技术文档

# 游戏一（冰墩墩）

纯属个人爱好，个人爱好冰墩墩。



# 游戏二（贪吃蛇）Greedy Snake

主要技术点：

1. 地图的构建，直接在绿色的背景下，绘制横竖的直线
2. 蛇的控制：
   1. 蛇移动的按键（方向键，上下左右进行操作蛇的方向）
   2. 当蛇的位置与食物的位置相同的情况下，得分进行加1，同时将存储蛇的数组的最后一个位置保留下来，补充成新的身体
   3. 蛇没有吃到食物的时候，也就是蛇处于运动状态下，蛇的数组的变换，当蛇移动到新的位置的，蛇头的位置用新的位置替换，蛇的数组向后移动，例如:

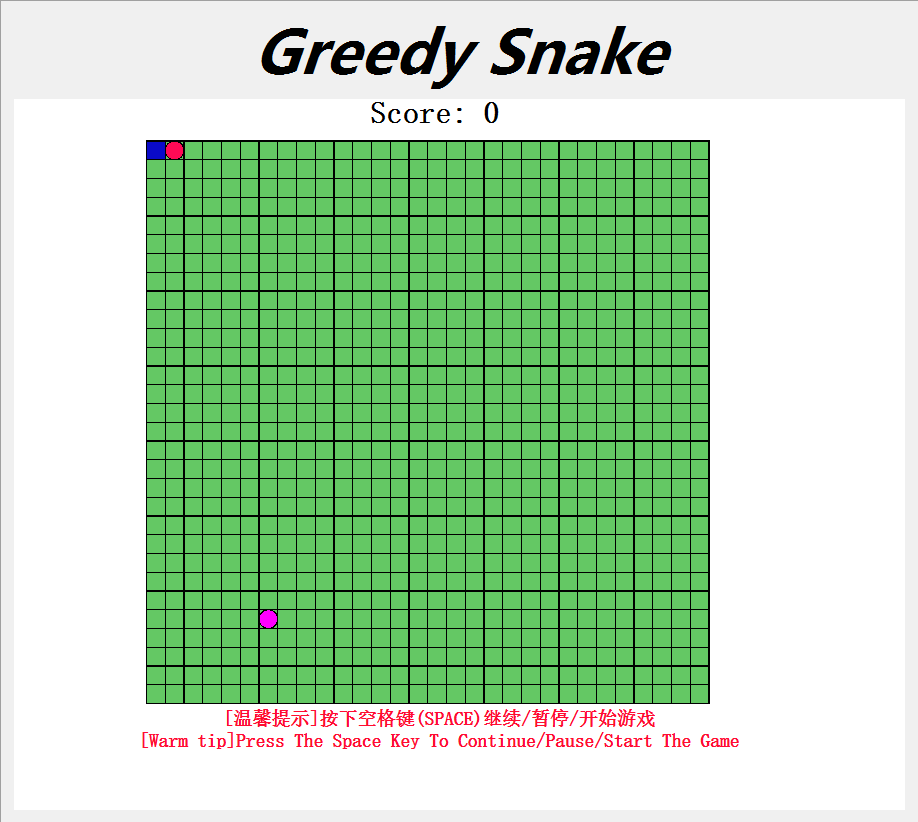
[(2, 0), (1, 0), (0, 0)]向右移动的情况下更新蛇头的(2, 0)->(3, 0)，将原来蛇头的位置更新成后面身体的位置，即(1, 0)->(2, 0), (0, 0)->(1, 0)，即[(2, 0), (1, 0), (0, 0)]-> [(3, 0), (2, 0), (1, 0)],修改1号位，剩下的位置后移，最后一个位置值删除

蛇吃到食物的时候，蛇的其他的位置与蛇没有吃到食物的状态一致，此时保留最后的一个位置，补充成新的身体位置，例如：

[(3, 0), (2, 0), (1, 0)]-> [(4, 0), (3, 0), (2, 0),(1, 0)]，吃到食物之后，再次随机生成食物的位置

* 1. 食物的生成，要求食物随机生成的位置不在蛇的身体上，同时保证在地图网格范围
  2. 游戏结束判断，蛇头接触到地图的边缘，蛇头的位置在蛇身体的里面，认定游戏结束

游戏截图：



* 1. 计分方式：蛇吃到食物，得分增加

操作方式：

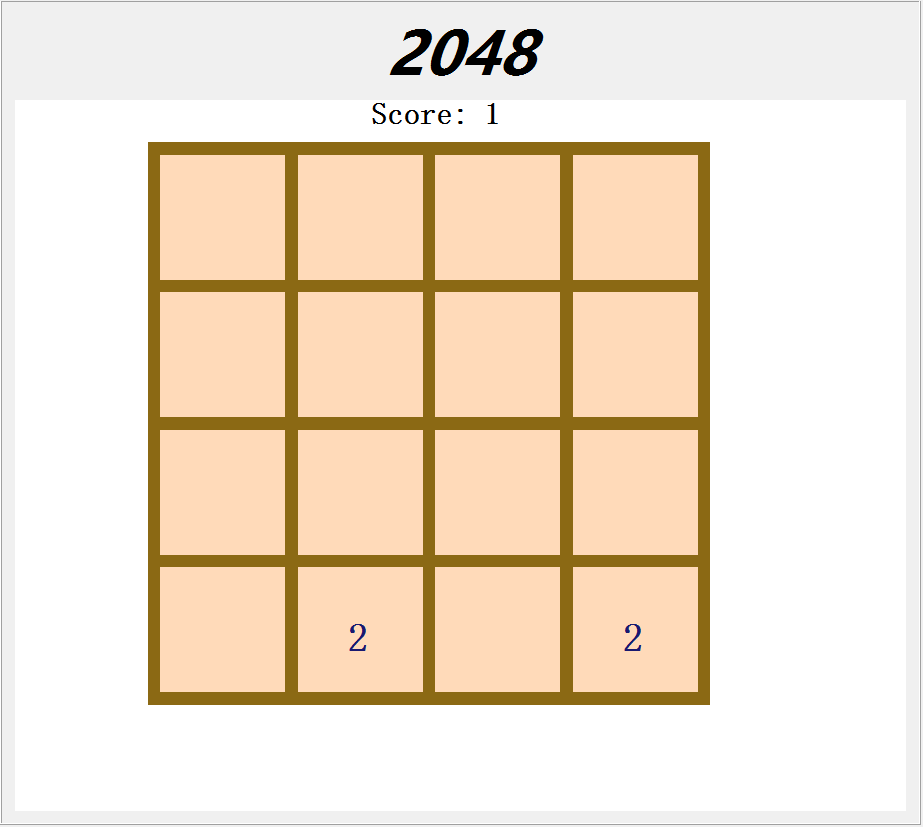
方向键进行游戏角色的移动

空格（Space）对游戏进行继续/暂停/开始游戏

# 游戏三（2048）

主要技术点

1. 构建4\*4的矩阵，同时生成2048地图
2. 初始值的设定，随机在4\*4的矩阵中，生成位置，对应的位置为2
3. 技术难点：以上移的情况举例，第一列为基准[4, 2, 0, 2]
   1. 首先第一个位置与第二个位置大小不同，且都不是0，那么此时让两者保持不变，同时让位置索引后移，此时得到的第一列的值为[4, 2, 0, 2]，后移之后相当于[2,0,2]
   2. 此时[2, 0, 2]中，首位为2，后一位为0，那么这时候遍历后面的所有的数，遇到第一个不等于0的数，倘若等于首位，那么就将首位的值\*2，将该位置的值设置为0，倘若都等于0，那么直接结束，倘若不等于首位且不等于0，那么久该位置与首位的下一位进行对调
   3. 结束的判断：当整个4\*4的矩阵没有剩下空格的情况下，那么认定游戏结束
   4. 游戏界面截图：



* 1. 计分方式：以整个矩阵中的最大值作为得分

操作方式：方向键进行控制

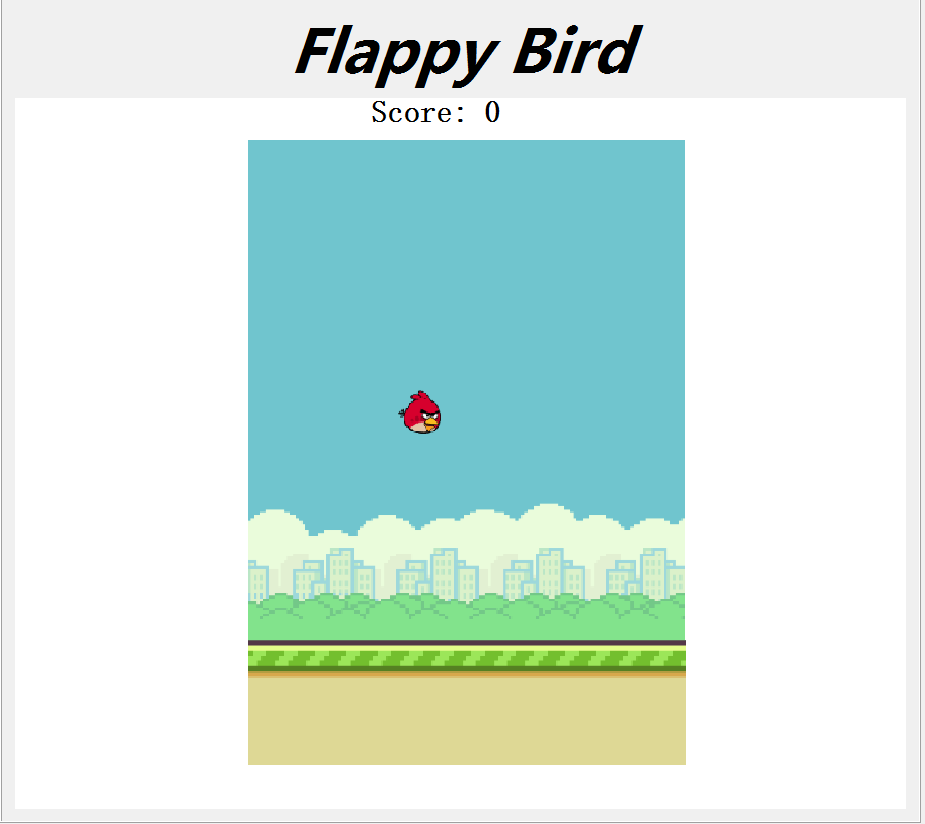
# 游戏四（飞鸟）Flappy Bird

主要技术点：

1. 角色鸟的运动方式：只能上下运动，不做左右运动
2. 管道的生成：随机生成两个随机数，让两个随机数满足两个相加不等于地图的高度以及两个小鸟的高度，即

X1+X2<MAP\_HEIGHT+2\*BIRD\_WIDTH

1. 游戏死亡判断，设置定时器，管道直接从右侧向左侧移动，每次移动的像素值为10，判断鸟与两个管道直接构成的矩形是否存在交集，存在交集的时候，表明游戏结束，同时鸟的接近地面表明游戏结束
2. 操作说明：使用空格按键进行操作，按下鸟上走，松下之后鸟下走，同时向下走的时候，鸟的速度越来越大，满足重力原因。
3. 游戏界面截图



1. 游戏积分的方式：小鸟过几个管道，则得几分。