**库文件: GetDeck.py 生成牌，洗牌，自动发牌，牌排序**

import random

class Deck:

#std\_number:初始手牌数 member:玩家数

def newDeck(self, init\_number=17, member=3, ):#得到一副扑克牌

ranks=['3','4','5','6','7','8','9','A10','J','Q','QK','YA','Z2']

suits = ['♠', '♥', '♦', '♣']

deck = []

for s in suits:

for r in ranks:

deck.append(s + r)

deck.append('Big\nJoker')

deck.append('Small\nJoker')

## 得到属性

self.deck = deck# 一套顺序牌

self.init\_number = init\_number# 初始手牌数

self.member = member# 玩家数

random.shuffle(self.deck) ## 洗牌

## 排序后替换函数

def g(self, x):

x = x.replace('A10', '10')

x = x.replace('QK','K')

x = x.replace('YA','A')

x = x.replace('Z2','2')

return x

## 确定对应座位上的玩家的身份

def confirm\_roles(self,member):

roles = {}

beginner = random.randint(0, member-1)

## ...此处待完善

roles['landlord'] = 0 #0对应座位序号

roles['farmer1'] = 1 #1对应座位序号

roles['farmer2'] = 2 #2对应座位序号

return roles

## 从牌堆取牌返回这些牌

def Deal\_hand(self):

## hand列表记录当前手牌

hand = []

## 从牌堆发init\_number张牌，牌堆减少init\_number张牌

hand = self.deck[:self.init\_number]

del self.deck[:self.init\_number]

## 返回摸的牌

return hand

def Deal(self):

self.newDeck()

roles = self.confirm\_roles(self.member)

hands = []

for i in range(self.member):

hands.append(self.Deal\_hand())

hands[roles['landlord']] += self.deck

## 排序

for hand in hands:

for i in range(1,len(hand)):

for j in range(1,len(hand)-i+1):

if hand[j][1:] < hand[j-1][1:]:

hand[j],hand[j-1] = hand[j-1],hand[j]

## 'A10','QK','YA','Z2'替换回来

for i in range(len(hands)):

hands[i] = list(map(self.g, hands[i]))

return hands

**play.py 调用GetDeck.py库，生成图形化界面：**

import tkinter as tk

import GetDeck as gd

game = tk.Tk()

game.title('欢迎来到斗地主')

OneDeck = gd.Deck()

hands = OneDeck.Deal()

out\_list0 = []

out\_list1 = []

out\_list2 = []

hand\_cards0=[]

hand\_cards1=[]

hand\_cards2=[]

def uplift0(i):

out\_list0.append(i)

def outs0():

for i in out\_list0:

hand\_cards0[i].grid(row=1)

def uplift1(i):

out\_list1.append(i)

def outs1():

for i in out\_list1:

hand\_cards1[i].grid(row=4)

def uplift2(i):

out\_list2.append(i)

def outs2():

for i in out\_list2:

hand\_cards2[i].grid(row=7)

def shuffle():

hands = OneDeck.Deal()

start()

def start():

START.destroy()

hand0 = hands[0]

tk.Label(game, text='landlord\'s cards').grid(row=0,column = 0)

for i in range(len(hand0)):

code = """

hand\_cards0.append(tk.Button(game, text=hand0[i], command=lambda : uplift0(%d),width=5, height=5, activebackground='yellow', relief='groove',anchor='n'))

hand\_cards0[i].grid(row=2, column=i+1, sticky='s', padx=2)

"""% i

exec(code)

if hand0[i][0] == '♥':

hand\_cards0[i]['fg'] ='red'

if hand0[i][0] == '♦':

hand\_cards0[i]['fg'] ='red'

Out = tk.Button(game, text='Outs', command=outs0, width=5, height=5)

Out.grid(row=2, column=len(hand0)+1, sticky='s',padx=5)

hand1 = hands[1]

tk.Label(game, text='farmer1\'s cards').grid(row=3, column = 0)

for i in range(len(hand1)):

code = """

hand\_cards1.append(tk.Button(game, text=hand1[i], command=lambda : uplift1(%d),width=5, height=5, activebackground='yellow', relief='groove',anchor='n'))

hand\_cards1[i].grid(row=5, column=i+1, sticky='s', padx=2)

"""% i

exec(code)

if hand1[i][0] == '♥':

hand\_cards1[i]['fg'] ='red'

if hand1[i][0] == '♦':

hand\_cards1[i]['fg'] ='red'

Out = tk.Button(game, text='Outs', command=outs1, width=5, height=5)

Out.grid(row=5, column=len(hand1)+1, sticky='s',padx=5)

hand2 = hands[2]

tk.Label(game, text='farmer2\'s cards').grid(row=6, column = 0)

for i in range(len(hand2)):

code = """

hand\_cards2.append(tk.Button(game, text=hand2[i], command=lambda : uplift2(%d),width=5, height=5, activebackground='yellow', relief='groove',anchor='n'))

hand\_cards2[i].grid(row=8, column=i+1, sticky='s', padx=2)

"""% i

exec(code)

if hand2[i][0] == '♥':

hand\_cards2[i]['fg'] ='red'

if hand2[i][0] == '♦':

hand\_cards2[i]['fg'] ='red'

Out = tk.Button(game, text='Outs', command=outs2, width=5, height=5)

Out.grid(row=8, column=len(hand2)+1, sticky='s',padx=5)

START = tk.Button(game, text='开始', command=start,\

width=10, anchor='n', bg='RoyalBlue', font = ('微软雅黑'))

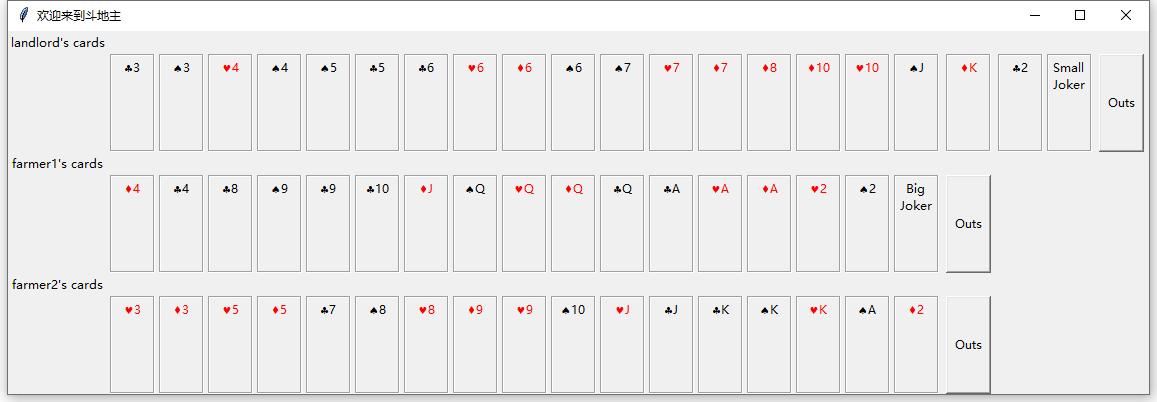
START.grid(row=2,column=2,sticky='s', padx=500)

**运行play.py截图：**

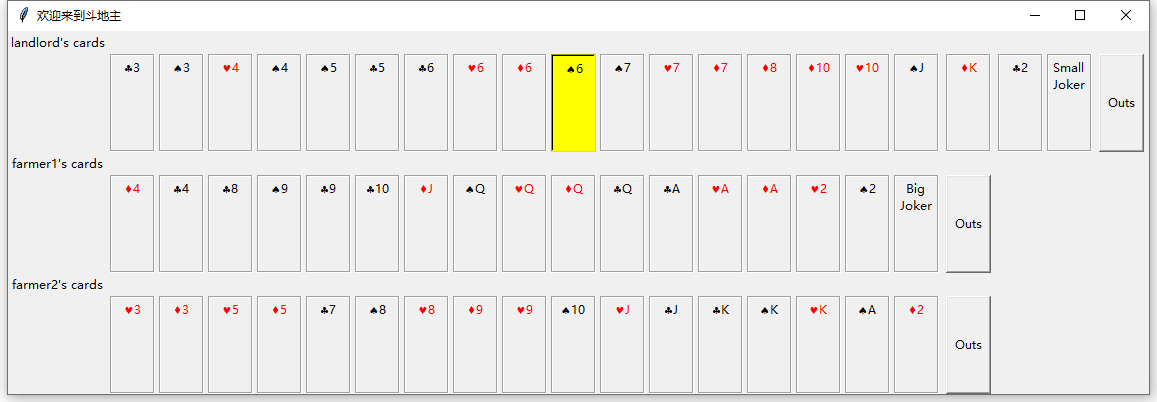
开始界面：



牌桌：



单击黑桃6



单击Outs

