* 提交内容:源代码和设计文档(包括问题描述,系统设计,功能模块描述等)

记忆翻牌游戏：与网络上的小游戏“记忆翻牌”的功能基本一致：两两成对，每一对各不相同牌刚开始扣住，玩家在点击牌之后牌可以翻过来若干时间，玩家通过短暂时间的查看记忆住部分牌。当玩家连续点击两张相同的牌时，牌便可以一直翻过来，这个游戏考验了玩家的记忆能力，消遣休闲玩一玩未尝不可，还可以增强记忆力。

我在这个游戏的基础功能上加了这些功能：

a.设置了游戏的三个难度，玩家可以自行选择，每种难度的卡牌个数和种类数都不一样。

b.在所有牌扣住之前有一个倒计时5秒，此时玩家是可以看到牌的内容的，以此减小游戏难度，也更有紧张感和刺激感。

c.牌在被点击一次后只要没有成对，就会马上扣回，不同于原本游戏的第一张牌翻开，第二张牌不成对时才翻回。

d.在游戏结束后，会根据玩家点击卡牌的次数进行分数计算，并且设置了一个历史排行榜，记录下历史高分。

下面是具体详细的设计和功能描述：

1. import已有库graphics，打开窗口，并设置各种参数。
2. 将整个游戏过程以界面划分：init\_interface, start\_interface, help\_interface, difficulty\_choice\_interface, game\_interface, finish\_interface. 并且将每个界面定义为函数，将游戏过程清晰展现出来。
3. 自顶向下设计，设计每一界面。
4. init\_interface：增加animation模块（稍后描述），打开游戏时会有动画展示（运用多线程Threading）和背景音乐（Winsound），增加游戏乐趣。并设置进入游戏后的欢迎语、按钮（开始、帮助和退出）和背景。

2.start\_interface：运用while循环获取玩家点击，判断接下来进入哪一个界面。

3.help\_interface：当玩家点击help后会进入此界面，定义cleanAll函数将之前的界面的出现的项目清除掉（之后每次进入新的画面时都用到相似方法）,并设置帮助文本和开始按钮（效果与初始界面中的开始按钮一致）。

1. difficulty\_choice\_interface：设置easy medium hard 三个按钮，分别对应16张、36张、64张牌。
2. game\_interface：

5.1画出每一张牌的边框（Rectangle），并设置扣牌时的颜色为白色，再设置每张牌的翻过来后的颜色（将所有颜色设置为一个列表并shuffle，以此打乱颜色，保证每局游戏不一样） 并设置矩阵，用0表示牌未被永远翻过来，1则相反。

5.2游戏开始，开始倒计时画面（此时由于需要较大的字体，所以修改了graphics里面的字体大小范围），运用sleep，过一段时间将Text 重设，以此实现倒计时画面。这是所用卡牌是翻过来的，将每种颜色draw出即可，当倒计时结束后所有牌又扣回（颜色变回white）。

5.3游戏核心阶段：

5.3.1首先利用while循环不断获取玩家点击的坐标，并用for循环寻找到玩家所点击的区域，将牌翻过来，将牌的颜色设置为color1，sleep（0.2）后牌会扣回。记录下玩家的点击数（clickNum+=1）

5.3.2重复相同动作，将颜色设置为color2

5.3.3判断color1与color2是否相同，如果相同两张牌都会翻过来，并将矩阵中的0改为1；如果不同会将color2设置为color1，重复以上动作。

5.3.4判断游戏是否结束：在每次点击前都会判断一次，定义Isfinished函数，如果矩阵中所有数都是1则返回True，否则False。

5.3.5游戏核心过程看似不难，事实上有几个问题难以解决：

5.3.5.1当玩家点击已永久翻过来的牌时，也会记录下颜色，当再点击没有翻过来的牌时，也会判断颜色是否相同，如果相同的话就会出现单张牌翻过来的情况。 解决方案：增加if—else语句：当玩家点击已翻过来的牌时会直接跳过下面的步骤。

5.3.5.2当玩家连续点一个位置时，因为颜色相同，所以这个牌会翻过来。 解决方案：增加if—else语句：当玩家连续点击一个位置的牌时，不会产生永久翻回的效果，只记录下第二次点击的颜色。

6.finish\_interface：当Isdinished函数返回True时进入结束界面，又用到animation模块，添加动画效果。设置Play again按钮（按下后会进入difficulty\_choice\_interface） 和Exit 按钮（按下后会退出游戏），另外设置点击数和分数显示以及当前难度排行榜显示。

6.1点击数和分数显示：设计点击数和分数的换算公式，并运用for循环，sleep和setText制造出分数（点击数）从0不断增加到所得分数（点击数）的效果。

6.2难度排行榜显示：增加record模块，历史排行榜的数据储存在文件中，获取玩家本次游戏的分数，将其与历史排行榜分数放入列表中，并运用sort进行比较，最后得到新的排行榜显示在finish\_interface并记录在文件中。

四、模块介绍：

animation：这个模块里面又引入自设的模块images, images里的函数就像是graphics里的Rectangle，Circle一样，但我自设了一个star和circle，用来画星和画圆，并直接设置颜色，尺寸，位置参数，方便许多。animation里面有两大主体函数，pengpeng和twinkle。pengpeng是由曾经做过的python习题笑脸弹壁借鉴而来，将物体的运动分解为x方向上和y方向上，当碰壁后改变dx和dy值，再加上一个for循环，另外还增加参数来控制碰壁的速度和时间；twinkle是让图形不停闪烁（利用sleep和不断的draw与undraw），也有时间和速度的设置。

record：初始化打开文件并读取文件中的分数，加入新的分数后进行重新排序（remain\_first3），并写入文件（writefile），输出一二三名（readfile\_1st，readfile\_2nd，readfile\_3rd）。

button：在已有的button模块基础上增加了draw和undraw函数，功能即是使button出现和隐藏。此外改变了按钮的颜色，让界面更加美观。