



Python Blackjack with Tkinter&Turtle

13006107 Introduction to Computers and Programming

Software Engineering Program

Faculty of Engineering, KMITL

By

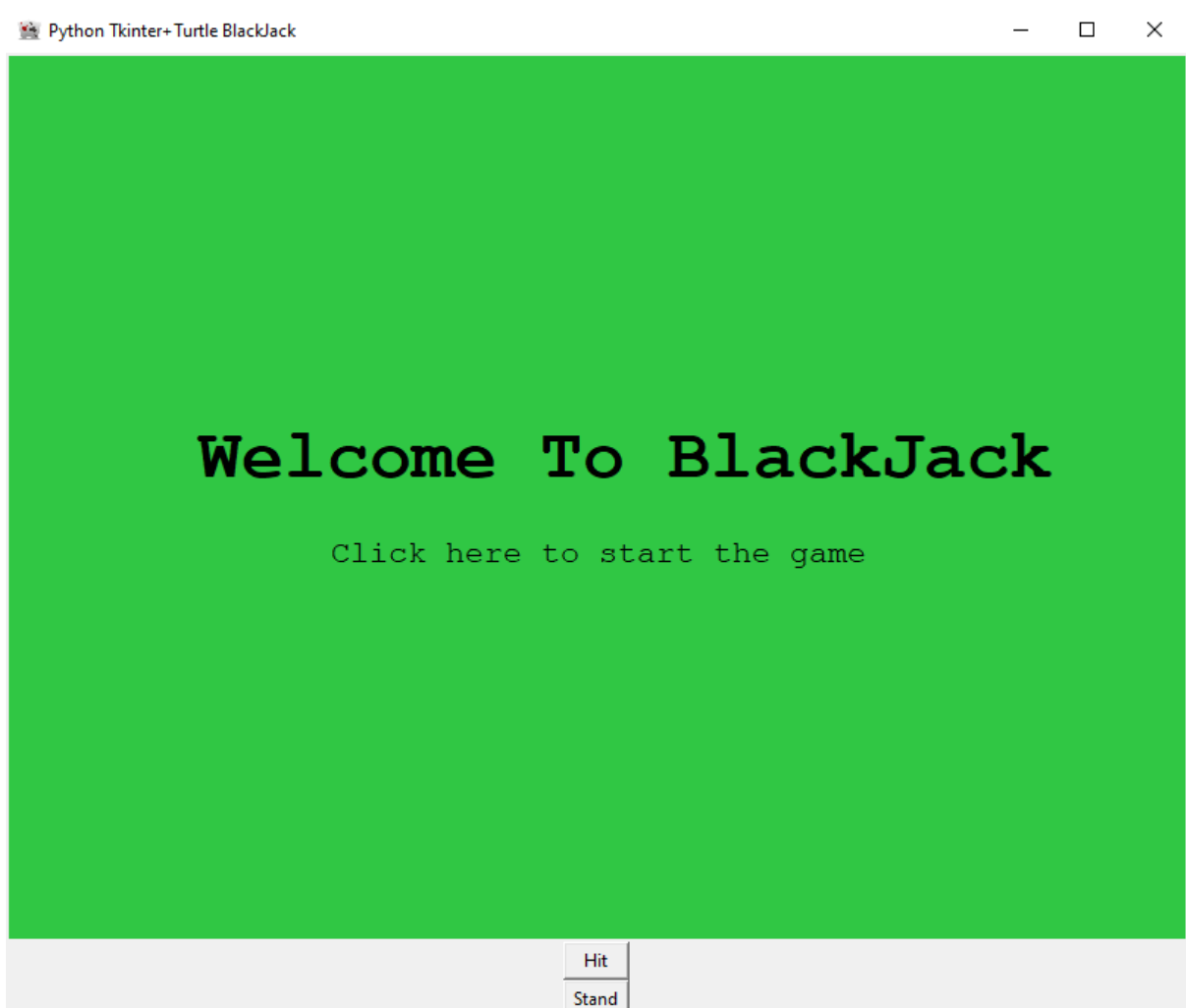
64011548 Phobphoomin Siriboon

Project Description:

This project is a Blackjack game made with Python Turtle and TKinter, additionally I've also added sound effect with the winsound all the animation are put together with turtle and are utilized with TKinter, the turtle screen and raw turtle is casted onto TKinter canvas to observe the output. Classes are also used to create objects in order to organize the program structure and the code efficiency.

The project aims the utilize the concept we've learned in the class and emphasize them as much as I can, most of them are classes and turtle we've been studying since the first day of the lab, I'd like to thank the Teacher in advance for coming up with course and curriculum, considering the fact that it is the sole reason I was able to come up with this project in the first place.

Project Example:





Dealer Score:10
Player Score:15



YOU LOSE!!



Dealer Score:10,20
Player Score:15,19

Source code are separated into two files:

card.py

```
import random
import abc

class Deck:

    def __init__(self):

        self.val = ['A', 2, 3, 4, 5, 6, 7, 8, 9, 10, 'J', 'Q', 'K']
        self.suit = ['club', 'diamond', 'heart', 'spade']
    def drawCard(self):
        valPick = self.val[random.randint(0, len(self.val) - 1)]
        suitPick = self.suit[random.randint(0, len(self.suit) - 1)]
        card = [suitPick, valPick]
        return card

class Hand(Deck):
    def __init__(self):
        super().__init__()
        self.hand = []

    def Hit(self):
        self.hand.append(self.drawCard())

    def initializeHand(self):
        for i in range(2):
            self.Hit()

    def seeHand(self):
        return self.hand

    def evalHand(self):
        handSum = 0

        for i in self.hand:
            if i[1] == 'K' or i[1] == 'Q' or i[1] == 'J':
                handSum += 10
            elif i[1] == 'A':
                if (handSum + 11) > 21:
                    handSum += 1
                else:
                    handSum += 11
            else:
                handSum += 1
```

```

        handSum += i[1]
        # print(i, handSum)

    return handSum

def clearHand(self):
    self.hand = []

def restartHand(self):
    self.clearHand()
    self.initializeHand()

class Player(Hand, abc.ABC):
    def __init__(self):
        super().__init__()

    @abc.abstractmethod
    def initializeHand(self):
        pass

class BlackjackPlayer(Player):
    def __init__(self):
        super().__init__()
        self.initializeHand()
    def initializeHand(self):
        for i in range(2):
            self.Hit()

class Dealer(Player):
    def __init__(self):
        super().__init__()
        self.initializeHand()

    def initializeHand(self):
        self.Hit()
    def dealerPlay(self):
        b = self.evalHand()
        if b < 5:
            dealerChoice = 2
        elif b > 5 and b < 14:
            dealerChoice = 2

        elif b > 14 and b <= 17:
            dealerChoice = random.choice([0,1])
        elif b > 17:
            dealerChoice = 0
        else:
            dealerChoice= random.choice([0,1,2])

    return dealerChoice

```

```

def bustCheck(score):
    if score > 21:
        return True
    if score <= 21:
        return False

def gameCheck(p1,p2):

    if bustCheck(p1) == True:
        print('Busted! You LOSE!')
        return True

    elif bustCheck(p2) == True:
        print('Dealer BUST! You Win!')
        return True
    elif (bustCheck(p1) and bustCheck(p2)) == True:
        print('Still, You LOSE!')
        return True

def blackjack(p1,p2):
    if p1 == 21 and p2 != 21:
        print('You WIN')
        return True
    elif p1 != 21 and p2 == 21:
        print('You LOSE ')
        return True
    elif p1 == 21 and p2 == 21:
        print('DRAW')
        return True
    elif p1 > 21 and p2 > 21 :
        print('Still,You LOSE')
        return True

# def main():
#     p1 = BlackjackPlayer()
#     dealer = Dealer()
#
#     cmd = 0
#     while cmd != 'q':
#         a = p1.evalHand()
#         b = dealer.evalHand()
#
#         print(f"Your Score: {p1.evalHand()}")
#         print(f"Your Hand: {p1.seeHand()}")
#         print(f"Dealer Score: {dealer.evalHand()}")
#         print(f"Dealer Hand: {dealer.seeHand()}")
#
#

```

```

#         if blackjack(a, b) == True:
#             p1.restartHand()
#             dealer.restartHand()
#             continue
#         if gameCheck(a, b) == True:
#             p1.restartHand()
#             dealer.restartHand()
#             continue
#
#         cmd = input("Enter your command: ")
#         if cmd == 'h':
#             p1.Hit()
#
#
#
#
#         elif cmd == 's':
#             dealer.dealerPlay()
#             print(f"Dealer Score: {dealer.evalHand()}")
#             print(f"Dealer Hand: {dealer.seeHand()}")
#
#             if gameCheck(p1.evalHand(), dealer.evalHand()) == True:
#                 pass
#
#             else:
#                 if p1.evalHand() > dealer.evalHand():
#                     print('You WIN')
#                 elif p1.evalHand() == dealer.evalHand():
#                     print('ITS A DRAW!')
#                 else:
#                     print('You LOSE!')
#
#             p1.restartHand()
#             dealer.restartHand()
#
#         elif cmd == 'q':
#             pass
#         else:
#             print('Invalid Input')
#             continue
#

```

main.py

```
from card import *
import time
import turtle
import tkinter as tk
import winsound

class BlackJackScreen:
    def __init__(self):

        self.player = BlackjackPlayer()
        self.dealer = Dealer()
        self.root = tk.Tk()
        self.root.title('Python Tkinter+Turtle BlackJack')
        self.root.iconbitmap(r'card.ico')
        self.canvas = tk.Canvas(self.root)
        self.canvas.config(width=800, height=600)
        self.canvas.pack()
        self.hitCount = -300
        self.dealerCount = -400
        self.updateCount = 90

        button_hit = tk.Button(self.root, text="Hit",
width=5,height=0,command=self.hitButton)
        button_hit.pack()
        button_stand = tk.Button(self.root,
text="Stand",width=5,height=0,command=self.standButton)
        button_stand.pack()

    def displayStart(self, yourHand, dealerHand):
        yourlst = self.displayHand(yourHand)

        dealst = self.displayHand(dealerHand)
        winsound.PlaySound('swipe', winsound.SND_FILENAME)

        self.t.speed(5)
        self.screen.addshape('cardbacktest.gif')
        self.t.shape('cardbacktest.gif')
        self.t.goto(-300,-100)
        self.t.stamp()
        self.t.forward(100)

        previous = 0
        for i in yourlst:
            try:
```



```

        self.t.shape(i)

        self.t.forward(100)
        self.t.stamp()
    except:
        self.screen.addshape(i)
        self.t.shape(i)
        self.t.forward(100)
        self.t.stamp()

self.t.shape('cardbacktest.gif')

self.t.goto(-300,200)

self.t.stamp()
self.t.speed(4)

self.t.forward(100)

for i in dealst:
    try:
        self.t.shape(i)

        self.t.forward(100)
        self.t.stamp()
    except:
        self.screen.addshape(i)
        self.t.shape(i)
        self.t.forward(100)
        self.t.stamp()
self.t.shape('arrow')
self.t.hideturtle()

self.t.speed(0)

self.t.goto(-105,-300)
self.t.write("Player Score:"+str(self.player.evalHand()),
font=("Courier New", 16, "bold"))
self.t.color('red')
self.t.goto(-105, -280)
self.t.write("Dealer Score:" + str(self.dealer.evalHand()),
font=("Courier New", 16, "bold"))

# print(self.player.seeHand())
# print(self.dealer.seeHand())

```

```

def displayHand(self, hand):
    displaylst = []
    for card in hand:
        displaylst.append(card[0]+str(card[1])+'.gif')
    return displaylst
def bustCheck(self, buster):
    if buster > 24:
        return True
    else:
        return False
def drawHit(self, card):
    card = card[0]+str(card[1])+'.gif'
    try:
        self.t.shape(card)
        winsound.PlaySound('swipe', winsound.SND_FILENAME)
        self.t.forward(400)
        self.t.stamp()
    except:
        self.screen.addshape(card)
        self.t.shape(card)
        winsound.PlaySound('swipe', winsound.SND_FILENAME)
        self.t.forward(400)
        self.t.stamp()
def dealerHit(self, card):
    card = card[0]+str(card[1])+'.gif'
    try:
        self.t.shape(card)
        self.t.fd(100)
        self.t.stamp()
    except:
        self.screen.addshape(card)
        self.t.shape(card)
        self.t.fd(100)
        self.t.stamp()

def RestartScreen(self):
    winsound.PlaySound('level', winsound.SND_FILENAME)
    self.hitCount = -300
    self.updateCount = 90
    self.dealerCount = -400
    self.player = BlackjackPlayer()
    self.dealer = Dealer()
    self.screen.clear()
    self.run()

def updateScore(self, robusta):
    if robusta == True:
        self.t.hideturtle()
        self.t.color('red')

```

```

        self.t.goto(self.updateCount, -300)
        self.t.write(str(self.player.evalHand())+" Busted!!" ,
font=("Courier New", 16, "bold"))
        self.t.goto(-160,0)
        self.t.write("YOU LOSE!!" , font=("Courier New", 40, "bold"))
        winsound.PlaySound('lose', winsound.SND_FILENAME)

    else:
        self.t.color('black')

        self.t.hideturtle()
        self.t.penup()
        self.t.goto(self.updateCount, -300)

        self.t.forward(3)
        self.t.write(", " + str(self.player.evalHand()), font=("Courier
New", 16, "bold"))
        self.updateCount += 50
    def updateDealer(self,robusta):
        if robusta == True:
            self.t.hideturtle()
            self.t.color('red')
            self.t.goto(87, -280)
            self.t.write(',') + str(self.dealer.evalHand()) + "Busted",
font=("Courier New", 16, "bold"))
            self.t.goto(-160,0)
            self.t.write("YOU WIN!!" , font=("Courier New", 40, "bold"))
            winsound.PlaySound('win', winsound.SND_FILENAME)
        else:
            self.t.hideturtle()
            self.t.color('red')
            self.t.goto(87, -280)
            self.t.write(',') + str(self.dealer.evalHand()), font=("Courier
New", 16, "bold"))
    def win_lose(self,det):
        if det == 1:
            self.t.goto(-160, 0)
            self.t.write("YOU WIN!!", font=("Courier New", 40, "bold"))
            winsound.PlaySound('win', winsound.SND_FILENAME)
        elif det == 2:
            self.t.goto(-140, 0)
            self.t.write("DRAW!!", font=("Courier New", 40, "bold"))
            winsound.PlaySound('win', winsound.SND_FILENAME)
        else:
            self.t.goto(-160, 0)
            self.t.write("YOU LOSE!!", font=("Courier New", 40, "bold"))
            winsound.PlaySound('lose', winsound.SND_FILENAME)

    def determine(self):
        a = self.player.evalHand()
        b = self.dealer.evalHand()

```

```

        if a > b:
            self.win_lose(1)
        elif b > a:
            self.win_lose(0)
        elif a == b:
            self.win_lose(2)

    def initializeScreen(self):
        self.screen = turtle.TurtleScreen(self.canvas)

        self.t = turtle.RawTurtle(self.screen)
        self.screen.bgcolor('#30c744')
        self.t.penup()
    def hitButton(self):

        self.player.Hit()
        ourHand = self.player.seeHand()
        self.t.hideturtle()
        self.t.speed(0)
        self.t.showturtle()
        self.t.goto(self.hitCount, -100)
        self.hitCount+=100
        self.t.speed(7)

        self.drawHit(ourHand[-1])
        self.updateScore(bustCheck(self.player.evalHand()))

        if bustCheck(self.player.evalHand()) == True:
            time.sleep(1)
            self.RestartScreen()
    def standButton(self):

        dealerChoice = self.dealer.dealerPlay()
        print(dealerChoice)
        self.t.showturtle()

        self.t.goto(self.dealerCount, 200)
        self.dealerCount+=10
        self.t.speed(7)

        for i in range(dealerChoice):
            self.dealer.Hit()
        dealerHand = self.dealer.seeHand()
        winsound.PlaySound('swipe', winsound.SND_FILENAME)
        for num in range(dealerChoice, 0, -1):
            self.dealerHit(dealerHand[num])
        self.updateDealer(bustCheck(self.dealer.evalHand()))
        print(self.dealer.seeHand())
        print(self.dealer.evalHand())
        if bustCheck(self.dealer.evalHand()) == True:

```

```

        time.sleep(1)
        self.RestartScreen()
    else:
        self.determine()
        time.sleep(1)
        self.RestartScreen()
def finalize(self):
    self.root.mainloop()
def run(self):
    self.initializeScreen()

    self.displayStart(self.player.seeHand(), self.dealer.seeHand())

    self.finalize()
def mainmenu(self):
    self.screen = turtle.TurtleScreen(self.canvas)

    self.t = turtle.RawTurtle(self.screen)
    self.screen.bgcolor('#30c744')
    self.t.penup()

    self.t.goto(-270,0)
    self.t.hideturtle()
    self.t.write("Welcome To BlackJack", font=("Courier New", 36,
"bold"))
    self.t.goto(-180, -50)
    self.t.write("Click here to start the game", font=("Courier New",
16, "normal"))
    def startgame(x,y):
        self.t.goto(-270,0)

        for i in range(12):

            self.t.color('#30c744')

            self.t.write("Welcome To BlackJack", font=("Courier New",
36, "bold"))
            self.t.color('white')
            self.t.write("Welcome To BlackJack", font=("Courier New",
36, "bold"))

            self.t.color('black')
            self.t.write("Welcome To BlackJack", font=("Courier New",
36, "bold"))
        winsound.PlaySound('win', winsound.SND_FILENAME)

    self.run()
    self.screen.onclick(startgame)

```

```
self.finalize()
```

```
t1= BlackJackScreen()  
t1.mainmenu()
```