**C语言项目报告**

**项目名称： 大脑训练游戏**

**项目成员： 郭嘉禹**

**填写日期：2020年9月6日**

1. **摘要（Abstract）**

**1.项目概述**

本项目设计了一个心理学的记忆游戏，用户首先观察屏幕显示的图片块，在规定的时间内记忆图片块的内容和位置，然后这些图片块被翻过去，用户根据记忆完成图片块的配对，通过两次点击鼠标完成相同图片块的配对。游戏分为三级，随着游戏级别的升高，图片块的数量逐渐增加，游戏难度逐渐加大。第1级游戏是2×2，第二级是4×4，第三级游戏是6×6。

**2.创新点**

该项目采用标准C语言完成所有底层程序设计，包括各种排序算法和文件读写操作。由于C语言的界面功能弱，因此本项目采用python语言完成游戏界面的编写。充分发挥了python语言和C语言的优势，实现的程序界面美现，能吸引游戏玩家。另外，在游戏策略的设计上，充分考虑了难度级别设定和选择，排序方法，程序的鲁棒性等问题。

1. **问题描述（Problem Statement）**

**1.项目背景**

心理学研究表明，大脑训练游戏可以一定程度、一定时间内提高大脑的记忆力和认知能力。因此，心理学界设计了若干种训练游戏，这些游戏可以采用实物的形式也可以通过电子游戏的方式。本项目选择一款用于训练记忆能力的游戏，即找配对的卡片，然后自行设计游戏界面及排行方法，如按游戏时间排序和点击次数排序。每次完成游戏后，都会给出用户在本级别的玩家中排行，以及用户本身在该级别的历史成绩趋势图。直观地显示出用户的记忆能力进展情况。

**2.拟解决的问题**

（1）游戏的难度及级别的设定

（2）基于python的界面设计与实现

（3）基于C语言的底层游戏策略实现

（4）python与C的无缝对接

1. **组内分工（Group Division）**

受疫情影响，本小组只有一名组员，项目的所有工作均有负责人完成。

1. **分析（Analysis）**

本项目的功能框架设计如图1所示。该框架主要包括四个模块，即界面设计模块，游戏策略实现模块，成绩排行榜模块，成绩趋势图等。

界面设计

游戏策略实现

排行榜

成绩趋势图

图1

1. **设计（Design）**

各功能模块的介绍及流程图

程序的整体流程图如图2所示。各模块的功能介绍如下。

**1.界面设计**

完成游戏级别的设定，用户的信息输入及存贮，调入当前级别的m\*m图片块，并合理分布在画布上。当用户完成游戏时，显示排行榜，显示成绩发展趋势图等功能。

**2. 游戏策略**

判断鼠标点击界面的位置，根据坐标值判断用户点击了哪一个图片。检测用户点击的图片位置，然后并判断和上一次点击的结果是不是匹配。如果匹配则继续进行下两个模块的点击，如果不匹配，则把上一次点击的模块再重新翻过去。同时记录用户完成游戏的时间和点击鼠标的次数，并将这些成数据写到文件，rank.txt。

**3. 排行榜**

游戏结束后，由C语言的代码完成后续的排行榜工作。首先从rank点txt的最后一条记录当中读取当前的用户和当前游戏的级别。然后根据该用户的名字和级别，从rank.txt当中提取出当前级别的所有用户记录。然后按游戏时间和点击次数分别排序之后写入文件rank\_clicks.txt.和rank\_time.txt。之后回到Python当中进行排行榜的显示。显示的内容包括当前用户玩家本场游戏所需要的时间和点击次数，然后是在当前玩家在当前级别游戏中的一个排行榜。

**4. 成绩发展趋势图**

在C当中还完成当前用户的所有历史记录的提取，并按照游戏的级别从早到现在进行排序。最后按照游戏的级别画出一个折线图。表明用户在训练过程当中的一个进展的过程。

游戏登录

显示1级游戏界面

用户在XX秒内记住图片

计数器counter=0

图片被翻牌

用户点击第一个图片

翻牌，计数器counter+1

两个图片相同？

将上两个图片复原为未单击

计数器counter-2

游戏结束？

用户点击第二个图片

翻牌，计数器counter+1

Y

N

N

Y

将姓名，点击次数，游戏时间显示，同时写入文件rank.txt

读取文件最后一条记录，获取当前用户名和游戏级别

读取rank.txt中级别等于当前用户级别的所有记录

按点击次数进行排序，写入文件rank\_clicks.txt

按游戏时间进行排序，写入文件rank\_time.txt

从文件rank.txt中读取姓名当于当前用户的记录

当前用户的历史记录按游戏级别排序，写入文件

按点击次数显示排行榜前三位

按游戏时间显示排行榜前三位

显示当前用户历史记录

Python代码

C语言代码

图 2 程序总体流程图

1. **实施（Implementation）**

核心模块函数的具体实现代码

**1.界面设计**

pygame.init()  
  
g\_ClickCount = 0  
background = pygame.image.load("./image/beijing.jpg")  
curSurface = pygame.display.set\_mode((1024,768))  
pygame.display.set\_caption("心理学小游戏")  
m = 2\*level #6  
W = 192 #64\*3 #######################  
n = 2\*level #  
lstBlockRect = [[0 for i in range(m)] for j in range(n)]  
print(lstBlockRect)  
lstBlockFlag = [[0 for i in range(m)] for j in range(n)]  
lstIcon = []  
lstBlockFlag\_over = [[1 for i in range(m)] for j in range(n)]  
print(lstBlockFlag\_over)  
  
game\_over=False  
  
def loadIcon(n):  
 for i in range(1,int(n\*\*2/2+1)):  
 iconname = './image/'+str(i) + '.png'  
 tmping = pygame.image.load(iconname)  
 tmping = pygame.transform.smoothscale(tmping,(int(W/level),int(W/level)))  
 lstIcon.append(tmping)  
 lstIcon.append(tmping)  
loadIcon(level\*2)  
random.shuffle(lstIcon)  
  
def generatePos():  
 for i in range(m):  
 for j in range(n):  
 lstBlockRect[i][j] = (70+i\*(W/level+30),90+j\*(W/level+30),W/level,W/level)  
  
generatePos()  
  
def drawBackGround():  
  
 curSurface.blit(background,(0,0))  
 for i in range(m):  
 for j in range(n):  
 if lstBlockFlag[i][j] == 0:  
 pygame.draw.rect(curSurface,(166,166,166),lstBlockRect[i][j])  
 else:  
 curSurface.blit(lstIcon[i\*2\*level+j],(70+i\*(int(W/level)+30),90+j\*(int(W/level)+30),int(W/level),int(W/level)))  
  
def drawBackGround1():  
  
 curSurface.blit(background,(0,0))  
 for i in range(m):  
 for j in range(n):  
 if lstBlockFlag[i][j] == 1:  
 pygame.draw.rect(curSurface,(166,166,166),lstBlockRect[i][j])  
 else:  
 curSurface.blit(lstIcon[i\*2\*level+j],(70+i\*(int(W/level)+30),90+j\*(int(W/level)+30),int(W/level),int(W/level))) #(lstIcon[i\*6+j],(70+i\*(W+30),90+j\*(W+30),W,W))  
  
def findBlockByPos(mouse\_x, mouse\_y):  
 m = 2\*level #6  
 n = 2\*level #6  
 for i in range(m):  
 for j in range(n):  
 if mouse\_x > lstBlockRect[i][j][0] and mouse\_x <= lstBlockRect[i][j][0] + int(W/level):  
 if mouse\_y > lstBlockRect[i][j][1] and mouse\_y <= lstBlockRect[i][j][1] + int(W/level):  
 print('pos',i,j)  
 return (i,j)  
 print(-1,-1)  
 return (-1,-1)  
  
def drawText(text, posx, posy, textHeight=48, fontColor=(0, 0, 0)):  
 fontObj = pygame.font.Font(None, textHeight) # 通过字体文件获得字体对象  
 textSurfaceObj = fontObj.render(text, True, fontColor) # 配置要显示的文字  
 textRectObj = textSurfaceObj.get\_rect() # 获得要显示的对象的rect  
 textRectObj.center = (posx, posy) # 设置显示对象的坐标  
 curSurface.blit(textSurfaceObj, textRectObj) # 绘制字  
\_input = True  
while True:  
 curSurface.blit(background,(0,0))  
 events = pygame.event.get()  
 for event in events:  
 if event.type == QUIT:  
 pygame.quit()  
 sys.exit()  
 elif event.type == pygame.KEYUP:  
 if event.key == 13:  
 level=int(textinput2.get\_text().split(": ")[1]+'\t')  
 m = 2 \* level # 6  
 W = 192 # 64\*3 #######################  
 n = 2 \* level #  
 lstBlockRect = [[0 for i in range(m)] for j in range(n)]  
 print(lstBlockRect)  
 lstBlockFlag = [[0 for i in range(m)] for j in range(n)]  
 lstIcon = []  
 lstBlockFlag\_over = [[1 for i in range(m)] for j in range(n)]  
 print(lstBlockFlag\_over)  
 loadIcon(level \* 2)  
 random.shuffle(lstIcon)  
 generatePos()  
 \_input = False  
  
 if textinput2.update(events):  
 print(textinput2.get\_text())  
 curSurface.blit(textinput2.get\_surface(),(0,0))  
 pygame.display.update()  
 if \_input == False:  
 break  
\_input = True  
while True:  
 curSurface.blit(background, (0, 0))  
 events = pygame.event.get()  
 for event in events:  
 if event.type == QUIT:  
 pygame.quit()  
 sys.exit()  
 elif event.type == pygame.KEYUP:  
 if event.key == 13:  
 name=textinput.get\_text().split(": ")[1]+'\t'  
 \_input = False  
 for event in events:  
 if event.type == QUIT:  
 pygame.quit()  
 sys.exit()  
 elif event.type == pygame.KEYUP:  
 if event.key == 13:  
 name = textinput.get\_text().split(": ")[1] + '\t'  
 \_input = False  
  
 if textinput.update(events):  
 print(textinput.get\_text())  
 curSurface.blit(textinput.get\_surface(), (0, 0))  
 pygame.display.update()  
 if \_input == False:  
 break  
\_time = 3\*level+4 #10, 让用户记忆图片的时间，随着难度level的加大，记忆的时间增长  
COUNT=pygame.USEREVENT+1  
pygame.time.set\_timer(COUNT,1000)  
while True:  
 # ~ \_time -= 1  
 drawBackGround1()  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 pygame.quit()  
 sys.exit()  
 elif event.type==COUNT:  
 \_time -= 1  
 drawText("You have "+str(\_time)+" seconds",800,100)  
 drawText("to remember!!!",800,150)  
 #clock.tick(1000)  
 pygame.display.update()  
 # ~ time.sleep(1)  
   
 if \_time == 0:  
 break

**2.游戏策略**

start=time.clock()  
while True:  
 mouse\_x,mouse\_y = -1,-1  
 drawBackGround()  
 # drawIcon()  
 for event in pygame.event.get():  
 if event.type == QUIT:  
 pygame.quit()  
 sys.exit()  
 elif event.type == MOUSEBUTTONDOWN:  
 mouse\_x,mouse\_y = event.pos  
 curIcon = findBlockByPos(mouse\_x,mouse\_y)  
 if curIcon != (-1,-1) and lstBlockFlag != lstBlockFlag\_over:  
 g\_ClickCount += 1  
 print(g\_ClickCount)  
 lstBlockFlag[curIcon[0]][curIcon[1]] = 1  
 index\_icon = curIcon[0]\*n+curIcon[1]  
 if g\_ClickCount % 2 == 1:  
 first\_ClickIcon = [lstIcon[index\_icon],curIcon]  
 print(first\_ClickIcon)  
 else:  
 second\_ClickIcon = [lstIcon[index\_icon],curIcon]  
 print(second\_ClickIcon)  
 if second\_ClickIcon[0] != first\_ClickIcon[0] or second\_ClickIcon[1] == first\_ClickIcon[1]:  
 pygame.time.wait(300)  
 lastIcon = first\_ClickIcon[1]  
 thisIcon = second\_ClickIcon[1]  
 lstBlockFlag[lastIcon[0]][lastIcon[1]] = 0  
 lstBlockFlag[thisIcon[0]][thisIcon[1]] = 0  
 print("Mismatch")  
 print(lstBlockFlag)  
 if lstBlockFlag == lstBlockFlag\_over:  
 drawBackGround()  
 #pygame.display.update()  
 game\_over=True  
 if game\_over:  
 pygame.display.update()  
 break  
 elif lstBlockFlag != lstBlockFlag\_over:  
 pygame.display.update()  
end=time.clock()  
print(end-start)  
file\_object = open("./rank.txt","a+")  
row=name+'\t'+str(format(end-start,'.2f'))+'\t'+str(g\_ClickCount)+'\t'+str(level)+'\n'

**3.排行榜**

void sort\_clicks(User U[],int n) //按点击次数从小到大排序，点击次数相同的，按时间从小到大排序

{

int i,j;

User temp;

for(i=0;i<n-1;i++)

for(j=0;j<n-1-i;j++)

if(U[j].clicks>U[j+1].clicks||U[j].clicks==U[j+1].clicks&&U[j].time>U[j+1].time)

{

temp=U[j];

U[j]=U[j+1];

U[j+1]=temp;

}

}

void sort\_time(User U[],int n) //按游戏所用时间从小到大排序，时间相同的，按点击次数从小到大排序

{

int i,j;

User temp;

for(i=0;i<n-1;i++)

for(j=0;j<n-1-i;j++)

if(U[j].time>U[j+1].time||U[j].time==U[j+1].time&&U[j].clicks>U[j+1].clicks)

{

temp=U[j];

U[j]=U[j+1];

U[j+1]=temp;

}

}

void output\_rank(FILE \*filename,User temp[],int n,char name[]) //将排序结果写入文件filename

{

User R[N];

int i,j,k,m,user\_num,flag,no;//user\_num为用户数，rec\_num为文件中的记录条数

int index;//下标

char res[31];

char buff[4],Le[3],time[10]; // //buff存点击次数，Le存游戏级别

R[0] = temp[0];

user\_num = 1;

index=0;

flag=0; //用于标当前用户是否出现在前三名，如果出现，排行榜只显示前三，否则，显示前三加当前用户，共4条记录

for (k = 0; k < n; k++)

{

if(strcmp(temp[k].name,name)==0) flag=1;

for(j=0;j<k;j++)

if (strcmp(temp[k].name, temp[j].name)==0)

break;

if(j>=k)

{ //如果当前遍历的元素和前一个元素姓名不同，则写入R[index++]中，即同一个用户只写入一次

R[index++] = temp[k];//存到R中

user\_num++;

}

if(index>2) break;

}

if(flag==0)

{

//如果当前用户没进入前三，则查找其在文件中的排名，并将名次序号返回，其记录信息存入R[index]

no=search(temp,n,name);

R[index++]=temp[no];

}

//printf("Number of users:=%d\n",user\_num);

for ( k = 0; k < index; ++k)

{

itoa(R[k].clicks, buff, 10); //将分数转化为10进制的ASCII码，4位，存入buff

itoa(R[k].level,Le,10); //将游戏等级转化为十进制字符串，存入Le

sprintf(time,"%.2f", R[k].time); //将实数时间转化为保留两位小数的字符串

for (i = 0, j = 0, m = 0; i < 31; ++i)

{

if (i < strlen(R[k].name) && R[k].name[i] != '\0')

res[i] = R[k].name[i];

else if (i < 12)

res[i] = '-';

else if (i < strlen(time)+12 && time[i-12] != '\0')

res[i] =time[i-12];

else if (i < 20)

res[i] = '-';

else if (i < strlen(buff)+20 && buff[i-20] != '\0')

res[i] = buff[j++];

else if(i<26)

res[i] = '-';

else if (i < strlen(Le)+26 && Le[i-26] != '\0') res[i]=Le[m++];

else res[i]=0;

}

//printf("res: %s\n",res);

if(k<3)

fprintf(filename, "%d: %s\n",k+1,res);

else

fprintf(filename, "%d: %s\n",no+1,res);

//fprintf(write, "%s\t%s\t%d\n", R[k].name,R[k].time,R[k].clicks);

}

fclose(filename);

}

**4.成绩趋势图**

void get\_user\_score(User U[],int n)

{

FILE \*user\_time,\*user\_clicks;

int i,j;

user\_time = fopen("time\_score.txt", "w");

if (user\_time == NULL)

{

printf("file not found!\n");

exit(0);

}

user\_clicks = fopen("clicks\_score.txt", "w");

if (user\_clicks == NULL)

{

printf("file not found!\n");

exit(0);

}

for(i=0;i<n;i++)

{

printf("%s\t%d\t%.2f\t%d\n", U[i].name, U[i].time, U[i].clicks,U[i].level);

fprintf(user\_time,"%.2f\n", U[i].time);

fprintf(user\_clicks,"%d\n", U[i].clicks);

}

fclose(user\_time);

fclose(user\_clicks);

}

t = list(range(len(Time)))  
plt.plot(t, Time, label='时间',color='g')  
# plt.plot(t, Clicks, label='点击次数')  
plt.title('游戏成绩趋势图', fontproperties='simhei', fontsize=24)  
plt.xlabel('次数', fontproperties='simhei', fontsize=18)  
plt.ylabel('游戏用时（秒)', fontproperties='simhei', fontsize=18)  
plt.show()  
t = list(range(len(Clicks)))  
plt.plot(t, Clicks, label='点击次数',color='r')  
# plt.plot(t, Clicks, label='点击次数')  
plt.title('游戏成绩趋势图', fontproperties='simhei', fontsize=24)  
plt.xlabel('次数', fontproperties='simhei', fontsize=18)  
plt.ylabel('点击次数', fontproperties='simhei', fontsize=18)  
plt.show()

1. **测试（Test）**

功能测试和界面展示

|  |  |
| --- | --- |
| （1）输入游戏级别 | （2）输入姓名 |
| （3）显示游戏界面 | （4）翻牌 |
| （5）游戏过程 | （6）输出排行榜 |
| （7）按点击次数发展趋势图 | （8）按游戏用时发展趋势图 |