from random import \*

from tkinter import \* #导入图形化用户界面模块

step\_number = 0 #设置步数的变量，初始值为0

difficulty = int(input('请输入数字华容道列数（3/4/5）：'))

def Button\_Click\_1(x,y): #按钮点击事件函数

"""声明空白按钮行列号和步数的变量为全局变量"""

global row\_of\_space

global col\_of\_space

global step\_number

"""判断判断点击按钮旁是否为空白按钮，是则交换位置"""

if abs(x-row\_of\_space) + abs(y-col\_of\_space) == 1:

step\_number += 1 #将步数赋值

label\_step\_number['text'] = '步数:' + str(step\_number) #将步数变量导入label控件

"""交换按钮位置"""

buttons[row\_of\_space,col\_of\_space]['text'] = buttons[x,y]['text']

buttons[x,y]['text'] = ' '

row\_of\_space = x

col\_of\_space = y

n = 0

for row in range(3):

for col in range(3):

"""对比所有按钮序列是否正确，不正确则跳出函数"""

if buttons[row,col]['text'] != numbers[n]:

return

n += 1

"""所有按钮判断完毕赢得游戏胜利"""

label\_welcomes['text'] = '你赢了'

def Button\_Click\_2(x,y): #按钮点击事件函数

"""声明空白按钮行列号和步数的变量为全局变量"""

global row\_of\_space

global col\_of\_space

global step\_number

"""判断判断点击按钮旁是否为空白按钮，是则交换位置"""

if abs(x-row\_of\_space) + abs(y-col\_of\_space) == 1:

step\_number += 1 #将步数赋值

label\_step\_number['text'] = '步数:' + str(step\_number) #将步数变量导入label控件

"""交换按钮位置"""

buttons[row\_of\_space,col\_of\_space]['text'] = buttons[x,y]['text']

buttons[x,y]['text'] = ' '

row\_of\_space = x

col\_of\_space = y

n = 0

for row in range(4):

for col in range(4):

"""对比所有按钮序列是否正确，不正确则跳出函数"""

if buttons[row,col]['text'] != numbers[n]:

return

n += 1

"""所有按钮判断完毕赢得游戏胜利"""

label\_welcomes['text'] = '你赢了'

def Button\_Click\_3(x,y): #按钮点击事件函数

"""声明空白按钮行列号和步数的变量为全局变量"""

global row\_of\_space

global col\_of\_space

global step\_number

"""判断判断点击按钮旁是否为空白按钮，是则交换位置"""

if abs(x-row\_of\_space) + abs(y-col\_of\_space) == 1:

step\_number += 1 #将步数赋值

label\_step\_number['text'] = '步数:' + str(step\_number) #将步数变量导入label控件

"""交换按钮位置"""

buttons[row\_of\_space,col\_of\_space]['text'] = buttons[x,y]['text']

buttons[x,y]['text'] = ' '

row\_of\_space = x

col\_of\_space = y

n = 0

for row in range(5):

for col in range(5):

"""对比所有按钮序列是否正确，不正确则跳出函数"""

if buttons[row,col]['text'] != numbers[n]:

return

n += 1

"""所有按钮判断完毕赢得游戏胜利"""

label\_welcomes['text'] = '你赢了'

"""创建华容道游戏窗口"""

root = Tk() #创建图形化用户界面实例

root.title('数字华容道') #设置窗口标题

root.geometry("400x400") #将窗口大小设为高400宽400

root.configure(bg = 'black') #将窗口背景设为黑色

root.resizable(width = False,height = False) #设置窗口为不可拉伸

"""设置欢迎语的label控件"""

label\_welcomes = Label(root,text = '欢迎来到数字华容道',bg = 'black',fg = 'red',font = ("Arial",13)) #设置label控件的属性

label\_welcomes.place(x = 20,y = 10,width = 250,height = 40) #设置label控件位置

"""设置显示操作方式的label控件"""

label\_operation = Label(root,text = '单击数字',bg = 'black',fg = 'white',font = ("Arial",10))

label\_operation.place(x = 3,y = 40,width = 50,height = 30)

label\_operation\_2 = Label(root,text = '移动方块',bg = 'black',fg = 'white',font = ("Arial",10))

label\_operation\_2.place(x = 3,y = 60,width = 50,height = 30)

"""设置显示步数的label控件"""

label\_step\_number = Label(root,text = '步数:' + str(step\_number),bg = 'black',fg = 'yellow',font = ("Arial",10))

label\_step\_number.place(x = 3,y = 20,width = 50,height = 30)

if difficulty == 3:

root.attributes("-topmost", True)

row\_of\_space = 0 #存放空白按钮的行号

col\_of\_space = 0 #存放空白按钮的列号

buttons = {} #存放数字按钮的数组

numbers = ['1','2','3','4','5','6','7','8',' '] #所有数字文本的列表

shuffle(numbers) #打乱数字列表中的数字顺序

"""制造数字华容道方阵"""

for row in range(3):

for col in range(3):

"""创建数字按钮，并将行列号传入该按钮的点击事件函数"""

button = Button(root,command = lambda x = row,y = col:Button\_Click\_1(x,y),bg = 'black',fg = 'green',font = ("Arial",35))

buttons[row,col] = button #将按钮导入数组

button['text'] = numbers.pop() #设置按钮上的文本

button.place(x = 60 + col \* 60,y = 60 + row \* 60,width = 50,height = 50) #设置数字按钮大小

if button['text'] == ' ': #判断是否为空白按钮，如果是，则记录到空白按钮行列号变量

row\_of\_space = row

col\_of\_space = col

numbers = ['1','2','3','4','5','6','7','8',' '] #还原数字列表

root.mainloop() #显示窗口

elif difficulty == 4:

root.attributes("-topmost", True)

row\_of\_space = 0 #存放空白按钮的行号

col\_of\_space = 0 #存放空白按钮的列号

buttons = {} #存放数字按钮的数组

numbers = ['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15',' '] #所有数字文本的列表

shuffle(numbers) #打乱数字列表中的数字顺序

"""制造数字华容道方阵"""

for row in range(4):

for col in range(4):

"""创建数字按钮，并将行列号传入该按钮的点击事件函数"""

button = Button(root,command = lambda x = row,y = col:Button\_Click\_2(x,y),bg = 'black',fg = 'green',font = ("Arial",35))

buttons[row,col] = button #将按钮导入数组

button['text'] = numbers.pop() #设置按钮上的文本

button.place(x = 60 + col \* 60,y = 60 + row \* 60,width = 50,height = 50) #设置数字按钮大小

if button['text'] == ' ': #判断是否为空白按钮，如果是，则记录到空白按钮行列号变量

row\_of\_space = row

col\_of\_space = col

numbers = ['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15',' '] #还原数字列表

root.mainloop() #显示窗口

elif difficulty == 5:

root.attributes("-topmost", True)

row\_of\_space = 0 #存放空白按钮的行号

col\_of\_space = 0 #存放空白按钮的列号

buttons = {} #存放数字按钮的数组

numbers = ['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18','19','20','21','22','23','24',' '] #所有数字文本的列表

shuffle(numbers) #打乱数字列表中的数字顺序

for row in range(5):

for col in range(5):

"""创建数字按钮，并将行列号传入该按钮的点击事件函数"""

button = Button(root,command = lambda x = row,y = col:Button\_Click\_3(x,y),bg = 'black',fg = 'green',font = ("Arial",35))

buttons[row,col] = button #将按钮导入数组

button['text'] = numbers.pop() #设置按钮上的文本

button.place(x = 60 + col \* 60,y = 60 + row \* 60,width = 50,height = 50) #设置数字按钮大小

if button['text'] == ' ': #判断是否为空白按钮，如果是，则记录到空白按钮行列号变量

row\_of\_space = row

col\_of\_space = col

numbers = ['1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18','19','20','21','22','23','24',' '] #还原数字列表

root.mainloop() #显示窗口

else:

print('未完成此类关卡')