```

```
# import various matplotlib modules to create plots and then draw those plots in
the gui
import matplotlib.pyplot as plt
from matplotlib.figure import Figure
import matplotlib.ticker as mtick
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg,
NavigationToolbar2Tk)

import random
import ipyplot

# import webbrowser for hyperlink use
import webbrowser

# import IPython display for wider coding screen (not required to run program)
from IPython.display import display, HTML
display(HTML("<style>.jp-Cell { width: 120% !important; }</style>"))


# ### 1* create card image variables

# In[14]:


#create image variable for card reverse
card_reverse_img = Image.open("CardPics/redcardback.png").resize((100, 145))

# create image variables for clubs
two_of_clubs_img = Image.open("CardPics/2_of_clubs.png").resize((100, 145))
three_of_clubs_img = Image.open("CardPics/3_of_clubs.png").resize((100, 145))
four_of_clubs_img = Image.open("CardPics/4_of_clubs.png").resize((100, 145))
five_of_clubs_img = Image.open("CardPics/5_of_clubs.png").resize((100, 145))
six_of_clubs_img = Image.open("CardPics/6_of_clubs.png").resize((100, 145))
seven_of_clubs_img = Image.open("CardPics/7_of_clubs.png").resize((100, 145))
eight_of_clubs_img = Image.open("CardPics/8_of_clubs.png").resize((100, 145))
nine_of_clubs_img = Image.open("CardPics/9_of_clubs.png").resize((100, 145))
ten_of_clubs_img = Image.open("CardPics/10_of_clubs.png").resize((100, 145))
jack_of_clubs_img = Image.open("CardPics/jack_of_clubs.png").resize((100, 145))
queen_of_clubs_img = Image.open("CardPics/queen_of_clubs.png").resize((100, 145))
king_of_clubs_img = Image.open("CardPics/king_of_clubs.png").resize((100, 145))
ace_of_clubs_img = Image.open("CardPics/ace_of_clubs.png").resize((100, 145))

# create image variables for diamonds
two_of_diamonds_img = Image.open("CardPics/2_of_diamonds.png").resize((100, 145))
three_of_diamonds_img = Image.open("CardPics/3_of_diamonds.png").resize((100, 145))
four_of_diamonds_img = Image.open("CardPics/4_of_diamonds.png").resize((100, 145))
five_of_diamonds_img = Image.open("CardPics/5_of_diamonds.png").resize((100, 145))
six_of_diamonds_img = Image.open("CardPics/6_of_diamonds.png").resize((100, 145))
seven_of_diamonds_img = Image.open("CardPics/7_of_diamonds.png").resize((100, 145))
eight_of_diamonds_img = Image.open("CardPics/8_of_diamonds.png").resize((100, 145))
```

```
nine_of_diamonds_img = Image.open("CardPics/9_of_diamonds.png").resize((100, 145))
ten_of_diamonds_img = Image.open("CardPics/10_of_diamonds.png").resize((100, 145))
jack_of_diamonds_img = Image.open("CardPics/jack_of_diamonds.png").resize((100,
145))
queen_of_diamonds_img = Image.open("CardPics/queen_of_diamonds.png").resize((100,
145))
king_of_diamonds_img = Image.open("CardPics/king_of_diamonds.png").resize((100,
145))
ace_of_diamonds_img = Image.open("CardPics/ace_of_diamonds.png").resize((100, 145))

# create image variables for hearts
two_of_hearts_img = Image.open("CardPics/2_of_hearts.png").resize((100, 145))
three_of_hearts_img = Image.open("CardPics/3_of_hearts.png").resize((100, 145))
four_of_hearts_img = Image.open("CardPics/4_of_hearts.png").resize((100, 145))
five_of_hearts_img = Image.open("CardPics/5_of_hearts.png").resize((100, 145))
six_of_hearts_img = Image.open("CardPics/6_of_hearts.png").resize((100, 145))
seven_of_hearts_img = Image.open("CardPics/7_of_hearts.png").resize((100, 145))
eight_of_hearts_img = Image.open("CardPics/8_of_hearts.png").resize((100, 145))
nine_of_hearts_img = Image.open("CardPics/9_of_hearts.png").resize((100, 145))
ten_of_hearts_img = Image.open("CardPics/10_of_hearts.png").resize((100, 145))
jack_of_hearts_img = Image.open("CardPics/jack_of_hearts.png").resize((100, 145))
queen_of_hearts_img = Image.open("CardPics/queen_of_hearts.png").resize((100, 145))
king_of_hearts_img = Image.open("CardPics/king_of_hearts.png").resize((100, 145))
ace_of_hearts_img = Image.open("CardPics/ace_of_hearts.png").resize((100, 145))

# create image variables for spades
two_of_spades_img = Image.open("CardPics/2_of_spades.png").resize((100, 145))
three_of_spades_img = Image.open("CardPics/3_of_spades.png").resize((100, 145))
four_of_spades_img = Image.open("CardPics/4_of_spades.png").resize((100, 145))
five_of_spades_img = Image.open("CardPics/5_of_spades.png").resize((100, 145))
six_of_spades_img = Image.open("CardPics/6_of_spades.png").resize((100, 145))
seven_of_spades_img = Image.open("CardPics/7_of_spades.png").resize((100, 145))
eight_of_spades_img = Image.open("CardPics/8_of_spades.png").resize((100, 145))
nine_of_spades_img = Image.open("CardPics/9_of_spades.png").resize((100, 145))
ten_of_spades_img = Image.open("CardPics/10_of_spades.png").resize((100, 145))
jack_of_spades_img = Image.open("CardPics/jack_of_spades.png").resize((100, 145))
queen_of_spades_img = Image.open("CardPics/queen_of_spades.png").resize((100, 145))
king_of_spades_img = Image.open("CardPics/king_of_spades.png").resize((100, 145))
ace_of_spades_img = Image.open("CardPics/ace_of_spades.png").resize((100, 145))


# ### 2* create card dictionaries

# In[15]:


two_of_clubs = {'suit': 'clubs', 'points':  2,  'image': two_of_clubs_img, 'face' :
'two', 'rank': 2}
three_of_clubs = {'suit': 'clubs', 'points':  3,  'image': three_of_clubs_img,
'face' : 'three', 'rank': 3}
```

```
four_of_clubs = {'suit': 'clubs', 'points':  4,  'image': four_of_clubs_img, 'face'
: 'four', 'rank': 4}
five_of_clubs = {'suit': 'clubs', 'points':  5,  'image': five_of_clubs_img, 'face'
: 'five', 'rank': 5}
six_of_clubs = {'suit': 'clubs', 'points':  6,  'image': six_of_clubs_img, 'face' :
'six', 'rank': 6}
seven_of_clubs = {'suit': 'clubs', 'points':  7,  'image': seven_of_clubs_img,
'face' : 'seven', 'rank': 7}
eight_of_clubs = {'suit': 'clubs', 'points':  8,  'image': eight_of_clubs_img,
'face' : 'eight', 'rank': 8}
nine_of_clubs = {'suit': 'clubs', 'points':  9,  'image': nine_of_clubs_img, 'face'
: 'nine', 'rank': 9}
ten_of_clubs = {'suit': 'clubs', 'points':  10,  'image': ten_of_clubs_img, 'face'
: 'ten', 'rank': 10}
jack_of_clubs = {'suit': 'clubs', 'points':  10,  'image': jack_of_clubs_img,
'face' : 'jack', 'rank': 11}
queen_of_clubs = {'suit': 'clubs', 'points':  10,  'image': queen_of_clubs_img,
'face' : 'queen', 'rank': 12}
king_of_clubs = {'suit': 'clubs', 'points':  10,  'image': king_of_clubs_img,
'face' : 'king', 'rank': 13}
ace_of_clubs = {'suit': 'clubs', 'points':  11,  'image': ace_of_clubs_img, 'face'
: 'ace', 'rank': 14}


two_of_diamonds = {'suit': 'diamonds', 'points':  2,  'image': two_of_diamonds_img,
'face' : 'two', 'rank': 2}
three_of_diamonds = {'suit': 'diamonds', 'points':  3,  'image':
three_of_diamonds_img, 'face' : 'three', 'rank': 3}
four_of_diamonds = {'suit': 'diamonds', 'points':  4,  'image':
four_of_diamonds_img, 'face' : 'four', 'rank': 4}
five_of_diamonds = {'suit': 'diamonds', 'points':  5,  'image':
five_of_diamonds_img, 'face' : 'five', 'rank': 5}
six_of_diamonds = {'suit': 'diamonds', 'points':  6,  'image': six_of_diamonds_img,
'face' : 'six', 'rank': 6}
seven_of_diamonds = {'suit': 'diamonds', 'points':  7,  'image':
seven_of_diamonds_img, 'face' : 'seven', 'rank': 7}
eight_of_diamonds = {'suit': 'diamonds', 'points':  8,  'image':
eight_of_diamonds_img, 'face' : 'eight', 'rank': 8}
nine_of_diamonds = {'suit': 'diamonds', 'points':  9,  'image':
nine_of_diamonds_img, 'face' : 'nine', 'rank': 9}
ten_of_diamonds = {'suit': 'diamonds', 'points':  10,  'image':
ten_of_diamonds_img, 'face' : 'ten', 'rank': 10}
jack_of_diamonds = {'suit': 'diamonds', 'points':  10,  'image':
jack_of_diamonds_img, 'face' : 'jack', 'rank': 11}
queen_of_diamonds = {'suit': 'diamonds', 'points':  10,  'image':
queen_of_diamonds_img, 'face' : 'queen', 'rank': 12}
king_of_diamonds = {'suit': 'diamonds', 'points':  10,  'image':
king_of_diamonds_img, 'face' : 'king', 'rank': 13}
ace_of_diamonds = {'suit': 'diamonds', 'points':  11,  'image':
ace_of_diamonds_img, 'face' : 'ace', 'rank': 14}
```

```
two_of_hearts = {'suit': 'hearts', 'points':  2,  'image': two_of_hearts_img,
'face' : 'two', 'rank': 2}
three_of_hearts = {'suit': 'hearts', 'points':  3,  'image': three_of_hearts_img,
'face' : 'three', 'rank': 3}
four_of_hearts = {'suit': 'hearts', 'points':  4,  'image': four_of_hearts_img,
'face' : 'four', 'rank': 4}
five_of_hearts = {'suit': 'hearts', 'points':  5,  'image': five_of_hearts_img,
'face' : 'five', 'rank': 5}
six_of_hearts = {'suit': 'hearts', 'points':  6,  'image': six_of_hearts_img,
'face' : 'six', 'rank': 6}
seven_of_hearts = {'suit': 'hearts', 'points':  7,  'image': seven_of_hearts_img,
'face' : 'seven', 'rank': 7}
eight_of_hearts = {'suit': 'hearts', 'points':  8,  'image': eight_of_hearts_img,
'face' : 'eight', 'rank': 8}
nine_of_hearts = {'suit': 'hearts', 'points':  9,  'image': nine_of_hearts_img,
'face' : 'nine', 'rank': 9}
ten_of_hearts = {'suit': 'hearts', 'points':  10,  'image': ten_of_hearts_img,
'face' : 'ten', 'rank': 10}
jack_of_hearts = {'suit': 'hearts', 'points':  10,  'image': jack_of_hearts_img,
'face' : 'jack', 'rank': 11}
queen_of_hearts = {'suit': 'hearts', 'points':  10,  'image': queen_of_hearts_img,
'face' : 'queen', 'rank': 12}
king_of_hearts = {'suit': 'hearts', 'points':  10,  'image': king_of_hearts_img,
'face' : 'king', 'rank': 13}
ace_of_hearts = {'suit': 'hearts', 'points':  11,  'image': ace_of_hearts_img,
'face' : 'ace', 'rank': 14}


two_of_spades = {'suit': 'spade', 'points':  2,  'image': two_of_spades_img, 'face'
: 'two', 'rank': 2}
three_of_spades = {'suit': 'spade', 'points':  3,  'image': three_of_spades_img,
'face' : 'three', 'rank': 3}
four_of_spades = {'suit': 'spade', 'points':  4,  'image': four_of_spades_img,
'face' : 'four', 'rank': 4}
five_of_spades = {'suit': 'spade', 'points':  5,  'image': five_of_spades_img,
'face' : 'five', 'rank': 5}
six_of_spades = {'suit': 'spade', 'points':  6,  'image': six_of_spades_img, 'face'
: 'six', 'rank': 6}
seven_of_spades = {'suit': 'spade', 'points':  7,  'image': seven_of_spades_img,
'face' : 'seven', 'rank': 7}
eight_of_spades = {'suit': 'spade', 'points':  8,  'image': eight_of_spades_img,
'face' : 'eight', 'rank': 8}
nine_of_spades = {'suit': 'spade', 'points':  9,  'image': nine_of_spades_img,
'face' : 'nine', 'rank': 9}
ten_of_spades = {'suit': 'spade', 'points':  10,  'image': ten_of_spades_img,
'face' : 'ten', 'rank': 10}
jack_of_spades = {'suit': 'spade', 'points':  10,  'image': jack_of_spades_img,
'face' : 'jack', 'rank': 11}
```

```
queen_of_spades = {'suit': 'spade', 'points':  10,  'image': queen_of_spades_img,
'face' : 'queen', 'rank': 12}
king_of_spades = {'suit': 'spade', 'points':  10,  'image': king_of_spades_img,
'face' : 'king', 'rank': 13}
ace_of_spades = {'suit': 'spade', 'points':  11,  'image': ace_of_spades_img,
'face' : 'ace', 'rank': 14}



# ### 3* create deck dictionary

# In[16]:


fullDeck = {}

fullDeck[1] = two_of_clubs
fullDeck[2] = three_of_clubs
fullDeck[3] = four_of_clubs
fullDeck[4] = five_of_clubs
fullDeck[5] = six_of_clubs
fullDeck[6] = seven_of_clubs
fullDeck[7] = eight_of_clubs
fullDeck[8] = nine_of_clubs
fullDeck[9] = ten_of_clubs
fullDeck[10] = jack_of_clubs
fullDeck[11] = queen_of_clubs
fullDeck[12] = king_of_clubs
fullDeck[13] = ace_of_clubs


fullDeck[14] = two_of_diamonds
fullDeck[15] = three_of_diamonds
fullDeck[16] = four_of_diamonds
fullDeck[17] = five_of_diamonds
fullDeck[18] = six_of_diamonds
fullDeck[19] = seven_of_diamonds
fullDeck[20] = eight_of_diamonds
fullDeck[21] = nine_of_diamonds
fullDeck[22] = ten_of_diamonds
fullDeck[23] = jack_of_diamonds
fullDeck[24] = queen_of_diamonds
fullDeck[25] = king_of_diamonds
fullDeck[26] = ace_of_diamonds


fullDeck[27] = two_of_hearts
fullDeck[28] = three_of_hearts
fullDeck[29] = four_of_hearts
fullDeck[30] = five_of_hearts
```

```
fullDeck[31] = six_of_hearts
fullDeck[32] = seven_of_hearts
fullDeck[33] = eight_of_hearts
fullDeck[34] = nine_of_hearts
fullDeck[35] = ten_of_hearts
fullDeck[36] = jack_of_hearts
fullDeck[37] = queen_of_hearts
fullDeck[38] = king_of_hearts
fullDeck[39] = ace_of_hearts


fullDeck[40] = two_of_spades
fullDeck[41] = three_of_spades
fullDeck[42] = four_of_spades
fullDeck[43] = five_of_spades
fullDeck[44] = six_of_spades
fullDeck[45] = seven_of_spades
fullDeck[46] = eight_of_spades
fullDeck[47] = nine_of_spades
fullDeck[48] = ten_of_spades
fullDeck[49] = jack_of_spades
fullDeck[50] = queen_of_spades
fullDeck[51] = king_of_spades
fullDeck[52] = ace_of_spades


# ### 4* load slot reel images

# In[17]:


clone_img = Image.open("slotpics/6.png").resize((135, 135))
jabba_img = Image.open("slotpics/8.png").resize((135, 135))
kylo_img = Image.open("slotpics/10.png").resize((135, 135))
r2d2_img = Image.open("slotpics/11.png").resize((135, 135))
c3po_img = Image.open("slotpics/12.png").resize((135, 135))
storm_img = Image.open("slotpics/14.png").resize((135, 135))
chewie_img = Image.open("slotpics/16.png").resize((135, 135))
vader_img = Image.open("slotpics/17.png").resize((135, 135))
ahsoka_img = Image.open("slotpics/20.png").resize((135, 135))
mando_img = Image.open("slotpics/21.png").resize((135, 135))
grogu_img = Image.open("slotpics/22.png").resize((135, 135))
boba_img = Image.open("slotpics/24.png").resize((135, 135))
maul_img = Image.open("slotpics/25.png").resize((135, 135))


# ### 5* create slot reeal symbol dictionaries

# In[18]:
```

```
clone_dict = {'rank' : 1, 'image': clone_img, 'side' : 'light', 'id' : '1'}
jabba_dict = {'rank' : 2, 'image': jabba_img, 'side' : 'dark', 'id' : '2'}
r2d2_dict = {'rank' : 0, 'image': r2d2_img, 'side' : 'wild', 'id' : '3'}
c3po_dict = {'rank' : 2, 'image': c3po_img, 'side' : 'light', 'id' : '4'}
storm_dict = {'rank' : 1, 'image': storm_img, 'side' : 'dark', 'id' : '5'}
chewie_dict = {'rank' : 2, 'image': chewie_img, 'side' : 'light', 'id' : '6'}
vader_dict = {'rank' : 4, 'image': vader_img, 'side' : 'dark', 'id' : '7'}
ahsoka_dict = {'rank' : 3, 'image': ahsoka_img, 'side' : 'light', 'id' : '8'}
mando_dict = {'rank' : 2, 'image': mando_img, 'side' : 'light', 'id' : '9'}
grogu_dict = {'rank' : 4, 'image': grogu_img, 'side' : 'light', 'id' : '10'}
boba_dict = {'rank' : 2, 'image': boba_img, 'side' : 'dark', 'id' : '11'}
maul_dict = {'rank' : 3, 'image': maul_img, 'side' : 'dark', 'id' : '12'}
kylo_dict = {'rank' : 2, 'image': kylo_img, 'side' : 'dark', 'id' : '13'}



# ### 6* initialize slot reel symbol list and add dictionaries

# In[19]:


slotRoll1 = [0]*39

slotRoll1[0] = clone_dict
slotRoll1[1] = clone_dict
slotRoll1[2] = clone_dict
slotRoll1[3] = clone_dict
slotRoll1[4] = clone_dict
slotRoll1[5] = clone_dict
slotRoll1[6] = storm_dict
slotRoll1[7] = storm_dict
slotRoll1[8] = storm_dict
slotRoll1[9] = storm_dict
slotRoll1[10] = storm_dict
slotRoll1[11] = storm_dict
slotRoll1[12] = c3po_dict
slotRoll1[13] = c3po_dict
slotRoll1[14] = c3po_dict
slotRoll1[15] = jabba_dict
slotRoll1[16] = jabba_dict
slotRoll1[17] = jabba_dict
slotRoll1[18] = chewie_dict
slotRoll1[19] = chewie_dict
slotRoll1[20] = chewie_dict
slotRoll1[21] = kylo_dict
slotRoll1[22] = kylo_dict
slotRoll1[23] = kylo_dict
slotRoll1[24] = mando_dict
slotRoll1[25] = mando_dict
```

```
slotRoll1[26] = mando_dict
slotRoll1[27] = boba_dict
slotRoll1[28] = boba_dict
slotRoll1[29] = boba_dict
slotRoll1[30] = ahsoka_dict
slotRoll1[31] = ahsoka_dict
slotRoll1[32] = maul_dict
slotRoll1[33] = maul_dict
slotRoll1[34] = grogu_dict
slotRoll1[35] = grogu_dict
slotRoll1[36] = vader_dict
slotRoll1[37] = vader_dict
slotRoll1[38] = r2d2_dict


# ### 7* blackjack gui

# In[20]:


def openBJGame():

    formatter = "${:,.2f}"

    playerHand = []
    playerTotal = 0
    playerDraws = 0
    playerBank = 0.00
    playerDeposit = 0.00
    playerBet = 0.00

    dealerHand = []
    dealerTotal = 0
    dealerDraws = 0

    tempDeck = fullDeck.copy()

    def startShow():

        webbrowser.open_new(r"https://www.youtube.com/watch?v=NfBRjQROH5c")


    def depositFunds():

        amount = depositAmount.get()

        if not amount.isnumeric():
            messagebox.showerror('Incorrect Format', 'Please enter only numbers for
the deposit amount.')
        else:
```

```
            amount = float(amount)
            nonlocal playerBank
            playerBank += amount
            walletAmount['text'] = f"{formatter.format(playerBank)}"

    def placeBet(n):

        nonlocal playerBet
        playerBet = n

        nonlocal playerBank

        if playerBank >= playerBet:

            playerBank -= playerBet

            walletAmount['text'] = f"{formatter.format(playerBank)}"

            bet10Button['state'] = 'disabled'
            bet100Button['state'] = 'disabled'
            bet1000Button['state'] = 'disabled'
            drawButton['state'] = 'normal'

        else:
            messagebox.showerror('Insufficient Funds', 'Please deposit more money
before betting.')


    def bet10():
        placeBet(10)
    def bet100():
        placeBet(100)
    def bet1000():
        placeBet(1000)

    def resetGame():

        nonlocal playerHand
        playerHand = []

        nonlocal playerDraws
        playerDraws = 0

        nonlocal playerTotal
        playerTotal = 0

        playerCard1['image'] = card_reverse_tkimg
        playerCard1.image = card_reverse_tkimg
        playerCard2['image'] = card_reverse_tkimg
        playerCard2.image = card_reverse_tkimg
```

```
    playerCard3['image'] = card_reverse_tkimg
    playerCard3.image = card_reverse_tkimg
    playerCard4['image'] = card_reverse_tkimg
    playerCard4.image = card_reverse_tkimg
    playerCard5['image'] = card_reverse_tkimg
    playerCard5.image = card_reverse_tkimg

    playerScore['text'] = f"Player Score: {playerTotal}"

    nonlocal dealerHand
    dealerHand = []

    nonlocal dealerDraws
    dealerDraws = 0

    nonlocal dealerTotal
    dealerTotal = 0

    dealerCard1['image'] = card_reverse_tkimg
    dealerCard1.image = card_reverse_tkimg
    dealerCard2['image'] = card_reverse_tkimg
    dealerCard2.image = card_reverse_tkimg
    dealerCard3['image'] = card_reverse_tkimg
    dealerCard3.image = card_reverse_tkimg
    dealerCard4['image'] = card_reverse_tkimg
    dealerCard4.image = card_reverse_tkimg
    dealerCard5['image'] = card_reverse_tkimg
    dealerCard5.image = card_reverse_tkimg

    dealerScore['text'] = f"Dealer Score: {dealerTotal}"

    bet10Button['state'] = 'normal'
    bet100Button['state'] = 'normal'
    bet1000Button['state'] = 'normal'

    drawButton['text'] = "Deal"



def concludeGame():

    nonlocal playerTotal
    nonlocal playerBet
    nonlocal playerBank
    nonlocal playerHand
    nonlocal dealerTotal

    reset = False

    if sorted(playerHand) in
```

```
(['ace','ten'],['ace','jack'],['ace','queen'],['ace','king']) and
sorted(dealerHand) not in
(['ace','ten'],['ace','jack'],['ace','queen'],['ace','king']):

            nextstep = messagebox.askquestion('BLACKJACK!!!!', 'You got
Blackjack!!! You get a 3:2 payout!.\nDo you want to play again?')

            playerBank += playerBet + playerBet*3/2

            if nextstep == 'yes':
                reset = True
            else:
                reset = False

        else:

            if playerTotal > 21:

                nextstep = messagebox.askquestion('BUST!', 'Your hand is greater
than 21.\nDo you want to play again?')

                if nextstep == 'yes':
                    reset = True
                else:
                    reset = False

            elif playerTotal == dealerTotal:

                nextstep = messagebox.askquestion('Draw/Push', 'Your hand ties the
dealer\'s.\nDo you want to play again?')

                playerBank += playerBet

                if nextstep == 'yes':
                    reset = True
                else:
                    reset = False

            elif playerTotal < dealerTotal:

                if dealerTotal <= 21:

                    nextstep = messagebox.askquestion('You lose.', 'The dealer\'s
hand beats yours.\nDo you want to play again?')

                    if nextstep == 'yes':
                        reset = True
                    else:
                        reset = False
```

```
                else:

                    nextstep = messagebox.askquestion('Dealer Busts!!!', 'The
dealer\'s hand is over 21, you win!\nDo you want to play again?')

                    playerBank += playerBet*2

                    if nextstep == 'yes':
                        reset = True
                    else:
                        reset = False
            else:

                nextstep = messagebox.askquestion('You win!!!', 'Your hand beats
the dealer\'s, you win!\nDo you want to play again?')

                playerBank += playerBet*2

                if nextstep == 'yes':
                    reset = True
                else:
                    reset = False

        walletAmount['text'] = f"{formatter.format(playerBank)}"

        if reset:
            resetGame()
        else:
            bjWindow.destroy()


    def stay():

        drawButton['state'] = 'disabled'

        nonlocal dealerTotal
        nonlocal playerTotal

        while dealerTotal < 17 and playerTotal != 0:
            dealerDraw()


    def dealerDraw():

        nonlocal tempDeck

        if len(tempDeck) < 10:
            tempDeck = fullDeck.copy()

        nonlocal dealerDraws
```

```
    dealerDraws += 1

    draw = random.choice(list(tempDeck.items()))

    del tempDeck[draw[0]]

    new_card_img = itk.PhotoImage(draw[1]['image'])

    if dealerDraws == 1:
        dealerCard1['image'] = new_card_img
        dealerCard1.image = new_card_img
    elif dealerDraws == 2:
        dealerCard2['image'] = new_card_img
        dealerCard2.image = new_card_img
    elif dealerDraws == 3:
        dealerCard3['image'] = new_card_img
        dealerCard3.image = new_card_img
    elif dealerDraws == 4:
        dealerCard4['image'] = new_card_img
        dealerCard4.image = new_card_img
    else:
        dealerCard5['image'] = new_card_img
        dealerCard5.image = new_card_img

    nonlocal dealerHand

    dealerHand.append(draw[1]['face'])

    nonlocal dealerTotal

    dealerTotal += draw[1]['points']

    dealerScore['text'] = f"Dealer Score: {dealerTotal}"

    if dealerTotal >= 17:
        concludeGame()


def drawCard():

    nonlocal tempDeck

    if len(tempDeck) < 10:
        tempDeck = fullDeck.copy()

    nonlocal playerDraws
    playerDraws += 1

    draw = random.choice(list(tempDeck.items()))
```

```
    del tempDeck[draw[0]]

    #return draw[1]['points'], draw[1]['image']

    new_card_img = itk.PhotoImage(draw[1]['image'])
    face = draw[1]['face']

    if playerDraws == 1:
        playerCard1['image'] = new_card_img
        playerCard1.image = new_card_img
    elif playerDraws == 2:
        playerCard2['image'] = new_card_img
        playerCard2.image = new_card_img
    elif playerDraws == 3:
        playerCard3['image'] = new_card_img
        playerCard3.image = new_card_img
    elif playerDraws == 4:
        playerCard4['image'] = new_card_img
        playerCard4.image = new_card_img
    else:
        playerCard5['image'] = new_card_img
        playerCard5.image = new_card_img

    nonlocal playerHand

    playerHand.append(draw[1]['face'])

    nonlocal playerTotal

    playerTotal += draw[1]['points']

    playerScore['text'] = f"Player Score: {playerTotal}"
    drawButton['text'] = "Hit"

    if playerTotal >= 21:

        #drawButton['state'] = 'disabled'
        stay()


    if playerDraws == 2:
        dealerDraw()

    if playerDraws == 1:
        drawCard()



bjWindow = Toplevel()
bjWindow.geometry('700x870')
```

```
    bjWindow.config(bg='green')

    bjFrame = Frame(bjWindow, bg='green')
    bjFrame.grid(row = 0)

    for i in range(9):
        bjFrame.grid_rowconfigure(i, weight=1)

    for i in range(5):
        bjFrame.grid_columnconfigure(i, weight=1)


    bjlogo_img = Image.open("bjlogocustom.png").resize((650, 175))
    bjlogo_tkimg = itk.PhotoImage(bjlogo_img)

    bjlogo = Label(bjFrame, image = bjlogo_tkimg, bg='green')
    bjlogo.image = bjlogo_tkimg
    bjlogo.grid(row = 0, column=0, columnspan=5, pady=5, sticky='news')


    card_reverse_tkimg = itk.PhotoImage(card_reverse_img)


    dealerPoints = Label(bjFrame, bg='green', text = "Dealer hand: ", fg='gold',
font=('Times New Roman',15, 'bold'))
    dealerPoints.grid(row = 1, column=0, columnspan = 2, sticky = 'w')
    dealerScore = Label(bjFrame, bg='green', text = "Dealer Score: 0 ", fg='gold',
font=('Times New Roman',12, 'bold'), borderwidth=3, relief="ridge", width = 13)
    dealerScore.grid(row = 1, column=4, sticky = 'e', ipadx=3, ipady=5)

    dealerCard1 = Label(bjFrame, image = card_reverse_tkimg, bg='green')
    dealerCard1.image = card_reverse_tkimg
    dealerCard1.grid(row = 2, column = 0, pady=15, sticky='news')
    dealerCard2 = Label(bjFrame, image = card_reverse_tkimg, bg='green')
    dealerCard2.image = card_reverse_tkimg
    dealerCard2.grid(row = 2, column = 1, pady=15, sticky='news')
    dealerCard3 = Label(bjFrame, image = card_reverse_tkimg, bg='green')
    dealerCard3.image = card_reverse_tkimg
    dealerCard3.grid(row = 2, column = 2, pady=15, sticky='news')
    dealerCard4 = Label(bjFrame, image = card_reverse_tkimg, bg='green')
    dealerCard4.image = card_reverse_tkimg
    dealerCard4.grid(row = 2, column = 3, pady=15, sticky='news')
    dealerCard5 = Label(bjFrame, image = card_reverse_tkimg, bg='green')
    dealerCard5.image = card_reverse_tkimg
    dealerCard5.grid(row = 2, column = 4, pady=15, sticky='news')


    playerPoints = Label(bjFrame, bg='green', text = "Player hand: ", fg='gold',
font=('Times New Roman',15, 'bold'))
    playerPoints.grid(row = 4, column=0, columnspan = 2, sticky = 'w')
```

```
    playerScore = Label(bjFrame, bg='green', text = "Player Score: 0 ", fg='gold',
font=('Times New Roman',12, 'bold'), borderwidth=3, relief="ridge", width = 13)
    playerScore.grid(row = 4, column=4, sticky = 'e', ipadx=3, ipady=5)

    playerCard1 = Label(bjFrame, image = card_reverse_tkimg, bg='green', width =
15)
    playerCard1.image = card_reverse_tkimg
    playerCard1.grid(row = 5, column = 0, pady=15, sticky='news')
    playerCard2 = Label(bjFrame, image = card_reverse_tkimg, bg='green', width =
15)
    playerCard2.image = card_reverse_tkimg
    playerCard2.grid(row = 5, column = 1, pady=15, sticky='news')
    playerCard3 = Label(bjFrame, image = card_reverse_tkimg, bg='green', width =
15)
    playerCard3.image = card_reverse_tkimg
    playerCard3.grid(row = 5, column = 2, pady=15, sticky='news')
    playerCard4 = Label(bjFrame, image = card_reverse_tkimg, bg='green', width =
15)
    playerCard4.image = card_reverse_tkimg
    playerCard4.grid(row = 5, column = 3, pady=15, sticky='news')
    playerCard5 = Label(bjFrame, image = card_reverse_tkimg, bg='green', width =
15)
    playerCard5.image = card_reverse_tkimg
    playerCard5.grid(row = 5, column = 4, pady=15, sticky='news')

    bet10Button = Button(bjFrame, text = "Bet $10", command = bet10, font=('Times
New Roman',12, 'bold'), width = 15)
    bet10Button.grid(row = 6, column=0, pady=10)
    bet100Button = Button(bjFrame, text = "Bet $100", command = bet100,
font=('Times New Roman',12, 'bold'), width = 15)
    bet100Button.grid(row = 7, column=0, pady=10)
    bet1000Button = Button(bjFrame, text = "Bet $1000", command = bet1000,
font=('Times New Roman',12, 'bold'), width = 15)
    bet1000Button.grid(row = 8, column=0, pady=10)
    showButton = Button(bjFrame, text = "Entertainment", command = startShow,
font=('Times New Roman',12, 'bold'), width = 15)
    showButton.grid(row = 9, column=0, pady=10)

    drawButton = Button(bjFrame, text = "Deal", command = drawCard, font=('Times
New Roman',12, 'bold'), width = 15)
    drawButton.grid(row = 6, column=2, pady=10)
    drawButton['state'] = 'disabled'

    stayButton = Button(bjFrame, text = "Stand", command = stay, font=('Times New
Roman',12, 'bold'), width = 15)
    stayButton.grid(row = 7, column=2, pady=10)

    quitButton = Button(bjFrame, text = "Quit Game", command = bjWindow.destroy,
font=('Times New Roman',12, 'bold'))
    quitButton.grid(row = 9, column=2, pady=15)
```

```
    depositButton = Button(bjFrame, text = "Deposit Funds:", font=('Times New
Roman',12, 'bold'), command=depositFunds, width = 15)
    depositButton.grid(row = 6, column = 4)
    depositAmount = Entry(bjFrame, font=('Times New Roman',15, 'bold'),
borderwidth=5, relief="ridge", width = 13, justify='r')
    depositAmount.grid(row = 7, column = 4, pady=5)

    walletLabel = Label(bjFrame, text = "Wallet:", bg='green', fg='gold',
font=('Times New Roman',15, 'bold'))
    walletLabel.grid(row = 8, column = 4, sticky='sew', padx = 20)
    walletAmount = Label(bjFrame, text = f"{formatter.format(playerBank)}",
bg='green', fg='gold', font=('Times New Roman',15, 'bold'), borderwidth=5,
relief="ridge", width = 10)
    walletAmount.grid(row = 9, column = 4, ipady=5, sticky='n')


    bjWindow.title('Boyle Casino - Blackjack')

    bjWindow.grid_rowconfigure(0, weight=1)
    bjWindow.grid_columnconfigure(0, weight=1)

    bjWindow.mainloop()


# ### 8* poker gui

# In[21]:


def openVPGame():

    pausePokerMusic = False
    global welcomePlayedLast
    welcomePlayedLast = False
    global pokerPlayedLast
    pokerPlayedLast = True
    global slotsPlayedLast
    slotsPlayedLast = False

    def stopMusic():

        nonlocal pausePokerMusic
        global welcomePlayedLast
        global pokerPlayedLast
        global slotsPlayedLast

        if pausePokerMusic:

            if not pokerPlayedLast:
```

```
                pygame.mixer.music.load(pokerSong)
                pygame.mixer.music.play(-1)
            else:
                pygame.mixer.music.unpause()

            pausePokerMusic = False

            pokerPlayedLast = True
            welcomePlayedLast = False
            slotsPlayedLast = False

        else:

            pygame.mixer.music.pause()

            pausePokerMusic = True

    def closevpWindow():

        pygame.mixer.music.stop()
        vpWindow.destroy()


    formatter = "${:,.2f}"

    bet1pay = [250.00, 50.00, 25.00, 6.00, 5.00, 4.00, 3.00, 2.00, 1.00]
    # bet5pay = [1250.00, 250.00, 125.00, 30.00, 25.00, 20.00, 15.00, 10.00, 5.00]
    # bet10pay = [2500.00, 500.00, 250.00, 60.00, 50.00, 40.00, 30.00, 20.00,
10.00]
    # bet50pay = [12500.00, 2500.00, 1250.00, 300.00, 250.00, 200.00, 150.00,
100.00, 50.00]
    # bet100pay = [25000.00, 5000.00, 2500.00, 600.00, 500.00, 400.00, 300.00,
200.00, 100.00]

    paylist = []

    holdList = [False,False,False,False,False]

    turns = 0
    ranks = []
    suits = []
    bank = 0.00
    deposit = 0.00
    bet = 0.00


    tempDeck = fullDeck.copy()

    def depositFunds():
```

```
        amount = depositAmount.get()

        if not amount.isnumeric():
            messagebox.showerror('Incorrect Format', 'Please enter only numbers for
the deposit amount.')
        else:
            amount = float(amount)
            nonlocal bank
            bank += amount
            walletAmount['text'] = f"{formatter.format(bank)}"

    def placeBet(n):

        nonlocal bet
        bet = n

        nonlocal bank

        if bank >= bet:

            bank -= bet

            walletAmount['text'] = f"{formatter.format(bank)}"

            bet1Button['state'] = 'disabled'
            bet5Button['state'] = 'disabled'
            bet10Button['state'] = 'disabled'
            bet50Button['state'] = 'disabled'
            bet100Button['state'] = 'disabled'

            dealButton['state'] = 'normal'

        else:
            messagebox.showerror('Insufficient Funds', 'Please deposit more money
before betting.')

    def bet1():

        nonlocal bet1pay
        nonlocal paylist

        paylist = bet1pay.copy()

        placeBet(1)

    def bet5():

        nonlocal bet1pay
        nonlocal paylist
```

```
    paylist = bet1pay.copy()*5

    placeBet(5)

def bet10():

    nonlocal bet1pay
    nonlocal paylist

    paylist = bet1pay.copy()*10

    placeBet(10)

def bet50():

    nonlocal bet1pay
    nonlocal paylist

    paylist = bet1pay.copy()*50

    placeBet(50)

def bet100():

    nonlocal bet1pay
    nonlocal paylist

    paylist = bet1pay.copy()*100

    placeBet(100)


def resetGame():

    nonlocal ranks
    ranks = []

    nonlocal suits
    suits = []

    nonlocal turns
    turns = 0

    nonlocal holdList
    holdList = [False,False,False,False,False]

    playerCard1['image'] = card_reverse_tkimg
    playerCard1.image = card_reverse_tkimg
    playerCard2['image'] = card_reverse_tkimg
    playerCard2.image = card_reverse_tkimg
```

```
    playerCard3['image'] = card_reverse_tkimg
    playerCard3.image = card_reverse_tkimg
    playerCard4['image'] = card_reverse_tkimg
    playerCard4.image = card_reverse_tkimg
    playerCard5['image'] = card_reverse_tkimg
    playerCard5.image = card_reverse_tkimg

    bet1Button['state'] = 'normal'
    bet5Button['state'] = 'normal'
    bet10Button['state'] = 'normal'
    bet50Button['state'] = 'normal'
    bet100Button['state'] = 'normal'


    holdButton1['state'] = 'disabled'
    holdButton2['state'] = 'disabled'
    holdButton3['state'] = 'disabled'
    holdButton4['state'] = 'disabled'
    holdButton5['state'] = 'disabled'

    nonlocal tempDeck

    if len(tempDeck) < 10:
        tempDeck = fullDeck.copy()

def playHand():


    nonlocal ranks
    nonlocal suits
    nonlocal paylist
    nonlocal bank

    reset = False

    ranks.sort()

    straight = True

    for i in range(len(ranks)-1):

        if ranks[i + 1] - ranks[i] != 1:
            straight = False


    if len(set(suits)) == 1 and straight:

        if min(ranks) > 9:

            nextstep = messagebox.askquestion('ROYAL FLUSH!!!!!!!!!',
```

```
'JACKPOT!!!!! You got a royal flush!!!!!.\nDo you want to play again?')

                bank += paylist[0]

                if nextstep == 'yes':
                    reset = True
                else:
                    reset = False
            else:

                nextstep = messagebox.askquestion('STRAIGHT FLUSH!!!!!',
'Congrats!!!!! You got a straight flush!!!!!.\nDo you want to play again?')

                bank += paylist[1]

                if nextstep == 'yes':
                    reset = True
                else:
                    reset = False


        elif len(set(suits)) == 1:

            nextstep = messagebox.askquestion('FLUSH!!!', 'Congrats! You got a
flush!!!.\nDo you want to play again?')

            bank += paylist[4]

            if nextstep == 'yes':
                reset = True
            else:
                reset = False

        elif straight:

            nextstep = messagebox.askquestion('STRAIGHT!!!!!', 'Congrats! You got a
straight!!!.\nDo you want to play again?')

            bank += paylist[5]

            if nextstep == 'yes':
                reset = True
            else:
                reset = False

        else:

            count = 1
            rank = 0
```

```
            for i in range(len(ranks)):

                if count < ranks.count(ranks[i]):
                    count = ranks.count(ranks[i])
                    rank = ranks[i]

            if count >= 3:
                if count > 3:

                    nextstep = messagebox.askquestion('4 OF A KIND!!!!', 'Congrats!
You got four of a kind!!!!.\nDo you want to play again?')

                    bank += paylist[2]

                    if nextstep == 'yes':
                        reset = True
                    else:
                        reset = False

                elif len(set(ranks)) == 2:

                    nextstep = messagebox.askquestion('FULL HOUSE!!!!!', 'Congrats!
You got a full house!!!!!.\nDo you want to play again?')

                    bank += paylist[3]

                    if nextstep == 'yes':
                        reset = True
                    else:
                        reset = False

                else:

                    nextstep = messagebox.askquestion('3 OF A KIND!!!', 'You got a
three of a kind!!!.\nDo you want to play again?')

                    bank += paylist[6]

                    if nextstep == 'yes':
                        reset = True
                    else:
                        reset = False

            else:

                if len(set(ranks)) == 3:

                    nextstep = messagebox.askquestion('TWO PAIR!!!!', 'You got two
pairs!!!!.\nDo you want to play again?')
```

```
                    bank += paylist[7]

                    if nextstep == 'yes':
                        reset = True
                    else:
                        reset = False

                else:

                    if rank > 10:

                        nextstep = messagebox.askquestion('JACKS OR BETTER!', 'You
got jacks or better!!.\nDo you want to play again?')

                        bank += paylist[8]

                        if nextstep == 'yes':
                            reset = True
                        else:
                            reset = False
                    else:

                        nextstep = messagebox.askquestion('PLAY AGAIN???', 'No
winning hands this round.\nDo you want to play again?')

                        if nextstep == 'yes':
                            reset = True
                        else:
                            reset = False

        walletAmount['text'] = f"{formatter.format(bank)}"

        if reset:
            resetGame()
        else:
            vpWindow.destroy()


    def hold1():

        nonlocal holdList

        holdList[0] = True

        holdButton1['state'] = 'disabled'

    def hold2():

        nonlocal holdList
```

```
    holdList[1] = True

    holdButton2['state'] = 'disabled'

def hold3():

    nonlocal holdList

    holdList[2] = True

    holdButton3['state'] = 'disabled'

def hold4():

    nonlocal holdList

    holdList[3] = True

    holdButton4['state'] = 'disabled'

def hold5():

    nonlocal holdList

    holdList[4] = True

    holdButton5['state'] = 'disabled'



def dealCards():

    nonlocal holdList
    nonlocal tempDeck
    nonlocal turns
    nonlocal ranks
    nonlocal suits


    if bool(ranks):
        deletes = 0

        for i in range(5):

            if not holdList[i]:
                del ranks[i-deletes]
                del suits[i-deletes]
                deletes += 1

    if not holdList[0]:
```

```
        draw = random.choice(list(tempDeck.items()))

        new_card_img = itk.PhotoImage(draw[1]['image'])

        del tempDeck[draw[0]]

        playerCard1['image'] = new_card_img
        playerCard1.image = new_card_img

        ranks.append(draw[1]['rank'])
        suits.append(draw[1]['suit'])

        holdButton1['state'] = 'normal'

    if not holdList[1]:

        draw = random.choice(list(tempDeck.items()))

        new_card_img = itk.PhotoImage(draw[1]['image'])

        del tempDeck[draw[0]]

        playerCard2['image'] = new_card_img
        playerCard2.image = new_card_img

        ranks.append(draw[1]['rank'])
        suits.append(draw[1]['suit'])

        holdButton2['state'] = 'normal'

    if not holdList[2]:

        draw = random.choice(list(tempDeck.items()))

        new_card_img = itk.PhotoImage(draw[1]['image'])

        del tempDeck[draw[0]]

        playerCard3['image'] = new_card_img
        playerCard3.image = new_card_img

        ranks.append(draw[1]['rank'])
        suits.append(draw[1]['suit'])

        holdButton3['state'] = 'normal'

    if not holdList[3]:

        draw = random.choice(list(tempDeck.items()))
```

```
        new_card_img = itk.PhotoImage(draw[1]['image'])

        del tempDeck[draw[0]]

        playerCard4['image'] = new_card_img
        playerCard4.image = new_card_img

        ranks.append(draw[1]['rank'])
        suits.append(draw[1]['suit'])

        holdButton4['state'] = 'normal'

    if not holdList[4]:

        draw = random.choice(list(tempDeck.items()))

        new_card_img = itk.PhotoImage(draw[1]['image'])

        del tempDeck[draw[0]]

        playerCard5['image'] = new_card_img
        playerCard5.image = new_card_img

        ranks.append(draw[1]['rank'])
        suits.append(draw[1]['suit'])

        holdButton5['state'] = 'normal'


    turns += 1

    if turns == 2:

        dealButton['state'] = 'disabled'
        playHand()



vpWindow = Toplevel()
vpWindow.geometry('800x900')
vpWindow.config(bg='blue')

vpFrame = Frame(vpWindow, bg='blue')
vpFrame.grid(row = 0)

for i in range(10):
    vpFrame.grid_rowconfigure(i, weight=1)
```

```
    for i in range(5):
        vpFrame.grid_columnconfigure(i, weight=1)

    vplogo_img = Image.open("pokerlabel.png").resize((600, 80))
    vplogo_tkimg = itk.PhotoImage(vplogo_img)

    vplogo = Label(vpFrame, image = vplogo_tkimg, bg='blue')
    vplogo.image = vplogo_tkimg
    vplogo.grid(row = 0, column=0, columnspan=5, pady=5)

    vptable_img = Image.open("pokertable.png").resize((730, 230))
    vptable_tkimg = itk.PhotoImage(vptable_img)

    vptable = Label(vpFrame, image = vptable_tkimg, bg='blue', borderwidth=5,
relief="ridge")
    vptable.image = vptable_tkimg
    vptable.grid(row = 1, column=0, columnspan=5, pady=5, sticky='news')


    card_reverse_tkimg = itk.PhotoImage(card_reverse_img)


    handLabel = Label(vpFrame, bg='blue', text = "Hand: ", fg='gold', font=('Times
New Roman',15, 'bold'))
    handLabel.grid(row = 2, column=0, columnspan = 2, sticky = 'w')
    stopButton = Button(vpFrame, text = 'Pause/Play Music', command = stopMusic,
font=('Times New Roman',10, 'bold'))
    stopButton.grid(row = 2, column=4, sticky = 'e', ipadx=3, ipady=5)

    playerCard1 = Label(vpFrame, image = card_reverse_tkimg, bg='blue', width = 15)
    playerCard1.image = card_reverse_tkimg
    playerCard1.grid(row = 3, column = 0, pady=15, sticky='news')
    playerCard2 = Label(vpFrame, image = card_reverse_tkimg, bg='blue', width = 15)
    playerCard2.image = card_reverse_tkimg
    playerCard2.grid(row = 3, column = 1, pady=15, sticky='news')
    playerCard3 = Label(vpFrame, image = card_reverse_tkimg, bg='blue', width = 15)
    playerCard3.image = card_reverse_tkimg
    playerCard3.grid(row = 3, column = 2, pady=15, sticky='news')
    playerCard4 = Label(vpFrame, image = card_reverse_tkimg, bg='blue', width = 15)
    playerCard4.image = card_reverse_tkimg
    playerCard4.grid(row = 3, column = 3, pady=15, sticky='news')
    playerCard5 = Label(vpFrame, image = card_reverse_tkimg, bg='blue', width = 15)
    playerCard5.image = card_reverse_tkimg
    playerCard5.grid(row = 3, column = 4, pady=15, sticky='news')


    holdButton1 = Button(vpFrame, text = "Hold",  state = 'disabled', font=('Times
New Roman',12, 'bold'), command = hold1)
    holdButton1.grid(row = 4, column = 0, pady=15, sticky='news')
    holdButton2 = Button(vpFrame, text = "Hold",  state = 'disabled', font=('Times
```

```
New Roman',12, 'bold'), command = hold2, width = 15)
    holdButton2.grid(row = 4, column = 1, pady=15, sticky='news')
    holdButton3 = Button(vpFrame, text = "Hold",  state = 'disabled', font=('Times
New Roman',12, 'bold'), command = hold3)
    holdButton3.grid(row = 4, column = 2, pady=15, sticky='news')
    holdButton4 = Button(vpFrame, text = "Hold",  state = 'disabled', font=('Times
New Roman',12, 'bold'), command = hold4, width = 15)
    holdButton4.grid(row = 4, column = 3, pady=15, sticky='news')
    holdButton5 = Button(vpFrame, text = "Hold", state = 'disabled', font=('Times
New Roman',12, 'bold'), command = hold5)
    holdButton5.grid(row = 4, column = 4, pady=15, sticky='news')

    bet1Button = Button(vpFrame, text = "Bet $1", command = bet1, font=('Times New
Roman',12, 'bold'), width = 15)
    bet1Button.grid(row = 5, column=0, pady=10)
    bet5Button = Button(vpFrame, text = "Bet $5", command = bet5, font=('Times New
Roman',12, 'bold'), width = 15)
    bet5Button.grid(row = 6, column=0, pady=10)
    bet10Button = Button(vpFrame, text = "Bet $10", command = bet10, font=('Times
New Roman',12, 'bold'), width = 15)
    bet10Button.grid(row = 7, column=0, pady=10)
    bet50Button = Button(vpFrame, text = "Bet $50", command = bet50, font=('Times
New Roman',12, 'bold'), width = 15)
    bet50Button.grid(row = 8, column=0, pady=10)
    bet100Button = Button(vpFrame, text = "Bet $100", command = bet100,
font=('Times New Roman',12, 'bold'), width = 15)
    bet100Button.grid(row = 9, column=0, pady=10)


    dealButton = Button(vpFrame, text = "Deal", command = dealCards, font=('Times
New Roman',15, 'bold'))
    dealButton.grid(row = 6, rowspan = 2, column=2, pady=10, sticky='news')
    dealButton['state'] = 'disabled'

    quitButton = Button(vpFrame, text = "Quit Game", command = closevpWindow,
font=('Times New Roman',12, 'bold'), width = 15)
    quitButton.grid(row = 9, column=2, pady=10, sticky='news')

    depositButton = Button(vpFrame, text = "Deposit Funds:", font=('Times New
Roman',12, 'bold'), command=depositFunds, width = 15)
    depositButton.grid(row = 6, column = 4)
    depositAmount = Entry(vpFrame, font=('Times New Roman',15, 'bold'),
borderwidth=5, relief="ridge", width = 13, justify='r')
    depositAmount.grid(row = 7, column = 4, pady=5)

    walletLabel = Label(vpFrame, text = "Wallet:", bg='blue', fg='gold',
font=('Times New Roman',15, 'bold'))
    walletLabel.grid(row = 8, column = 4, sticky='sew', padx = 20)
    walletAmount = Label(vpFrame, text = f"{formatter.format(bank)}", bg='blue',
fg='gold', font=('Times New Roman',15, 'bold'), borderwidth=5, relief="ridge",
```

```
width = 10)
    walletAmount.grid(row = 9, column = 4, ipady=5, sticky='n')


    vpWindow.title('Boyle Casino - Video Poker')

    vpWindow.grid_rowconfigure(0, weight=1)
    vpWindow.grid_columnconfigure(0, weight=1)

    pokerSong = r'music/frank.mp3'

    pygame.init()
    pygame.mixer.init()
    pygame.mixer.music.load(pokerSong)
    pygame.mixer.music.play(-1)

    vpWindow.mainloop()


# ### 9* slots gui

# In[22]:


def openSlots():

    pauseSlotsMusic = False
    global welcomePlayedLast
    welcomePlayedLast = False
    global pokerPlayedLast
    pokerPlayedLast = False
    global slotsPlayedLast
    slotsPlayedLast = True

    formatter = "${:,.2f}"

    slotBet = 0
    slotBank = 0

    slotRoll2 = slotRoll1.copy()
    slotRoll3 = slotRoll1.copy()
    slotRoll3[34] = ahsoka_dict
    slotRoll3[36] = maul_dict

    bgCounter = 0

    def closeSlotWindow():

        pygame.mixer.music.stop()
        slotWindow.destroy()
```

```
def stopMusic():

    nonlocal pauseSlotsMusic
    global welcomePlayedLast
    global pokerPlayedLast
    global slotsPlayedLast

    if pauseSlotsMusic:

        if not slotsPlayedLast:

            nonlocal bgCounter

            if bgCounter == 1:
                slotSong = r'music/cantina.mp3'
            elif bgCounter == 2:
                slotSong = r'music/forcetheme.mp3'
            elif bgCounter == 3:
                slotSong = r'music/acrossthestars.mp3'
            elif bgCounter == 4:
                slotSong = r'music/impmarch.mp3'
            elif bgCounter == 5:
                slotSong = r'music/swfinale.mp3'
            else:
                slotSong = r'music/swOriginalTheme.mp3'

            pygame.mixer.music.load(slotSong)
            pygame.mixer.music.play(-1)
        else:
            pygame.mixer.music.unpause()

        pauseSlotsMusic = False

        welcomePlayedLast = False
        pokerPlayedLast = False
        slotsPlayedLast = True

    else:

        pygame.mixer.music.pause()

        pauseSlotsMusic = True


def playMusic():
    pygame.mixer.music.play(-1)

def changeBG():
```

```
nonlocal bgCounter

if bgCounter == 0:

    stopMusic()

    slotSong = r'music/cantina.mp3'
    pygame.mixer.music.load(slotSong)
    pygame.mixer.music.play(-1)

    bg_img = Image.open('swpics/moseisleybg1.jpg').resize((800, 500))
    bg_tkimg = itk.PhotoImage(bg_img)
    bgLabel['image'] = bg_tkimg
    bgLabel.image = bg_tkimg

    bgCounter +=1

elif bgCounter == 1:

    stopMusic()

    slotSong = r'music/forcetheme.mp3'
    pygame.mixer.music.load(slotSong)
    pygame.mixer.music.play(-1)

    bg_img = Image.open('swpics/jeditemplebg1.png').resize((800, 500))
    bg_tkimg = itk.PhotoImage(bg_img)
    bgLabel['image'] = bg_tkimg
    bgLabel.image = bg_tkimg

    bgCounter +=1

elif bgCounter == 2:

    stopMusic()

    slotSong = r'music/acrossthestars.mp3'
    pygame.mixer.music.load(slotSong)
    pygame.mixer.music.play(-1)

    bg_img = Image.open('swpics/naboobg1.jpg').resize((800, 500))
    bg_tkimg = itk.PhotoImage(bg_img)
    bgLabel['image'] = bg_tkimg
    bgLabel.image = bg_tkimg

    bgCounter += 1

elif bgCounter == 3:

    stopMusic()
```

```
            slotSong = r'music/impmarch.mp3'
            pygame.mixer.music.load(slotSong)
            pygame.mixer.music.play(-1)

            bg_img = Image.open('swpics/impbg2.jpeg').resize((800, 500))
            bg_tkimg = itk.PhotoImage(bg_img)
            bgLabel['image'] = bg_tkimg
            bgLabel.image = bg_tkimg

            bgCounter += 1

        elif bgCounter == 4:

            stopMusic()

            slotSong = r'music/swfinale.mp3'
            pygame.mixer.music.load(slotSong)
            pygame.mixer.music.play(-1)

            bg_img = Image.open('swpics/finalswbg.jpg').resize((800, 500))
            bg_tkimg = itk.PhotoImage(bg_img)
            bgLabel['image'] = bg_tkimg
            bgLabel.image = bg_tkimg

            bgCounter += 1

        elif bgCounter == 5:

            stopMusic()

            slotSong = r'music/swOriginalTheme.mp3'
            pygame.mixer.music.load(slotSong)
            pygame.mixer.music.play(-1)

            bg_img = Image.open('swpics/starbg4.jpg').resize((800, 500))
            bg_tkimg = itk.PhotoImage(bg_img)
            bgLabel['image'] = bg_tkimg
            bgLabel.image = bg_tkimg

            bgCounter = 0

    def showPayTable():

        tableWindow = Toplevel()
        tableWindow.geometry('730x530')
        tableWindow.resizable(False, False)

        tablebg_img = Image.open('swpics/starbg4.jpg').resize((730, 530))
        tablebg_tkimg = itk.PhotoImage(tablebg_img)
```

```
        tablebgLabel = Label(tableWindow, image=tablebg_tkimg)
        tablebgLabel.image = tablebg_tkimg
        tablebgLabel.grid(row = 0, sticky = 'news')

        table_img = Image.open('swpics/swpaytable.png').resize((600, 435))
        table_tkimg = itk.PhotoImage(table_img)

        tableLabel = Label(tableWindow, image=table_tkimg, borderwidth=3,
relief="ridge", bg="gold")
        tableLabel.image = table_tkimg
        tableLabel.grid(row = 0)


        tableWindow.title('SLOT WARS - Pay Table')


    def scoreRound(ranks, sides, ids):

        nonlocal slotBet
        nonlocal slotBank

        highestRank = max(ranks)
        uniqueSides = len(set(sides))
        uniqueIds = len(set(ids))

        if sides.count('wild') > 0:
            iswild = True
        else:
            iswild = False

        if sides.count('light') > 0 and sides.count('dark') > 0:
            lightanddark = True
        else:
            lightanddark = False



        if uniqueIds == 1 or (uniqueIds == 2 and iswild):

            if ids.count('3') == 3:

                slotBank += slotBet*250

                messagebox.showinfo('MEGA JACKPOT!!!!!', 'You got three R2-D2s,
that\'s a 250x payout!\n')

            elif highestRank > 3:

                slotBank += slotBet*100
```

```
                if sides.count('light') > 0:

                    messagebox.showinfo("LIGHTSIDE JACKPOT!!!!!", 'You got three
Grogus, that\'s a 100x payout!\n')

                else:

                    messagebox.showinfo("DARKSIDE JACKPOT!!!!!", 'You got three
Darth Vaders, that\'s a 100x payout!\n')


            elif highestRank > 2:

                slotBank += slotBet*50

                if sides.count('light') > 0:

                    messagebox.showinfo("LIGHTSIDE TIER 3 WIN!!!", 'You got three
Ahsokas, that\'s a 50x payout!\n')

                else:

                    messagebox.showinfo("DARKSIDE TIER 3 WIN!!!", 'You got three
Darth Mauls, that\'s a 50x payout!\n')


            elif highestRank > 1:

                slotBank += slotBet*20

                if sides.count('light') > 0:

                    messagebox.showinfo("LIGHTSIDE TIER 2 WIN!!", 'You got three
tier 2 lightside characters, that\'s a 20x payout!\n')
                else:

                    messagebox.showinfo("DARKSIDE TIER 2 WIN!!", 'You got three
tier 2 darkside characters, that\'s a 20x payout!\n')


            else:

                slotBank += slotBet*10

                if sides.count('light') > 0:

                    messagebox.showinfo("LIGHTSIDE TIER 1 WIN!", 'You got three
clone troopers, that\'s a 10x payout!\n')
                else:
```

```
                    messagebox.showinfo("DARKSIDE TIER 1 WIN!", 'You got three
storm troopers, that\'s a 10x payout!\n')


        elif not lightanddark:

            if highestRank > 3:

                slotBank += slotBet*5

                if sides.count('light') > 0:

                    messagebox.showinfo("5x LIGHTSIDE FORCE COMBO!!!", 'You got a
5x force combo!\n')
                else:

                    messagebox.showinfo("5x DARKSIDE FORCE COMBO!!!", 'You got a 5x
force combo!\n')


            elif highestRank > 2:

                slotBank += slotBet*3

                if sides.count('light') > 0:

                    messagebox.showinfo("3x LIGHTSIDE FORCE COMBO!!!", 'You got a
3x force combo!\n')
                else:

                    messagebox.showinfo("3x DARKSIDE FORCE COMBO!!!", 'You got a 3x
force combo!\n')


            elif highestRank > 1:

                slotBank += slotBet*1.5

                if sides.count('light') > 0:

                    messagebox.showinfo("1.5x LIGHTSIDE FORCE COMBO!!", 'You got a
1.5x force combo!\n')
                else:

                    messagebox.showinfo("1.5x DARKSIDE FORCE COMBO!!", 'You got a
1.5x force combo!\n')

        slotWalletAmount['text'] = f"{formatter.format(slotBank)}"
```

```
    def spinReels():

        nonlocal slotBank
        nonlocal slotBet

        if slotBank < slotBet:

            messagebox.showerror('Insufficient Funds', 'Please deposit more money
before playing.')

        else:

            slotBank -= slotBet

            slotWalletAmount['text'] = f"{formatter.format(slotBank)}"

            ranks = []
            sides = []
            ids = []

            spin1counter = 0

            while spin1counter < 10:

                spin1 = random.choice(slotRoll1)

                if spin1['side'] == 'light':

                    reel1Label['bg'] = 'cyan'

                elif spin1['side'] == 'dark':

                    reel1Label['bg'] = 'red'

                else:

                    reel1Label['bg'] = 'gold'

                new_reel1_img = itk.PhotoImage(spin1['image'])

                reel1Label['image'] = new_reel1_img
                reel1Label.image = new_reel1_img

                slotWindow.update()

                time.sleep(0.1)

                spin1counter += 1
```

```
ranks.append(spin1['rank'])
sides.append(spin1['side'])
ids.append(spin1['id'])

spin2counter = 0

while spin2counter < 10:

    spin2 = random.choice(slotRoll2)

    if spin2['side'] == 'light':

        reel2Label['bg'] = 'cyan'

    elif spin2['side'] == 'dark':

        reel2Label['bg'] = 'red'

    else:

        reel2Label['bg'] = 'gold'

    new_reel2_img = itk.PhotoImage(spin2['image'])

    reel2Label['image'] = new_reel2_img
    reel2Label.image = new_reel2_img

    slotWindow.update()

    time.sleep(0.1)

    spin2counter += 1

ranks.append(spin2['rank'])
sides.append(spin2['side'])
ids.append(spin2['id'])

spin3counter = 0

while spin3counter < 10:

    spin3 = random.choice(slotRoll3)

    if spin3['side'] == 'light':

        reel3Label['bg'] = 'cyan'

    elif spin3['side'] == 'dark':

        reel3Label['bg'] = 'red'
```

```
            else:

                reel3Label['bg'] = 'gold'

            new_reel3_img = itk.PhotoImage(spin3['image'])

            reel3Label['image'] = new_reel3_img
            reel3Label.image = new_reel3_img

            slotWindow.update()

            time.sleep(0.1)

            spin3counter += 1

        ranks.append(spin3['rank'])
        sides.append(spin3['side'])
        ids.append(spin3['id'])

        scoreRound(ranks, sides, ids)

def placeSlotBet(n):

    nonlocal slotBet

    slotBet = n

    nonlocal slotBank

    slotBank -= slotBet



def slotBet1():

    nonlocal slotBet

    slotBet = 1

    if slotBank >= slotBet:

        slotBet1Button['state'] = 'disabled'

        slotBet10Button['state'] = 'normal'

        slotBet100Button['state'] = 'normal'

        spinButton['state'] = 'normal'
```

```
        else:
            messagebox.showerror('Insufficient Funds', 'Please deposit more money
before betting.')


    def slotBet10():

        nonlocal slotBet

        slotBet = 10

        if slotBank >= slotBet:

            slotBet10Button['state'] = 'disabled'

            slotBet1Button['state'] = 'normal'

            slotBet100Button['state'] = 'normal'

            spinButton['state'] = 'normal'

        else:
            messagebox.showerror('Insufficient Funds', 'Please deposit more money
before betting.')


    def slotBet100():

        nonlocal slotBet

        slotBet = 100

        if slotBank >= slotBet:

            slotBet100Button['state'] = 'disabled'

            slotBet1Button['state'] = 'normal'

            slotBet10Button['state'] = 'normal'

            spinButton['state'] = 'normal'

        else:
            messagebox.showerror('Insufficient Funds', 'Please deposit more money
before betting.')


    def slotDepositFunds():

        amount = slotDepositAmount.get()
```

```
        if not amount.isnumeric():
            messagebox.showerror('Incorrect Format', 'Please enter only numbers for
the deposit amount.')
        else:
            amount = float(amount)
            nonlocal slotBank
            slotBank += amount
            slotWalletAmount['text'] = f"{formatter.format(slotBank)}"


    slotWindow = Toplevel()
    slotWindow.geometry('800x500')
    slotWindow.resizable(False, False)

    bg_img = Image.open('swpics/starbg4.jpg').resize((800, 500))
    bg_tkimg = itk.PhotoImage(bg_img)

    bgLabel = Label(slotWindow, image=bg_tkimg)
    bgLabel.image = bg_tkimg
    bgLabel.grid(row = 0, rowspan = 7, columnspan = 5, sticky = 'news')

    slotFrame = Frame(slotWindow, background='')
    slotFrame.grid(row = 0, padx = 10, pady = 10)

    bgButton = Button(slotWindow, text = 'Change Planet', command = changeBG,
font=('Verdana',10, 'bold'), width = 12)
    bgButton.grid(row = 0, column = 4, sticky = 'ne', padx = 5, pady = 5)

    stopButton = Button(slotWindow, text = 'Pause/Play Music', command = stopMusic,
font=('Verdana',10, 'bold'), width = 15)
    stopButton.grid(row = 0, column = 4, sticky = 'e', padx = 5, pady = 5)

    # playButton = Button(slotWindow, text = 'Play Music', command = playMusic,
font=('Verdana',10, 'bold'), width = 12)
    # playButton.grid(row = 0, column = 4, sticky = 'se', padx = 5, pady = 5)

    tableButton = Button(slotWindow, text = 'Show Pay Table', command =
showPayTable, font=('Verdana',10, 'bold'), width = 12)
    tableButton.grid(row = 2, column = 4, sticky = 'e', padx = 5, pady = 5)

    swLogo_img = Image.open('swpics/SWlogo1.png').resize((615, 65))
    swLogo_tkimg = itk.PhotoImage(swLogo_img)

    swLabel = Label(slotFrame, image = swLogo_tkimg, borderwidth=3, relief="ridge",
bg="gold")
    swLabel.image = swLogo_tkimg
    swLabel.grid(row = 1, column = 1, columnspan = 3, sticky = 's')

    slotFrame2 = Frame(slotWindow, bg="black")
```

```
    slotFrame2.grid(row=3, column = 0, columnspan = 4)

    #r2d2_img = Image.open('slotpics/11.png'))
    r2d2_tkimg = itk.PhotoImage(r2d2_img)

    reel1Label = Label(slotFrame2, image = r2d2_tkimg, bg="gold")
    reel1Label.grid(row = 0, column = 0)
    reel2Label = Label(slotFrame2, image = r2d2_tkimg, bg="gold")
    reel2Label.grid(row = 0, column = 1)
    reel3Label = Label(slotFrame2, image = r2d2_tkimg, bg="gold")
    reel3Label.grid(row = 0, column = 2)

    spinButton = Button(slotFrame2, text = 'TRUST IN \nTHE FORCE',
font=('Verdana',12, 'bold'), command = spinReels)
    spinButton.grid(row = 0, column = 3, padx = 5, pady = 5, sticky = 'news')
    spinButton['state'] = 'disabled'

    slotFrame3 = Frame(slotWindow, bg="black")
    slotFrame3.grid(row=4, column = 0, columnspan = 4, pady=10)

    slotBet1Button = Button(slotFrame3, text = "Bet $1", command = slotBet1,
font=('Verdana',10, 'bold'))
    slotBet1Button.grid(row = 1, column=0)

    slotBet10Button = Button(slotFrame3, text = "Bet $10", command = slotBet10,
font=('Verdana',10, 'bold'))
    slotBet10Button.grid(row = 1, column=1)

    slotBet100Button = Button(slotFrame3, text = "Bet $100", command = slotBet100,
font=('Verdana',10, 'bold'))
    slotBet100Button.grid(row = 1, column=2)

    closeButton = Button(slotWindow, text = "Close Window", command =
closeSlotWindow, font=('Verdana',10, 'bold'))
    closeButton.grid(row=5, pady = 10)


    slotDepositButton = Button(slotWindow, text = "Deposit Funds:",
font=('Verdana',10, 'bold'), command=slotDepositFunds, width = 12)
    slotDepositButton.grid(row = 3, column = 4, sticky='s')
    slotDepositAmount = Entry(slotWindow, font=('Verdana',11, 'bold'),
borderwidth=5, relief="ridge", width = 10, justify='r')
    slotDepositAmount.grid(row = 4, column = 4, pady=5, sticky='n')

    slotWalletLabel = Label(slotWindow, text = "Wallet:", bg='black', fg='gold',
font=('Verdana',12, 'bold'), borderwidth=5, relief="ridge")
    slotWalletLabel.grid(row = 5, column = 4, sticky='s', padx = 20)
    slotWalletAmount = Label(slotWindow, text = f"{formatter.format(slotBank)}",
bg='black', fg='gold', font=('Verdana',11, 'bold'), borderwidth=5, relief="ridge",
width = 10)
```

```
    slotWalletAmount.grid(row = 6, column = 4, ipady=5, pady=5, sticky='n')



    for i in range(7):
        slotWindow.grid_rowconfigure(i, weight=1)

    for i in range(5):
        slotWindow.grid_columnconfigure(i, weight=1)


    slotWindow.title('Boyle Casino - SLOT WARS')

    slotSong = r'music/swOriginalTheme.mp3'

    pygame.init()
    pygame.mixer.init()
    pygame.mixer.music.load(slotSong)
    pygame.mixer.music.play(-1)

    slotWindow.mainloop()


# ### *10 welcome menu gui

# In[23]:


def openBoyleCasino():

    pauseMusic = False
    global welcomePlayedLast
    welcomePlayedLast = True
    global pokerPlayedLast
    pokerPlayedLast = False
    global slotsPlayedLast
    slotsPlayedLast = False

    def closeWelcomeWindow():

        pygame.mixer.music.stop()
        welcomeWindow.destroy()

    def stopMusic():

        nonlocal pauseMusic
        global welcomePlayedLast
        global pokerPlayedLast
        global slotsPlayedLast
```

```
        if pauseMusic:

            if not welcomePlayedLast:
                pygame.mixer.music.load(welcomeSong)
                pygame.mixer.music.play(-1)
            else:
                pygame.mixer.music.unpause()

            pauseMusic = False

            welcomePlayedLast = True
            pokerPlayedLast = False
            slotsPlayedLast = False

        else:

            pygame.mixer.music.pause()

            pauseMusic = True



    def playBlackjack():

        openBJGame()

    def playPoker():

        openVPGame()

    def playSlots():

        openSlots()

    welcomeWindow = Tk()
    welcomeWindow.geometry = ('1100x620')

    welcome_img = Image.open('bcwelcome.png').resize((1100, 620))
    welcome_tkimg = itk.PhotoImage(welcome_img)

    welcomeLabel = Label(welcomeWindow, image=welcome_tkimg)
    welcomeLabel.image = welcome_tkimg
    welcomeLabel.grid(row = 0, rowspan = 7, columnspan = 5, sticky = 'news')

    poker_img = Image.open("pokerlabel2.png").resize((250, 35))
    poker_tkimg = itk.PhotoImage(poker_img)

    pokerButton = Button(welcomeWindow, image=poker_tkimg, justify='center',
command = playPoker)
    pokerButton.image = poker_tkimg
```

```
    pokerButton.grid(row = 4, column = 4)

    bj_img = Image.open("bjlogocustom2.png").resize((200, 50))
    bj_tkimg = itk.PhotoImage(bj_img)

    bjButton = Button(welcomeWindow, image=bj_tkimg, justify='center', command =
playBlackjack)
    bjButton.image = bj_tkimg
    bjButton.grid(row = 4, column = 3, sticky = 'e')

    slots_img = Image.open('swpics/SWlogo2.png').resize((300, 33))
    slots_tkimg = itk.PhotoImage(slots_img)

    slotsButton = Button(welcomeWindow, image=slots_tkimg, justify='center',
command = playSlots)
    slotsButton.image = slots_tkimg
    slotsButton.grid(row = 3, column = 3, columnspan = 2, sticky = 's')

    exitButton = Button(welcomeWindow, text = 'Exit Casino', command =
closeWelcomeWindow, font=('Times New Roman',17, 'bold'))
    exitButton.grid(row = 5, column = 3, columnspan = 2, sticky = 'n')

    stopButton = Button(welcomeWindow, text = 'Pause/Play Music', command =
stopMusic, font=('Times New Roman',10, 'bold'))
    stopButton.grid(row = 6, column = 4, columnspan = 2, sticky = 'se', pady=5,
padx=5)

    welcomeWindow.title('Boyle Family Casino - Welcome Menu')

    welcomeSong = r'music/casino.mp3'

    pygame.init()
    pygame.mixer.init()
    pygame.mixer.music.load(welcomeSong)
    pygame.mixer.music.play(-1)

    welcomeWindow.mainloop()



# ### *11 open program

# In[24]:


openBoyleCasino()
```