数字搜索游戏

实验概览

实验目的

1. 了解 A^* 算法的基本原理，掌握启发式搜索的基本原理，实现高效路径搜索。
2. 掌握 `JavaScript` 相关开发工具和库，增强代码调试和优化能力。
3. 培养解决问题的编程能力，增强代码调试和优化技巧。

实验要求

1. 深入理解 A^* 算法
2. 搭建数字搜索游戏实验环境
3. 使用 `JavaScript` 实现 A^* 算法
4. 测试实现算法的正确性，运行搜索游戏

加分项

1. 代码中是否存在可以被优化的多余数组？如何对多余数组进行优化？
2. 提高算法搜索效率

提交地址

https://icloud.qd.sdu.edu.cn:7777/link/C1E15F6BD186B8A5D54AB5BD9334CDC5

截止日期

3月30日（星期日）

指导

1. 启发式搜索—— A^* 算法

A^* 算法是一种图遍历和路径寻找算法，因其完整性、最优性和最优效率而广泛应用于计算机科学的多个领域。给定一个加权图、一个开始节点和一个结束节点，该算法可以找到从开始节点到结束节点的最短路径。

1.1 基本原理

A^* 算法是基于图的搜索算法，它通过综合考虑以下两个要素来决定搜索的顺序：

- 实际代价 $g(n)$：从开始节点到中间节点 n 的实际代价，表示已经走过的路径的花费。
- 估计代价 $h(n)$：从中间节点 n 到结束节点的估计代价，通常使用一种启发式方法来估算剩余的路径代价。

由此，可以构造出 A^* 算法的评分函数 $f(n)$：

$$f(n) = g(n) + h(n)$$

A^* 算法的目标就是每次选择一个“最优”节点前进，即优先向那些 $f(n)$ 值最小的节点移动，这样能够快速接近目标，同时保证最终找到最短路径。

1.2 启发式函数

A^* 算法的关键在于启发式函数 $h(n)$
，这个函数用来估算从当前节点到结束节点的剩余代价。启发式函数必须是“可接受的”，即它不能高估实际的剩余代价，否则可能导致算法无法找到最优路径。常见的启发式函数有以下几个。

曼哈顿距离

- 定义：两个点之间的横向和纵向的总距离，它假设可以在水平和垂直4的个方向上移动。
- 公式：

$$h(n) = |x_1 - x_2| + |y_1 - y_2|$$

- 常用于棋盘状网格环境中，特别是允许上下左右四个方向移动的场景.

切比雪夫距离

- 定义：两个点之间各坐标数值差绝对值的最大值，它假设可以在对角线，水平和垂直的8个方向上移动。
- 公式：

$$h(n) = \max\left(|x_1 - x_2|, |y_1 - y_2|\right)$$

- 适用于可以在八个方向自由移动的环境，如一些策略游戏中的路径规划。

欧几里得距离

- 定义：欧几里得距离是最直接的直线距离计算方法，它假设可以沿着任意方向行走。
- 公式：

$$h(n) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- 适合没有网格约束、可以沿任意方向移动的环境，比如航海或自动驾驶路径规划。

1.3 搜索过程

A^* 算法在搜索时使用两个列表来管理待处理的节点：

- 开放列表：存储待扩展的节点。
- 闭合列表：存储已扩展的节点。

搜索过程可以概括为以下几个关键步骤：

1. 初始化：将开始节点加入开放列表，计算其 $f(n)$ 值。
2. 选择当前节点：从开放列表中选择 $f(n)$ 值最小的节点作为当前节点。如果当前节点是结束节点，算法结束，返回路径；否则将当前节点移出开放列表，移入闭合列表。
3. 扩展邻居节点：对于当前节点的每个邻居
    - 如果邻居在闭合列表中，跳过。
    - 如果邻居不在开放列表中，计算其 $f(n)$ 值，更新邻居的代价和前置节点，加入开放列表。

    - 如果邻居已经在开放列表中，检查通过当前节点是否能得到更低的 $f(n)$ 值，如果能，则更新邻居的代价和前置节点。
4. 重复：继续从开放列表中选择下一个 $f(n)$ 值最小的节点进行扩展。
5. 路径构建：当结束节点被找到时，回溯所有节点，构建路径。

2. 数字搜索游戏

2.1 简介

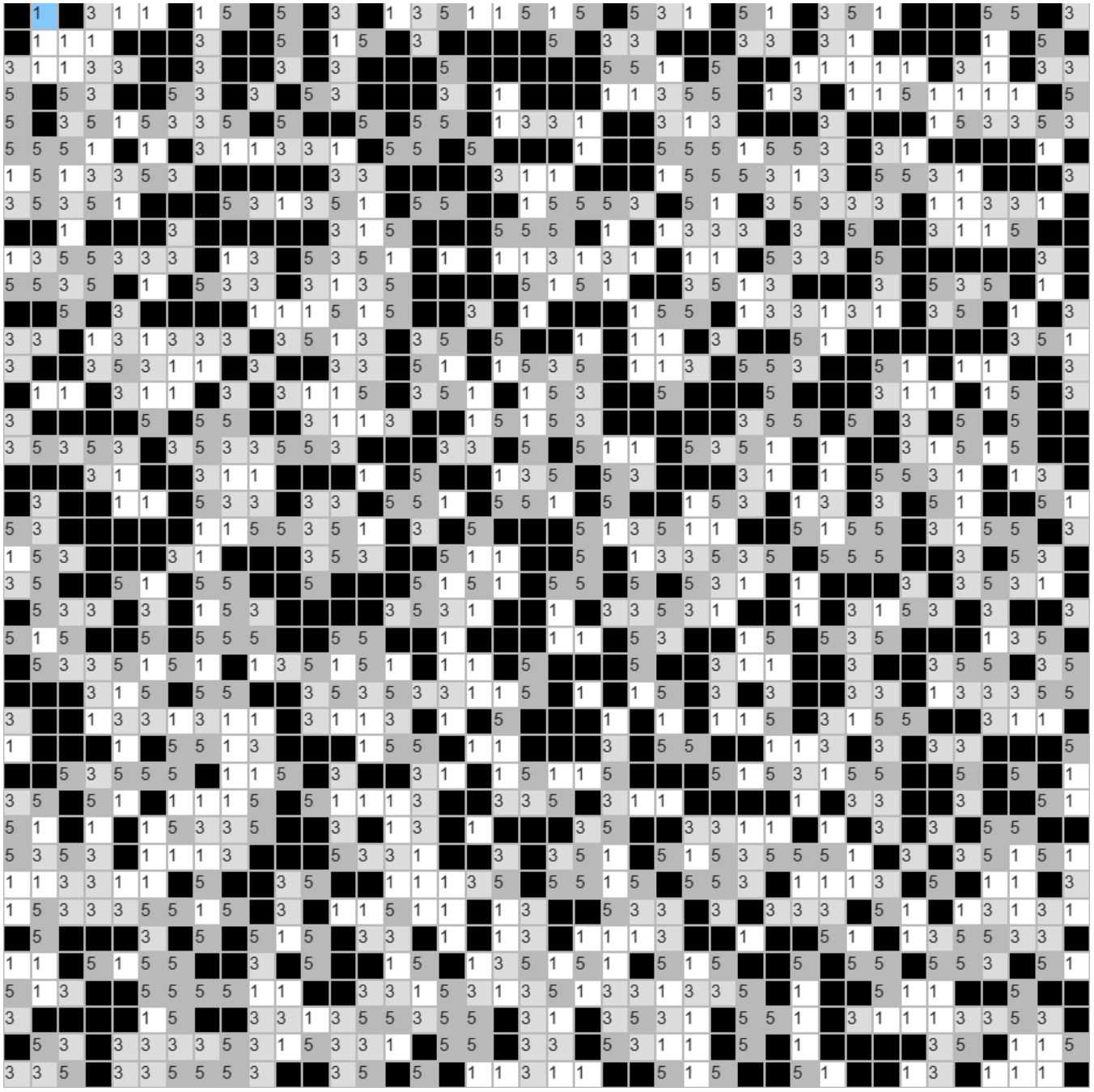
本次实验的数字搜索游戏模拟了在迷宫里的路径寻找过程。

给定一张网格迷宫地图，地图上用不同的色块表示不同的区域

- 黑色色块代表墙壁，权重为 0
- 蓝色色块代表玩家的当前位置
- 其他色块代表道路，可以具有不同的正整数权重；其中，白色色块权重为 1

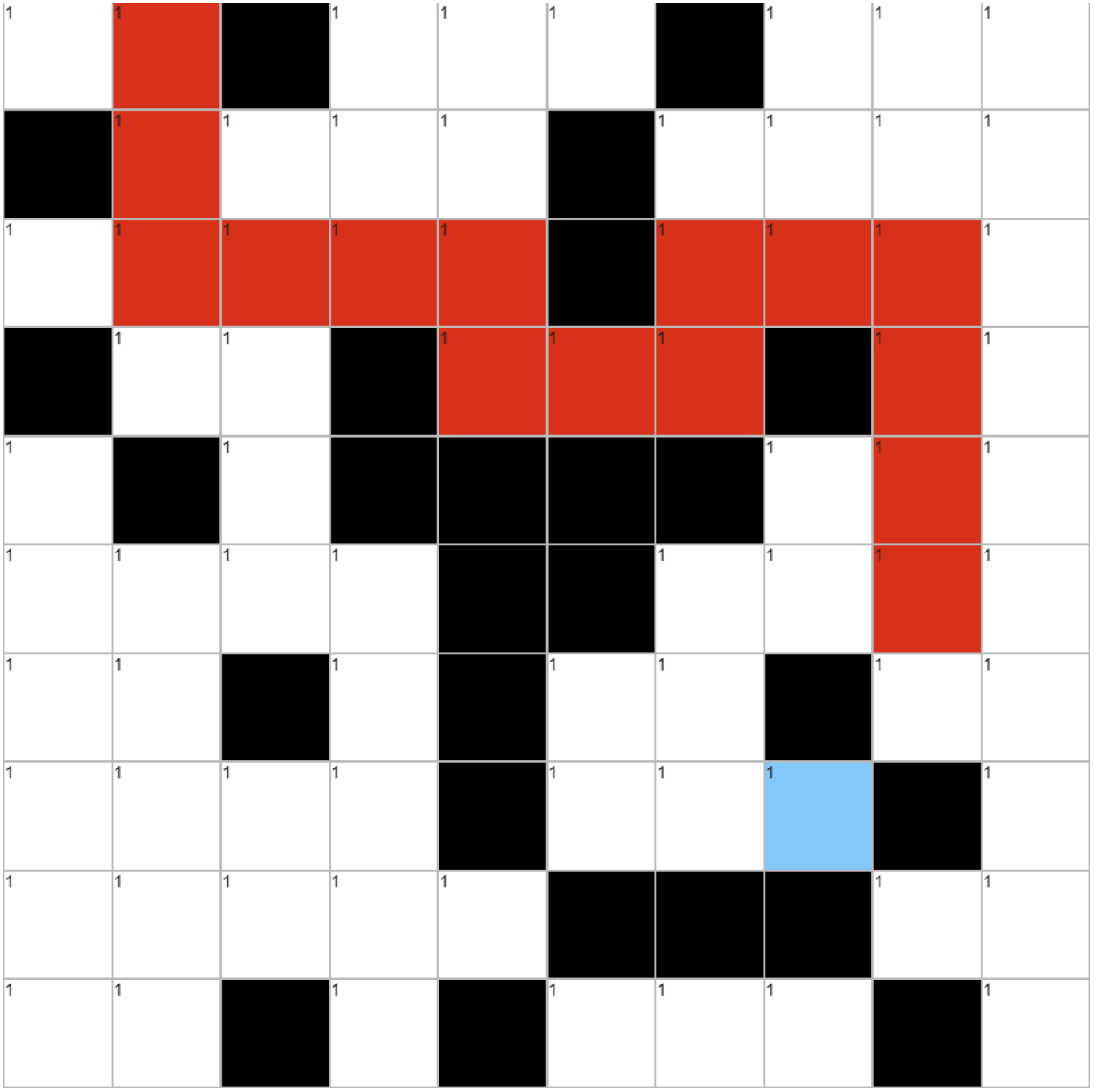
2.1.1 要求

- 通过实现 A^* 算法，寻找任意两个节点之间的最短路径（如果最短路径存在的话）。
    - 起点和终点以蓝色显示。
    - 算法找到的路径会以红色显示。
- 可以选择仅允许上下左右四个方向的移动，或允许上下左右以及对角线八个方向的移动
- 可通行的道路可以具有不同的正整数权重

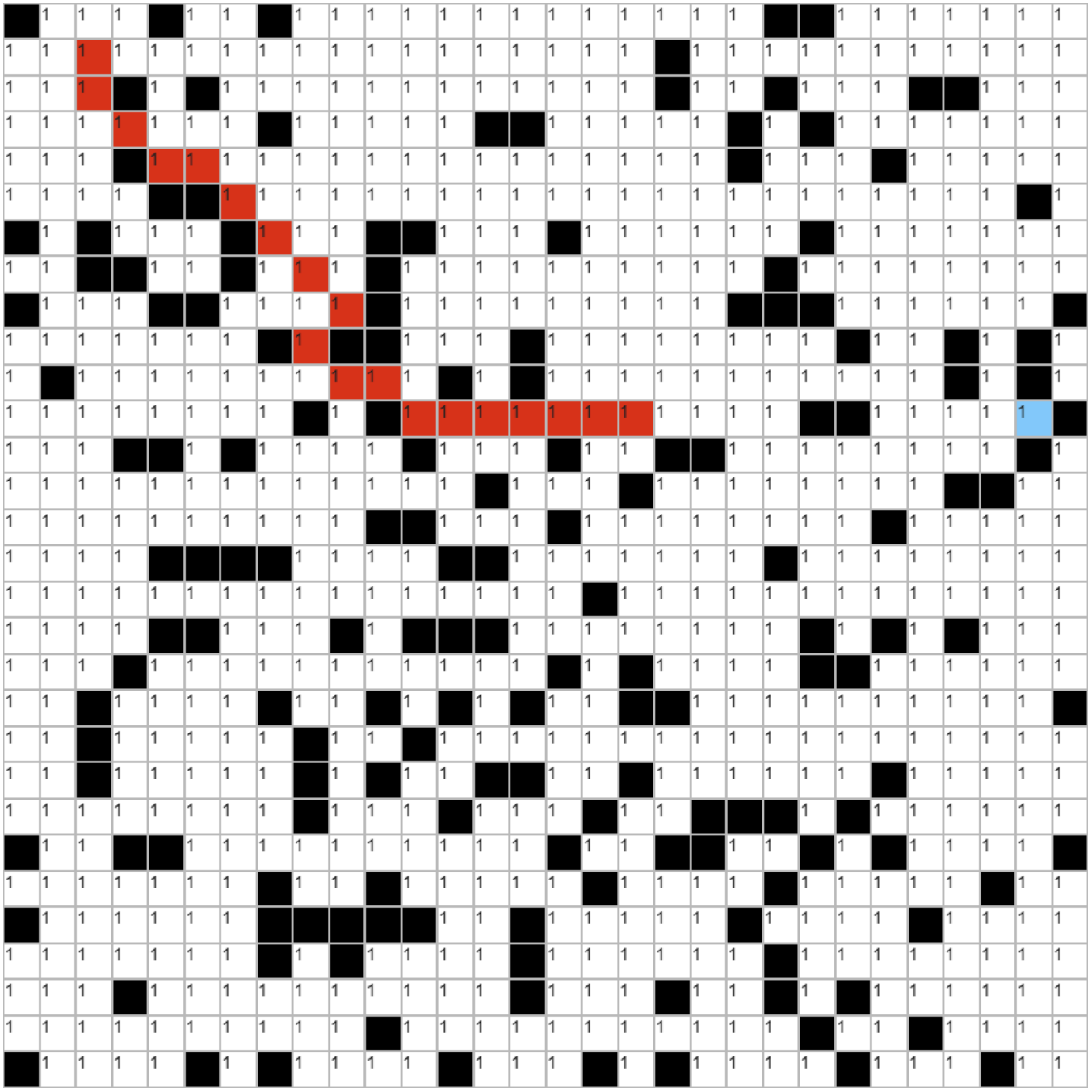


2.2 示例

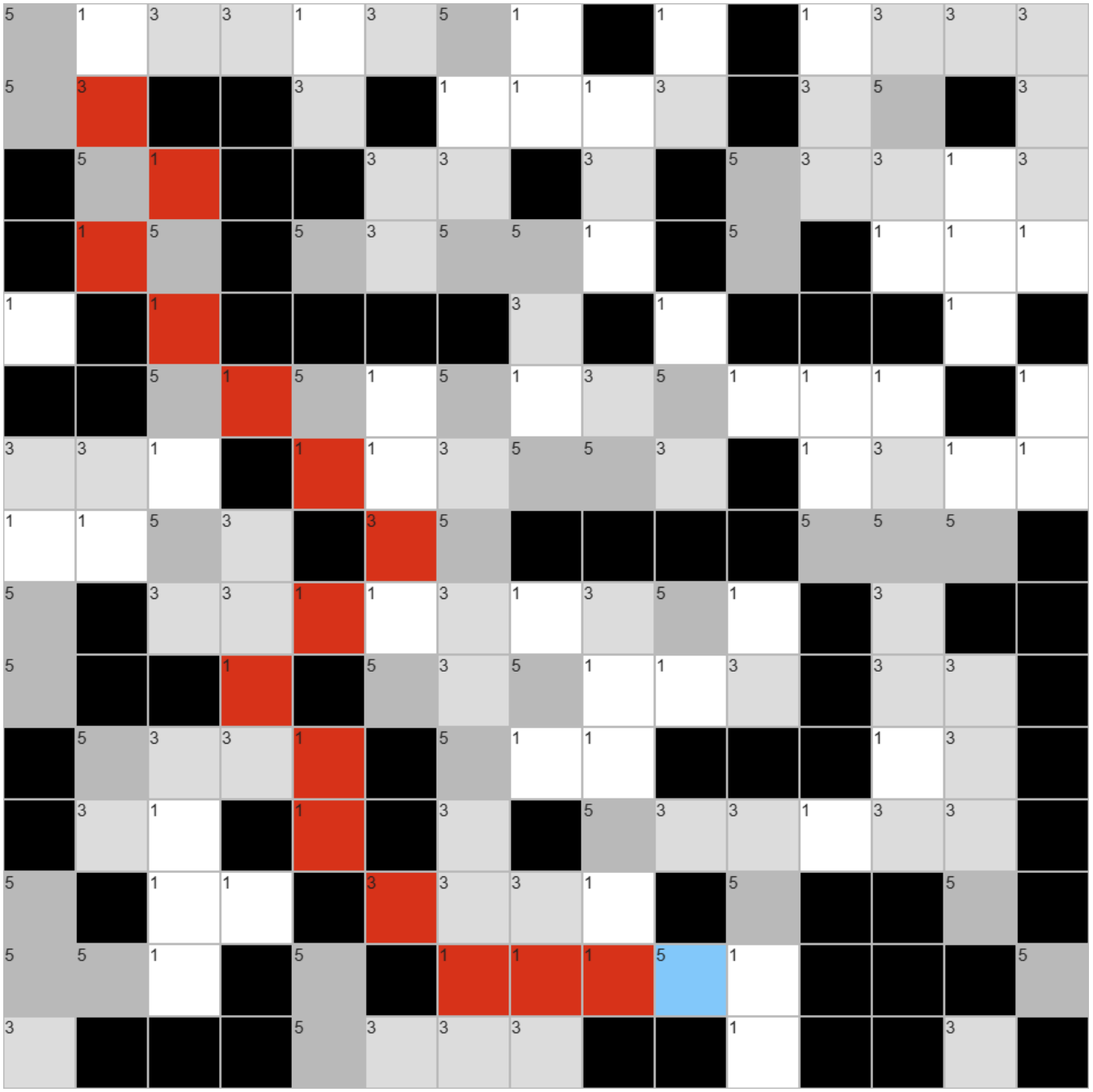
2.2.1 大小为10x10，节点权重只有0和1，仅允许上下左右四个方向移动


1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1

2.2.2 大小为30x30，节点权重只有0和1，允许上下左右和对角线八个方向移动



2.2.3 大小为15x15，节点权重有0，1，3和5，允许上下左右和对角线八个方向移动



3. 游戏实现，运行和测试

- 下载实验压缩包，根据里面的 `README.md` 进行实现，运行和测试