用pygame写军旗游戏课题研究报告

研究背景

简介

我叫胡宗尧，这是我和我的弟弟合作的一个项目：用 $pygame$ [1] 编写军旗游戏。

我对 $python$ 有浅显的了解，并且用 $pygame$ 做过一些小游戏。

我的弟弟叫胡宗禹，他从 C 语言入门，对图形界面不是特别了解，但接触过 $OpenGL$ 。

源码下载地址，提取码：`n3g3`

字体下载地址： 提取码：`4atl`

没有安装$python$ | $pygame$也想玩？

EXE下载地址，提取码：`qano`

背景

作为一个军旗爱好者，我经常在家和弟弟下军旗，翻翻棋 [2] 作为一种冷门的下法，在很多军旗的App中都没有得到实现，所以在这个课题中，我们就用 $pygame$ 实现这个游戏。

研究目的

为了让更多人了解翻翻棋。能玩上军棋，爱上玩军棋。

问题探究

需要的工具

我们在这个项目中使用的工具如下：

- `Sublime Test 3`
- `Windows 7`
- `python 3.5.1`
- `pygame 1.9.4`

项目结构

```
# -*- coding: utf-8 -*-
# Time : 2019/1/29 21:39
# Author : hzy

#初始化pygame
import pygame,random
from pygame.locals import *
from sys import exit

pygame.init()
SCREEN_SIZE = (540,1080)
screen = pygame.display.set_mode(SCREEN_SIZE, 0, 32)
```

```
#所有的变量定义
font = pygame.font.Font("HanYiYanKaiW-2.ttf", 30)
font_height = font.get_linesize()
move=False
compare=False
red_turn=True
select_block=None
clock=pygame.time.Clock()

listfast=[(0,1),(1,1),(2,1),(3,1),(4,1),(0,2),(4,2),(0,3),(4,3),(0,4),(4,4),
(0,5),(1,5),(2,5),(3,5),(4,5),(0,6),(1,6),(2,6),(3,6),(4,6),(0,7),(4,7),(0,8),
(4,8),(0,9),(4,9),(0,10),(1,10),(2,10),(3,10),(4,10)]

list_qi=['军旗1', '司令1','军长1','师长1','师长1', '旅长1', '旅长1', '团长1', '团长
1',
    '营长1', '营长1', '连长1', '连长1', '连长1', '工兵1', '工兵1', '工兵1', '排长1',
    '排长1', '排长1', '地雷1', '地雷1', '地雷1', '炸弹1', '炸弹1', '军旗0', '司令0',
    '军长0','师长0', '师长0', '旅长0', '旅长0', '团长0', '团长0', '营长0', '营长0',
    '连长0', '连长0', '连长0', '工兵0', '工兵0', '工兵0', '排长0', '排长0', '排长0',
    '地雷0', '地雷0', '地雷0', '炸弹0', '炸弹0']

list_circle=[(1,2),(3,2),(2,3),(1,4),(3,4),(1,7),(3,7),(2,8),(1,9),(3,9)]
```

```
list_block=[(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (0,
8), (0, 9), (0, 10),(0, 11), (1, 0), (1, 1), (1, 3), (1, 5), (1, 6), (1, 8), (1,
10), (1, 11), (2, 0), (2, 1), (2, 2),(2, 4), (2, 5), (2, 6), (2, 7), (2, 9), (2,
10), (2, 11), (3, 0), (3, 1), (3, 3), (3, 5), (3, 6),(3, 8), (3, 10), (3, 11),
(4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (4, 8),(4, 9),
(4, 10), (4, 11)]

dAttackNumber={'司令':8, '军长':7, '师长':6, '旅长':5, '团长':4, '营长':3, '连长':2,
'排长':1, '工兵':0, '地雷':9, '炸弹':10, '军旗': 9}
```

```
#函数列表:
def toPygame(x,y,is_circle):

def fastGoWhere(pos,x,y,color,is_gb,times):

def whereToGo(pos):

def drawBlock():

def line(pos1,pos2):

def drawAll():

def select():

def change():

def moveqi():

def isbigger(chiqiText,beichiqiText):

def eatqi(text):
```

```
#初始化dictionary
# ----------------------init------------------------
d={}
for x in range(-1,5):
    for y in range(12):
        a=(x,y)
        if a in list_block:
            i=random.randint(0,len(list_qi)-1)
            d[a]=[list_qi[i][-1],list_qi.pop(i)[0:2],0]
        else:
            d[a]=[]

pygame.display.set_caption("红方走棋")
drawAll()
drawBlock()
# -------------------end====init--------------------

#主循环
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            exit()
        if event.type == MOUSEBUTTONDOWN:
            pos = pygame.mouse.get_pos()
            a=(pos[0]//120,pos[1]//90)
            qi=d[a]
            drawAll()
            if qi==[]:
                if select_block!=None:
                    drawBlock()
                    whereToGo(select_block)
                    if screen.get_at(pos)==(0,255,0):
                        select()
                        moveqi()
                    elif screen.get_at(pos)==(255,0,0):
                        select()
                        eatqi(qi[1])
                    else:
                        select_block=None
                        drawAll()
            elif qi[2]:
                if select_block==None:#无选中棋
                    if ((red_turn and qi[0]=="0") or ((not red_turn) and
qi[0]=="1")) \
                        and qi[1]!="地雷" and qi[1]!="军旗":
                        select_block=a
                        select()
                else:#有选中棋
                    drawBlock()
                    whereToGo(select_block)
                    if ((qi[0]=="0" and red_turn) or (qi[0]!="0" and not
red_turn)) \
                        and qi[1]!="地雷" and qi[1]!="军旗":
                        select_block=a#切换选中的棋
                        select()
                        drawAll()
                        select()
                    elif screen.get_at(pos)==(255,0,0):
                        select()
                        eatqi(qi[1])#吃棋
                    elif screen.get_at(pos)==(0,255,0):
                        select()
                        moveqi()
                    else:
                        select_block=None
                        drawAll()
            elif select_block==None:
                qi[2]=1
                select_block=None
                change()
            else:
                whereToGo(select_block)
                if screen.get_at(pos)==(255,0,0):
                    qi[2]=1
                    if qi[0]==d[select_block][0]:
                        change()
                        select_block=None
                        drawAll()
                    else:
                        select()
                        eatqi(qi[1])#吃棋
                    else:
                        select_block=None
                        drawAll()
            drawBlock()
            whereToGo(select_block)
            if red_turn:
                screen.blit(font.render("红方", True, (255,0,0)), (240,1050))
            else:
                screen.blit(font.render("黄方", True, (255,255,0)), (240,1050))

    if move:
        drawAll()
        drawBlock()
        pos1=(pos1[0]+step_x,pos1[1]+step_y)
        pygame.draw.rect(screen, (255,255,255), Rect(pos1,(60,60)),0)
        screen.blit(font.render(text, True, (255,255*color,0)), pos1)
        if abs(float(pos1[0]-pos2[0]))<=abs(step_x) and abs(float(pos1[1]-pos2[1]))<=abs(step_y):
            move=False
            if compare:
                if isbigger(text,beichiqiText)>0:
                    d[a]=[str(color),text,1]
                elif isbigger(text,beichiqiText)==0:
                    d[a]=[]
                elif isbigger(text,beichiqiText)<0:
                    pass
                compare=False
            else:
                d[a]=[str(color),text,1]
            drawAll()
            drawBlock()
    pygame.display.update()
    # clock.tick(20)
```

研究中的问题

1.安装pygame时屡次出错

解决方案：在网上查阅很多资料得知要用`1.9.4`版本的`pygame`，问题解决

2.工兵移动的问题

众所周知，工兵的移动非同寻常棋子------它在轨道上可以拐弯，这就增加了很多难度。

解决方案：我使用了递归语句判断是否达到拐点，问题解决。代码如下：

```
def fastGowhere(pos,x,y,color,is_gb):
    try:
        if is_gb:
            for a,b in [(-1,0),(0,1),(0,-1),(1,0)]:
                if (a,b)!=(-x,-y):
                    nextPos=(pos[0]+a,pos[1]+b)
                    nextQi=d[nextPos]
                    if nextPos in listfast:
                        if nextQi==[]:
                            pygame.draw.circle(screen, (0,255,0),
                                toPygame(*nextPos,True), 15, 0)
                            fastGoWhere(nextPos,a,b,color,True)
                        elif (not nextQi[2]) or (nextQi[0]!=color):
                            pygame.draw.circle(screen, (255,0,0),
                                toPygame(*nextPos,True), 15, 0)
        else:
            nextPos=(pos[0]+x,pos[1]+y)
            nextQi=d[nextPos]
            if nextPos in listfast:
                if nextQi==[]:
                    pygame.draw.circle(screen, (0,255,0),
                        toPygame(*nextPos,True), 15, 0)
                    fastGoWhere(nextPos,x,y,color,False,times+1)
                elif (not nextQi[2]) or (nextQi[0]!=color):
                    pygame.draw.circle(screen, (255,0,0),
                        toPygame(*nextPos,True), 15, 0)
    except KeyError:
        pass

def whereToGo(pos):
    if pos!=None:
        pygame_pos=toPygame(*pos,True)
        for x in [-1,0,1]:
            for y in [-1,0,1]:
                try:
                    if screen.get_at((pygame_pos[0]+x*44,pygame_pos[1]+y*33))==
(0,0,255):
                        nearbyPos=(pos[0]+x,pos[1]+y)
                        nearbyQi=d[nearbyPos]
                        qi=d[pos]
                        if nearbyQi==[]:
                            if nearbyPos in listfast and pos in listfast:
                                fastGoWhere(nearbyPos,x,y,qi[0],qi[1]=='工兵',1)
                            pygame.draw.circle(screen, (0,255,0),
                                (pygame_pos[0]+120*x,pygame_pos[1]+90*y), 15, 0)
                        elif (not nearbyQi[2] or nearbyQi[0]!=qi[0]) and
(nearbyPos not in list_circle):
                            pygame.draw.circle(screen, (255,0,0),
                                (pygame_pos[0]+120*x,pygame_pos[1]+90*y), 15, 0)
                except IndexError:
                    pass
```

3.递归可能造成的堆栈错误

如果工兵走的路线正好是一个圈时，递归就会永无止境------直至报错。

解决方案：多加了一个times变量用来计数，每次传递到下一个函数时就加一，大于40时就退出递归。

代码很简单：

```
def fastGoWhere(pos,x,y,color,is_gb,times):
    try:
        if is_gb and times<40:
            for a,b in [(-1,0),(0,1),(0,-1),(1,0)]:
                if (a,b)!=(-x,-y):
                    nextPos=(pos[0]+a,pos[1]+b)
                    nextQi=d[nextPos]
                    if nextPos in listfast:
                        if nextQi==[]:
                            pygame.draw.circle(screen, (0,255,0),
                                toPygame(*nextPos,True), 15, 0)
                            fastGoWhere(nextPos,a,b,color,True,times+1)
                        elif (not nextQi[2]) or (nextQi[0]!=color):
                            pygame.draw.circle(screen, (255,0,0),
                                toPygame(*nextPos,True), 15, 0)
        elif times<40:
            nextPos=(pos[0]+x,pos[1]+y)
            nextQi=d[nextPos]
            if nextPos in listfast:
                if nextQi==[]:
                    pygame.draw.circle(screen, (0,255,0),
                        toPygame(*nextPos,True), 15, 0)
                    fastGoWhere(nextPos,x,y,color,False,times+1)
                elif (not nextQi[2]) or (nextQi[0]!=color):
                    pygame.draw.circle(screen, (255,0,0),
                        toPygame(*nextPos,True), 15, 0)
    except KeyError:
        pass

def whereToGo(pos):
    if pos!=None:
        pygame_pos=toPygame(*pos,True)
        for x in [-1,0,1]:
            for y in [-1,0,1]:
                try:
                    if screen.get_at((pygame_pos[0]+x*44,pygame_pos[1]+y*33))==
(0,0,255):
                        nearbyPos=(pos[0]+x,pos[1]+y)
                        nearbyQi=d[nearbyPos]
                        qi=d[pos]
                        if nearbyQi==[]:
                            if nearbyPos in listfast and pos in listfast:
                                fastGoWhere(nearbyPos,x,y,qi[0],qi[1]=='工兵',1)
                            pygame.draw.circle(screen, (0,255,0),
                                (pygame_pos[0]+120*x,pygame_pos[1]+90*y), 15, 0)
                        elif (not nearbyQi[2] or nearbyQi[0]!=qi[0]) and
(nearbyPos not in list_circle):
                            pygame.draw.circle(screen, (255,0,0),
                                (pygame_pos[0]+120*x,pygame_pos[1]+90*y), 15, 0)
                except IndexError:
                    pass
```

4.打包成EXE中出现的问题

报错：应用程序异常退出

解决方法：

查阅了资料得知，是字体没有找到的原因。

只要把我的字体模块`HanYiYanKaiW-2.ttf`放到源码所在的目录下即可

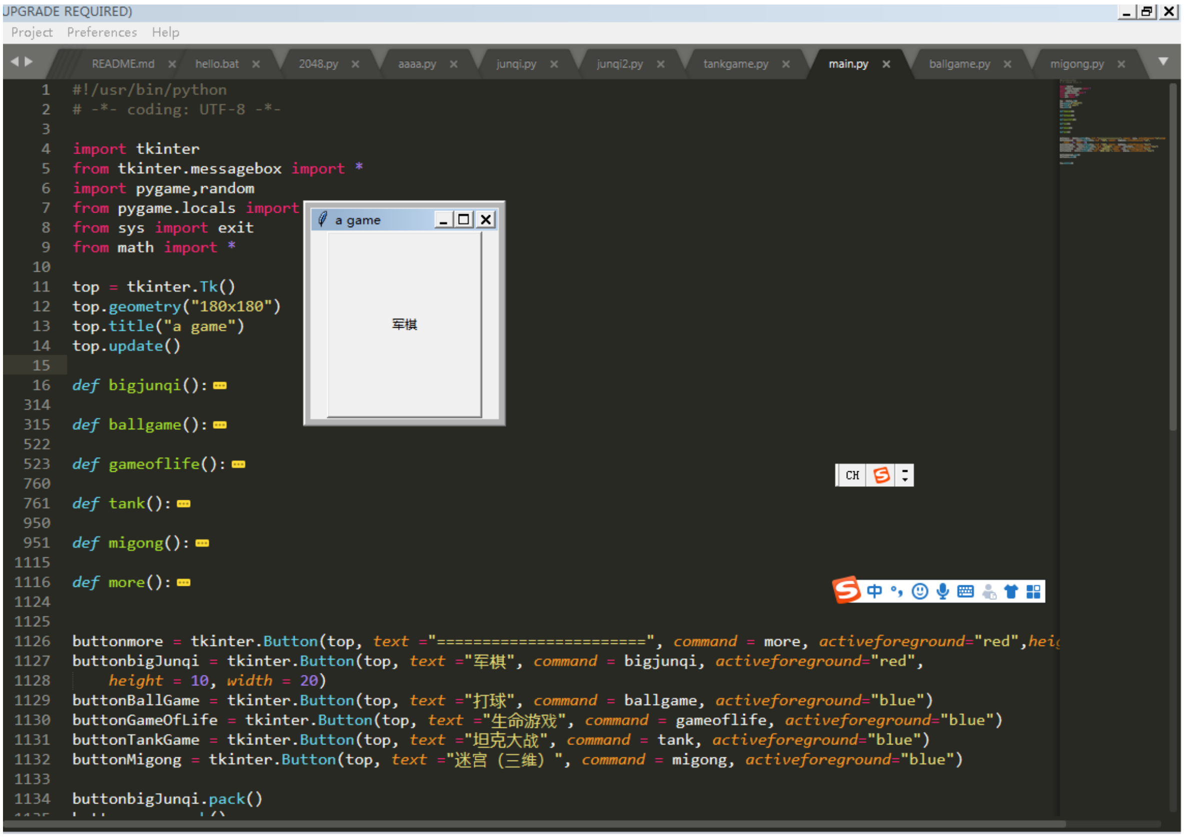
总结

这个军棋游戏还是费了我很多心思的。但是做出来后感到很开心。

成果展示

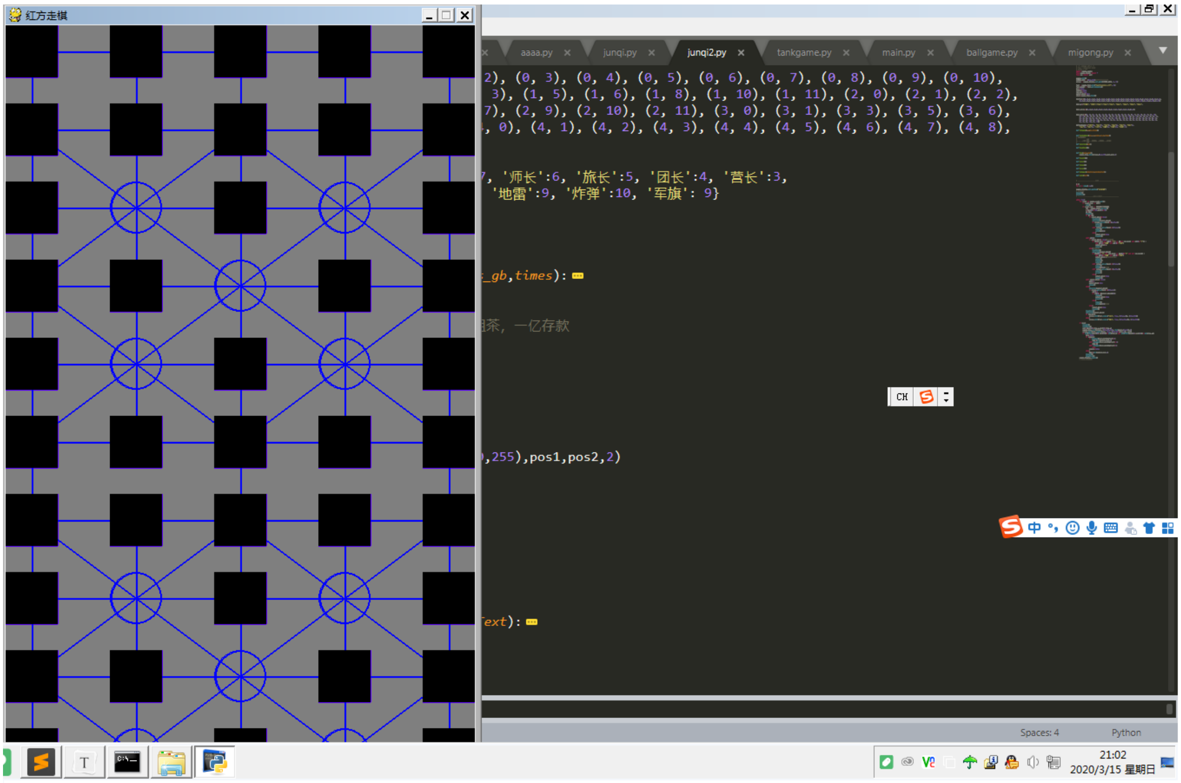
电脑端

这是用`pyinstaller`打包后的游戏界面有个大大的【军棋】框：



```python
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import tkinter
from tkinter.messagebox import *
import pygame,random
from pygame.locals import
from sys import exit
from math import *

top = tkinter.Tk()
top.geometry("180x180")
top.title("a game")
top.update()

def bigjunqi(): ⋯

def ballgame(): ⋯

def gameoflife(): ⋯

def tank(): ⋯

def migong(): ⋯

def more(): ⋯


buttonmore = tkinter.Button(top, text ="=======================", command = more, activeforeground="red",hei
buttonbigJunqi = tkinter.Button(top, text ="军棋", command = bigjunqi, activeforeground="red",
    height = 10, width = 20)
buttonBallGame = tkinter.Button(top, text ="打球", command = ballgame, activeforeground="blue")
buttonGameOfLife = tkinter.Button(top, text ="生命游戏", command = gameoflife, activeforeground="blue")
buttonTankGame = tkinter.Button(top, text ="坦克大战", command = tank, activeforeground="blue")
buttonMigong = tkinter.Button(top, text ="迷宫（三维）", command = migong, activeforeground="blue")

buttonbigJunqi.pack()
```

点了【军棋】后会出现军棋界面：



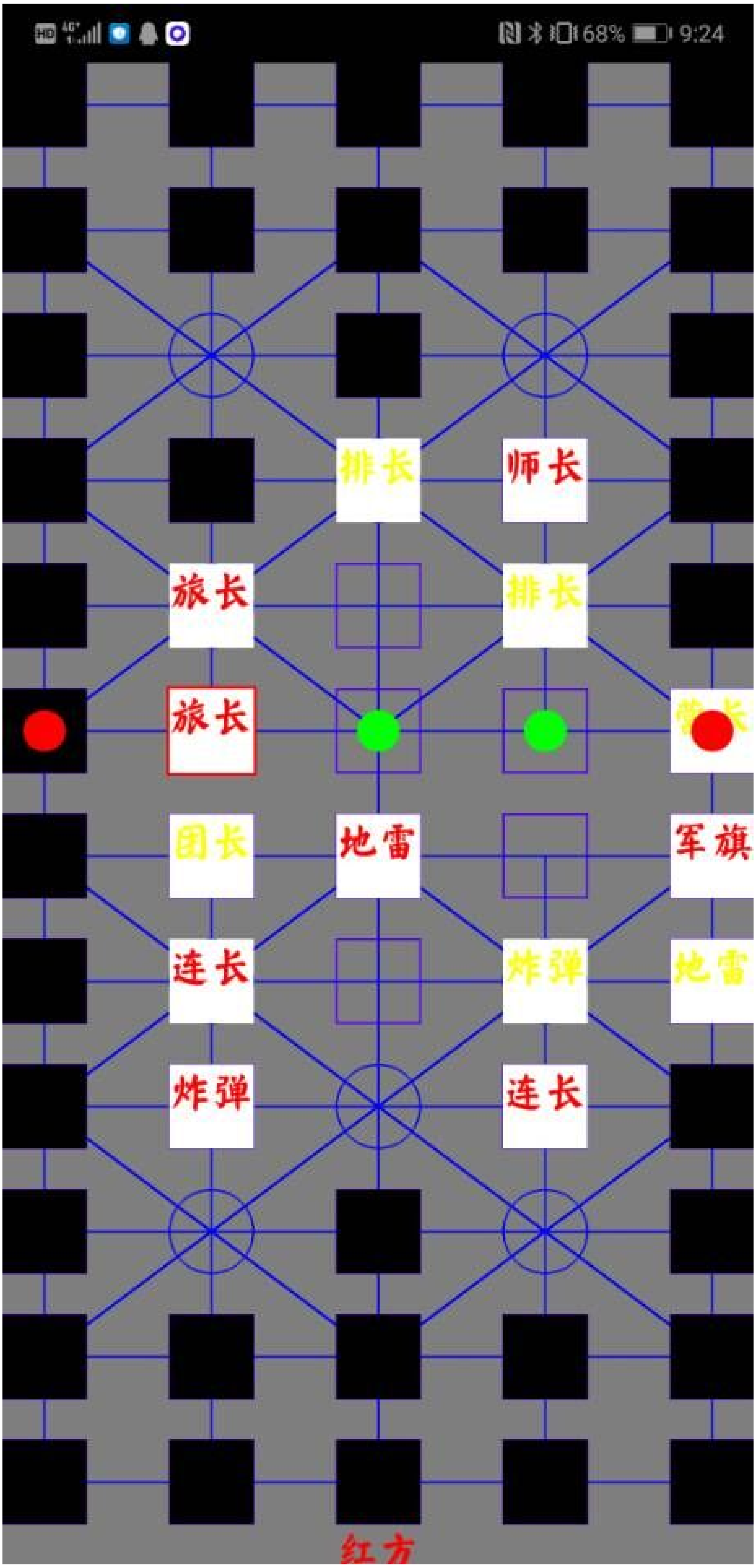
```
2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), (0, 8), (0, 9), (0, 10),
 3), (1, 5), (1, 6), (1, 8), (1, 10), (1, 11), (2, 0), (2, 1), (2, 2),
7), (2, 9), (2, 10), (2, 11), (3, 0), (3, 1), (3, 3), (3, 5), (3, 6),
, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5), (4, 6), (4, 7), (4, 8),


7, '师长':6, '旅长':5, '团长':4, '营长':3,
 '地雷':9, '炸弹':10, '军旗': 9}


_gb,times):


茶, 一亿存款


,255),pos1,pos2,2)


ext):
```

然后就可以愉快的玩耍啦！！！！！！！！！！！！

手机端

在手机端玩翻翻棋的方法：

1. 下载$pydroid$
2. 在$pydroid$上安装$pygame$模块
3. 导入$junqi.py$文件
4. 运行



有待解决的问题

- 被吃掉的军棋无法查看的问题---------------我会马上解决的！

- 最好能弄一个人机对战的军棋游戏---------------这就有点难度了，我最近正好在研究深度学习。相信不久就能训练出来能够进行人机对战的
- EXE文件只能在win7上运行---------------暂时没有时间，但我也想到了一个办法。运行一个win10虚拟机，打包成适合win10的EXE文件。然后在MacBook Pro上也打包一遍，就有了一个Unix可执行文件了。

参考

目光博客：一个IT人的清静小后院

维基百科

[1] 是跨平台Python模块，专为电子游戏设计。包含图像、声音。创建在SDL基础上，允许实时电子游戏研发而无需被低端语言，如C语言或是更低端的汇编语言束缚。基于这样一个设想，所有需要的游戏功能和理念都（主要是图像方面）完全简化位游戏逻辑本身，所有的资源结构都可以由高级语言提供，如Python。—————————-来自维基百科 ↩

[2] 规则说明：2人参与游戏，玩家坐下后，会根据玩家角色性别随机选定军棋翻翻棋的角色秀。若当桌的两个玩家为同性，则角色分别为不同的2个形象。开始游戏后，2人轮流将棋盘上反面放置的军棋翻起以确定控制方，直到某一玩家先翻到与前一张翻到棋子颜色相同，则该棋子的颜色就属于该玩家；开棋方由系统随机选取。棋子分为两色，一共50颗棋子，每方各有25个棋子，分别是：3个工兵，3个排长，3个连长，2个营长，2个团长，2个旅长，2个师长，1个军长，1个司令，2个炸弹，3个地雷，1个军旗　大小：**司令 > 军长 > 师长 > 旅长 > 团长 > 营长 > 连长 > 排长 > 工兵** 地雷：**地雷不能移动。工兵能挖或者炸弹能炸，其他棋子都不能主动触碰地雷。**军旗：不能移动。当对方3个地雷全被挖后，可以用本方任何棋子去扛军旗，挖掉军旗则获胜。**炸弹：**可以移动。能炸掉对方任何棋子（同归于尽），若对方地雷未被挖光，则无法碰撞对方军旗。其他可以移动棋子可以主动碰触炸弹以同时消失。————————-来自百度百科 ↩