### 创建对话框，绘制五子棋，并实现五子棋的输赢判断

import tkinter as tk

class Gobang:

def \_\_init\_\_(self):

self.board = [[0]\*15 for \_ in range(15)] # 初始化15\*15的棋盘矩阵

self.player = 1 # 玩家1为黑方，玩家2为白方，默认为玩家1先行

self.winner = 0 # 胜利者，0表示无人胜出

self.root = tk.Tk()

self.canvas = tk.Canvas(self.root, width=450, height=450)

self.canvas.pack()

self.draw\_board() # 绘制棋盘

self.canvas.bind("<Button-1>", self.move) # 左键点击事件，下棋

self.canvas.bind\_all("<KeyPress-space>", self.reset) # 按空格键重置游戏

self.root.mainloop()

def draw\_board(self):

for i in range(15): # 绘制横线

self.canvas.create\_line(15, 15+i\*30, 435, 15+i\*30)

for j in range(15): # 绘制竖线

self.canvas.create\_line(15+j\*30, 15, 15+j\*30, 435)

def move(self, event):

if self.winner != 0: # 游戏已结束

return

x, y = event.x, event.y

col, row = x//30, y//30

if self.board[row][col] != 0: # 已有棋子

return

self.put\_piece(col, row)

self.check\_winner()

def put\_piece(self, col, row):

x, y = col\*30+15, row\*30+15 # 棋子中心点坐标

if self.player == 1:

self.canvas.create\_oval(x-13, y-13, x+13, y+13, fill="black") # 绘制黑色棋子

self.board[row][col] = 1

self.player = 2

else:

self.canvas.create\_oval(x-13, y-13, x+13, y+13, fill="white") # 绘制白色棋子

self.board[row][col] = 2

self.player = 1

def check\_winner(self):

for i in range(15):

for j in range(11):

if self.board[i][j]==self.board[i][j+1]==self.board[i][j+2]==self.board[i][j+3]==self.board[i][j+4]!=0:

self.winner = self.board[i][j]

for i in range(11):

for j in range(15):

if self.board[i][j]==self.board[i+1][j]==self.board[i+2][j]==self.board[i+3][j]==self.board[i+4][j]!=0:

self.winner = self.board[i][j]

for i in range(11):

for j in range(11):

if self.board[i][j]==self.board[i+1][j+1]==self.board[i+2][j+2]==self.board[i+3][j+3]==self.board[i+4][j+4]!=0:

self.winner = self.board[i][j]

for i in range(11):

for j in range(4, 15):

if self.board[i][j]==self.board[i+1][j-1]==self.board[i+2][j-2]==self.board[i+3][j-3]==self.board[i+4][j-4]!=0:

self.winner = self.board[i][j]

if self.winner != 0:

winner\_str = "黑方" if self.winner==1 else "白方"

self.canvas.create\_text(225, 225, text=winner\_str+"获胜！", font=("Arial", 20, "bold"), fill="red")

def reset(self, event):

self.board = [[0]\*15 for \_ in range(15)]

self.player = 1

self.winner = 0

self.canvas.delete("all")

self.draw\_board()

if \_\_name\_\_ == '\_\_main\_\_':

gobang = Gobang()