

界面开始



游戏规则

选中的两个方块是相同的

两个选中的方块之间连接线的折点不超过两个（由与X轴和Y轴的平行线组成）

程序设计思路

1、图标方块的布局

游戏中一共有12种方块，，每种方块个数随机，先安顺序将每种图片排好放入列表 `tmp_list`中，然后用`random`打乱列表元素的顺序，依次从 `tmp_list` 中取一个图标方块放入 `m_list`中。

```
# 构建成对数据
for i in range(0, settings.map_total, 2):
    # 随机生成成对的图片元素标号 (1-12)，存放于 tmp_list
    e = math.ceil(random.random()*settings.element_num)
    # double append
    t_list.append(e)
    t_list.append(e)

# 打乱数据
for i in range(0, settings.map_total, 1):
    # 将 tmp_list 中的图片元素随机排列在 m_list
    index = int(random.random()*(settings.map_total-i))
    m_list.append(t_list[index])
    # 删除已保存到 m_list 中的元素
    t_list.pop(index)

return m_list
```

2、连通算法

连接一共有3种情况，我们可以将它分成X轴方向和Y轴方向两种情况。

1)直连方式

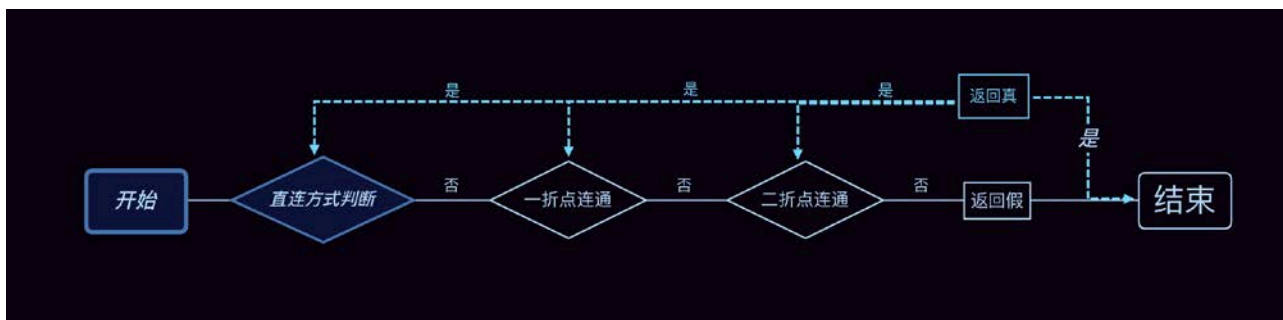
在直连方式中，要求两个选中的方块x和y相同，即在一条直线上， 并且之间没有其他任何图案的方块。直连方式在3种连接方式中最简单。

2)一个折点

这两个方块是一对对角顶点， 另外两个顶点中的某个顶点(即折点)如果可以同时和这两个方块直连，就说明可以“一折连通”。

3)两个折点

这种方式的两个折点(z1、 z2) 必定在两个目标点(两个选中的方块)p1、 p2所在的x方向或y方向的直线上。



以水平或垂直为例，流程图如下：

3、消除图片元素并判断是否消除完毕

```
if settings.points != []:
    settings.points.append(btn)
    # 检查是否相同、能否相连
    if can_clear(settings.points):
        # 消灭它们
        for point in settings.points:
            print("+++++", point.number)
            print("-----", map_list)
            map_list[point.number] = 0
            point.number = 0
            point.hide()
    else:
        # 不匹配，恢复图片状态
        for point in settings.points:
            point.reset()
    # 这次判断完毕，清除记录的点
    settings.points.clear()
```

先点击一个元素，并记录元素。再通过点击第二个点的坐标，来判断是否相同。

不足与反思

- 1、没有画出两个元素相消时的轨迹，不够直观
- 2、如果游戏陷入死局，则无法判断
- 3、界面不够精美，无bgm等
- 4、游戏缺乏创新性