import random

import pygame

import os

import time

from tkinter import \*

from tkinter import messagebox as ms

import sqlite3

from sys import exit

pygame.mixer.pre\_init(44100,16,2,4096)

pygame.init() #initialises pygame

skipturn = False

reverseturn = False

Player1wins = False

Computerwins = False

global down

x = 800

y = 600

div\_iwidth = 750

div\_iheight = 550

white = (255,255,255)

black = (0,0,0)

green = (0,255,0)

red = (255,0,0)

blue = (0,0,255)

orange = (255,165,0)

mouseposition = pygame.mouse.get\_pos()

class Card:

    def \_\_init\_\_(self, suit, number):

        self.suit = suit

        vals = {10: "reverse", 11:"skip", 12:"+2"}

        if type(number) is int:

            if number < 10:

                self.number = str(number)

            else:

                self.number = vals[number]

        self.image = pygame.image.load(str(self) + ".png")

    def \_\_eq\_\_(self, other):

        if self.suit == other.suit and self.number == other.number:

            return True

        return False

    def \_\_repr\_\_(self):

        return self.showcard()

    def getImage():

        return self.image

    def showcard(self):

        return "{} {}".format(self.suit, self.number)

class Deck:

    def \_\_init\_\_(self):

        self.cards = []

        self.builddeck()

    def builddeck(self):

        for s in ["yellow","red","blue","green"]:

            for n in range(0,13):

                self.cards.append(Card(s, n))

    def showdeck(self):

        for card in self.cards:

            print(card)

    def shuffle(self):

        random.shuffle(self.cards)

    def drawcard(self):

        return self.cards.pop()

class Player:

    def \_\_init\_\_(self, name):

        self.hand = []

        self.name = name

    def draw(self, deck, x):

        for i in range(x):

            self.hand.append(deck.drawcard())

            #allows player draw multiple cards

    def showhand(self):

      print("{} Hand is: ".format(self.name))

      h=30

      for card in self.hand:

          #print(card)

          displayimage(card.image,div\_iwidth,div\_iheight-h)

          pygame.display.update()

          h=h+25

      #if len(self.hand) == 1:

      #   global UNO\_called

      #   while UNO\_called == False:

      #      gamtext\_display("Press U to call UNO", 2,4,15)

       #     print("call uno thing")

       #     #pygame.display.update()

       #     if UNO\_called == True:

       #         gametext\_display("UNO CALLED", 2,5, 15)

       #         break

       #  pygame.display.update()   UNO CALLED FUNCTION NOT WORKING, WILL FIX

      if len(self.hand) == 0:

        Player1wins = True

        add\_screen()

        gametext\_display("Player1 won. Game Over", 2, 2, 40)

        pygame.mixer.music.load("winnermusic.mp3")

        pygame.mixer.music.play(-1)

        pygame.display.update()

        time.sleep(5)

        pygame.quit()

        os.\_exit(1)

    def discard(self):

      global down

      if len(self.hand) != 0:

        discard\_card = self.hand[down]

        self.throwAway(discard\_card)

        if discard\_card.number == "+2":

            gametext\_display("Computer Draws 2 more cards", 2, 4,15)

            pygame.display.update()

            Computer.draw(deck, 2)

        elif discard\_card.number == "skip":

            global skipturn

            skipturn = True

            gametext\_display("Computer's Turn Will Be Skipped Next Round", 2, 4,15)

            pygame.display.update()

        elif discard\_card.number == "reverse":

            global reverseturn

            reverseturn = True

            gametext\_display("Computer's Turn Will Be Reversed Next Round", 2,4,15)

            pygame.display.update()

    def throwAway(self, discard):

       for card in self.hand:

          if card == discard:

              self.hand.remove(card)

              maingamepile.append(card)

class AI(Player):

    def discard(self):

        lastcardplaced = maingamepile[-1]

        for i in range (len(self.hand)):

            ai\_card = self.hand[i]

            if ai\_card.suit == lastcardplaced.suit or ai\_card.number == lastcardplaced.number: #checks if card the computer wants to discard is the same suit or number as the card the player first discarded

                print("computer discarded", ai\_card)

                self.throwAway(ai\_card)

                if ai\_card.number == "+2":

                    gametext\_display("Player1 Draws 2 more cards", 2, 5,15)

                    pygame.display.update()

                    Player1.draw(deck, 2)

                elif ai\_card.number == "skip":

                    gametext\_display("Player1's Turn Skipped", 2, 5,15)

                    pygame.display.update()

                    Computer.discard()

                elif ai\_card.number == "reverse":

                    gametext\_display("Player1's Turn Reversed", 2,5,15)

                    pygame.display.update()

                    Computer.discard()

                break

            else:

                gametext\_display("Computer draws a card", 2, 5, 15)

                Computer.draw(deck, 1)

                break

    def showhand(self):

        print("{}'s Hand is: ".format(self.name))

        h=30

        for card in self.hand:

            print(card)

            displayimage(deckImg,div\_iwidth-600,div\_iheight-h)

            pygame.display.update()

            h=h+25

        if len(self.hand) == 0:

            Computerwins = True

            add\_screen()

            gametext\_display("Computer won. Game Over", 2, 2, 40)

            pygame.display.update()

            time.sleep(5)

            pygame.quit()

            os.\_exit(1)

deck = Deck()

maingamepile = []

Player1 = Player('Player1')

Computer = AI('Computer')

#GUI + EXTRA FUNCTION STUFF

def displayimage(image\_name,div\_iwidth, div\_iheight):

    iwidth = x-div\_iwidth

    iheight = y-div\_iheight

    uno\_window.blit(image\_name, (iwidth,iheight))

    time.sleep(0.3)

def text\_objects(text, font): #this function takes the rectangle and puts it over the whole of the text so it can be moved as one

    textSurface = font.render(text, True, black)

    return textSurface, textSurface.get\_rect()

def gametext\_display(text,divby\_x,divby\_y,fontsize):

    gametext = pygame.font.Font('freesansbold.ttf', fontsize)

    TextSurf,TextRect = text\_objects(text, gametext)

    TextRect.center = ((x/divby\_x) ,(y/divby\_y))

    uno\_window.blit(TextSurf, TextRect)

def deal\_deck\_selected():

        deck.shuffle()

        Player1.draw(deck, 3)

        Player1.showhand()

        Computer.draw(deck, 3)

        Computer.showhand()

        gametext\_display('Player1 starts first, use the number keys to select a card',2,12,15)

def display\_last\_discarded():

    lastcardplaced = maingamepile[-1]

    displayimage(lastcardplaced.image, div\_iwidth-300, div\_iheight-150)

    gametext\_display("Last placed card is:", 2, 3.5, 15)

    pygame.display.update()

def discard\_card\_selected():

    global skipturn

    global reverseturn

    if Player1wins == False and Computerwins == False and skipturn == False and reverseturn == False:

        Player1.discard()

        Computer.discard()

        empty\_singleplayer\_screen()

        Player1.showhand()

        Computer.showhand()

        display\_last\_discarded()

        time.sleep(4)

    elif skipturn == True:

        Player1.discard()

        skipturn = False

    elif reverseturn == True:

        Player1.discard()

        reverseturn = False

def deck\_image(width,height):

    global deckImg

    deckImg = pygame.image.load('deck\_image.png')

    uno\_window.blit(deckImg, (width,height))

def add\_screen():

    global uno\_window

    uno\_window = pygame.display.set\_mode((x, y)) #creates a window with specified resolution (x,y)

    uno\_window.fill(white)

    pygame.display.set\_caption('Python UNO') #sets window title

    fps = pygame.time.Clock() #creates a clock that counts fps

    fps.tick(20)

    backgroundImg = pygame.image.load('background\_image.png')

    displayimage(backgroundImg, x, y)

    pygame.display.update()

def createbutton(button\_name,x1,y2,w1,h2,inactive\_colour,active\_colour,action=None):

    mouse = pygame.mouse.get\_pos()

    click = pygame.mouse.get\_pressed()

    if x1+w1 > mouse[0] > x1 and y2+h2 > mouse[1] > y2:

        pygame.draw.rect(uno\_window, active\_colour,(x1,y2,w1,h2))

        if click[0] == 1 and action != None:

            action()

    else:

        pygame.draw.rect(uno\_window, inactive\_colour,(x1,y2,w1,h2))

    button\_font\_size = pygame.font.Font("freesansbold.ttf",20)

    textSurf, textRect = text\_objects(button\_name, button\_font\_size)

    textRect.center = ( (x1+(w1/2)), (y2+(h2/2)) )

    uno\_window.blit(textSurf, textRect)

def startup\_menu():

    add\_screen()

    pygame.mixer.music.load("Menumusic.mp3")

    pygame.mixer.music.play(-1)

    startup = True

    while startup:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                quit()

        gametext\_display('Welcome to UNO',2,2,50)

        singeplayer\_button = createbutton('SINGLEPLAYER',50,450,160,50,green,orange,singleplayer)

        mutliplayer\_button = createbutton('MULTILPLAYER',250,450,160,50,blue,orange,multiplayer)

        quit\_button = createbutton('QUIT',450,450,100,50,red,orange,quitgame)

        help\_button = createbutton('HELP', 600,450,100, 50, white, orange,help\_screen)

        pygame.display.update()

def quitgame():

    pygame.quit()

    quit()

def help\_screen():

    add\_screen()

    startup = True

    while startup:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                quit()

        back\_button = createbutton('BACK',50,450,100,40,white, orange,startup\_menu)

        rulescreen\_button = createbutton('RULES',500,450,100,40,white,orange,rulescreen)

        gametext\_display('KEYBINDS                             ACTIONS',2,12,25)

        gametext\_display('Tab                                       Deal Cards', 2,6,18)

        gametext\_display('x                     Discard Card x in Hand', 2,4,18)

        gametext\_display('Space                                    Draw a card',2,3,18)

        gametext\_display('s                           Show last played card',2,2.3,18)

        pygame.display.update()

def rulescreen():

    add\_screen()

    startup = True

    while startup:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                quit()

        back\_button = createbutton('BACK',50,450,100,40,white, orange,help\_screen)

        gametext\_display("Rules: The rules of uno are simple.", 3.5,12,20)

        gametext\_display("Your aim is to get rid of your cards before your opponent.", 2.3,9,20)

        gametext\_display("Start the game by pressing the TAB button",3.3,7,20)

        gametext\_display("Press the numbers keys to discard a card, e.g. 3 will discard your third card", 2,6,20)

        gametext\_display("If you can't match the colour or number of the card on the pile, press SPACE",2,5,20)

        gametext\_display("to draw another card",4.5,4.5,20)

        gametext\_display("Action Cards: Reverse- Reverses the turn order",2.5,3.5,20)

        gametext\_display("Skip - skips the opponents turn", 3.5,3,20)

        gametext\_display("+2- your opponent gets two extra cards", 2.5,2.5,20)

        pygame.display.update()

def singleplayer():

    pygame.mixer.music.load("Menumusic\_2.mp3")

    pygame.mixer.music.play(-1)

    add\_screen()

    play = True

    while play == True:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                quit()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_TAB:

                deal\_deck\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_1:

                global down

                down = 0

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_2:

                down = 1

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_3:

                down = 2

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_4:

                down = 3

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_5:

                down = 4

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_6:

                down = 5

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_7:

                down = 6

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_8:

                down = 7

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_9:

                down = 8

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_0:

                down = 9

                discard\_card\_selected()

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_s:

                display\_last\_discarded()

                time.sleep(4)

            elif event.type == pygame.KEYDOWN and event.key == pygame.K\_SPACE:

                Player1.draw(deck, 1)

                Player1.showhand()

                gametext\_display("You've drawn a card from the pile",2,5,15)

            '''elif event.type == pygame.KEYDOWN and event.key == pygame.K\_u:

                global Uno\_called

                UNO\_called = True'''

        gametext\_display('Player1', 12, 12, 15)

        gametext\_display('Computer', 1.2, 12, 15)

        back\_button = createbutton('BACK',300,500,200,40,white, orange,startup\_menu)

        width = (x/2.3) #location on screen

        height = (y/3) #location on screen

        deck\_image(width,height)

        pygame.display.update()

def empty\_singleplayer\_screen():

    add\_screen()

    #display\_last\_discarded()

    gametext\_display('Player1', 12, 12, 15)

    gametext\_display('Computer', 1.2, 12, 15)

    width = (x/2.3) #location on screen

    height = (y/3) #location on screen

    deck\_image(width,height)

def multiplayer():

    add\_screen()

    startup = True

    while startup:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                quit()

        back\_button = createbutton('BACK',50,450,100,40,white, orange,startup\_menu)

        #multiplayer\_local\_button = createbutton('Multiplayer Local',250,450,100,50,blue,orange,multiplayer\_local)

        multiplayer\_online\_button = createbutton('Multiplayer Online',375,450,250,50,blue,orange,multiplayer\_online)

        pygame.display.update()

def multiplayer\_online():

    pass

def uno\_gui():

    if login == True:

        startup\_menu()

    else:

        print("User not logged in")

# make database and users (if not exists already) table at programme start up

with sqlite3.connect('uno\_user\_database.db') as db:

    c = db.cursor()

c.execute('CREATE TABLE IF NOT EXISTS users (user\_id INTEGER PRIMARY KEY, username TEXT NOT NULL ,password TEX NOT NULL);')

db.commit()

db.close()

#login class for uno

class Unologin:

    def \_\_init\_\_(self,master):

        # Window

        self.master = master

        # Some Usefull variables

        self.username = StringVar()

        self.password = StringVar()

        self.n\_username = StringVar()

        self.n\_password = StringVar()

        #Create Widgets

        self.widgets()

    #Login Function

    def login(self):

        #Establish Connection

        with sqlite3.connect('uno\_user\_database.db') as db:

            c = db.cursor()

        #Find user If there is any take proper action

        find\_user = ('SELECT \* FROM users WHERE username = ? and password = ?')

        c.execute(find\_user,[(self.username.get()),(self.password.get())])

        result = c.fetchall()

        if result:

            self.logf.pack\_forget()

            self.head['text'] = self.username.get() + '\n Logged In'

            self.head['pady'] = 100

            self.head['padx'] = 100

            global login

            login = True

            uno\_gui()

        else:

            ms.showerror('Username Not Found.')

    def new\_user(self):

        #Establish Connection

        with sqlite3.connect('uno\_user\_database.db') as db:

            c = db.cursor()

        #Find Existing username if any take proper action

        find\_user = ('SELECT \* FROM users WHERE username = ?')

        c.execute(find\_user,[(self.username.get())])

        if c.fetchall():

            ms.showerror('Error!','Username Taken Try a Diffrent One.')

        else:

            ms.showinfo('Success!')

            self.log()

        #Create New Account

        insert = 'INSERT INTO users(username,password) VALUES(?,?)'

        c.execute(insert,[(self.n\_username.get()),(self.n\_password.get())])

        db.commit()

        #Frame Packing Methords

    def log(self):

        self.username.set('')

        self.password.set('')

        self.crf.pack\_forget()

        self.head['text'] = 'LOGIN'

        self.logf.pack()

    def cr(self):

        self.n\_username.set('')

        self.n\_password.set('')

        self.logf.pack\_forget()

        self.head['text'] = 'Create Account'

        self.crf.pack()

    #Draw Widgets

    def widgets(self):

        self.head = Label(self.master,text = 'LOGIN',font = ('',35),pady = 10)

        self.head.pack()

        self.logf = Frame(self.master,padx =10,pady = 10)

        Label(self.logf,text = 'Username: ',font = ('',20),pady=5,padx=5).grid(sticky = W)

        Entry(self.logf,textvariable = self.username,bd = 5,font = ('',15)).grid(row=0,column=1)

        Label(self.logf,text = 'Password: ',font = ('',20),pady=5,padx=5).grid(sticky = W)

        Entry(self.logf,textvariable = self.password,bd = 5,font = ('',15),show = '\*').grid(row=1,column=1)

        Button(self.logf,text = ' Login ',bd = 3 ,font = ('',15),padx=5,pady=5,command=self.login).grid()

        Button(self.logf,text = ' Create Account ',bd = 3 ,font = ('',15),padx=5,pady=5,command=self.cr).grid(row=2,column=1)

        self.logf.pack()

        self.crf = Frame(self.master,padx =10,pady = 10)

        Label(self.crf,text = 'Username: ',font = ('',20),pady=5,padx=5).grid(sticky = W)

        Entry(self.crf,textvariable = self.n\_username,bd = 5,font = ('',15)).grid(row=0,column=1)

        Label(self.crf,text = 'Password: ',font = ('',20),pady=5,padx=5).grid(sticky = W)

        Entry(self.crf,textvariable = self.n\_password,bd = 5,font = ('',15),show = '\*').grid(row=1,column=1)

        Button(self.crf,text = 'Create Account',bd = 3 ,font = ('',15),padx=5,pady=5,command=self.new\_user).grid()

        Button(self.crf,text = 'Go to Login',bd = 3 ,font = ('',15),padx=5,pady=5,command=self.log).grid(row=2,column=1)

#create window and application object

root = Tk()

root.title("Login Form")

Unologin(root)

root.mainloop()