一、算法和程序结构的分析与设计

1.1 理论分析

因为“移动方向仅允许垂直和水平方向移动”而且墙体和边界不能穿越，所以至多有四个子状态。优先搜索距离目的节点更近的节点，按照A\*算法的原理，搜索到目的节点时的路径即为初始节点到目的节点的最短路径。

1.2 算法设计

将启发式函数定义为当前节点目的节点的横纵坐标之差的绝对值之和，按照A搜索算法进行搜索即可。

1.3 程序结构设计

1. Main类就是一个main()方法来新建Client对象以启动程序。
2. ClientGUI用于展示输出结果。buttonNext为“下一步”的点击按钮；start和end为传入的初始节点和目的节点，用于执行A\*算法时的参数；map记录了地图中各节点的信息，也是作为执行A\*算法的参数；mapView为NodeView的二维数组，用于展示输出结果。initMap()方法为初始化地图，即设置墙、初始节点和目的节点的位置；update用于在数据更新后更新用户图形界面以输出结果。
3. HeuristicSearch用于执行A\*算法。closedList和openList分别用于记录closed表和open表；endNode和startNode分别用于记录目的节点和初始节点；steps用于记录初始节点到目的节点的步数，即最短路径的长度。
4. Node用于记录节点信息。f、g和h分别为该节点的f值、g值和h值；fatherNode用于记录其父状态，以便最后输出最短路径；type为节点类型，用于不同节点的类型，从而以不同的颜色和信息输出的界面；x和y为该节点的位置，用于寻找子状态和获取位置。
5. NodeView为图形界面的方格，与节点一一对应。textF、textG和textH分别用于输出该节点的F值、G值和H值。

* 设计类图如图2.3(a)所示
* 算法流程图如图2.3(b)所示
* 获取最短路径流程图如图2.3(c)所示

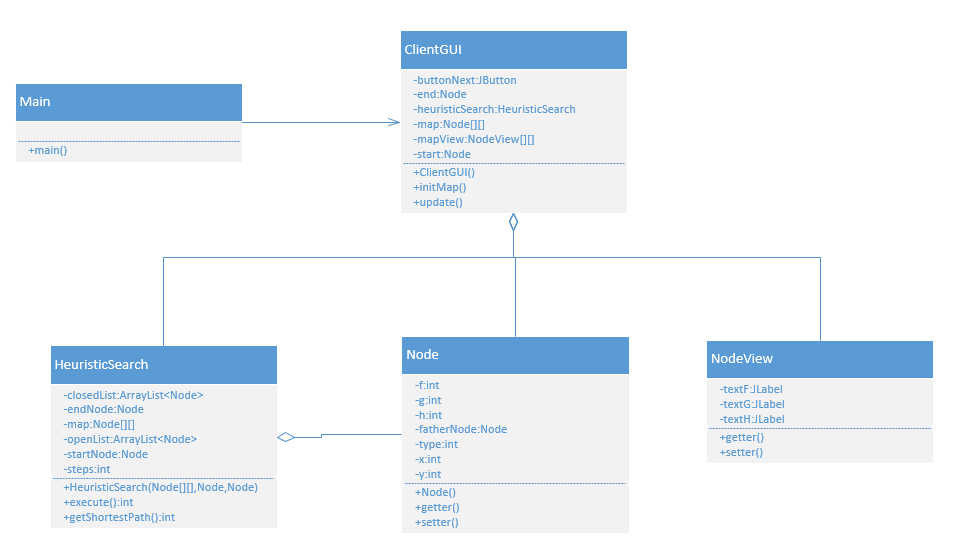


图2.3(a) 设计类图

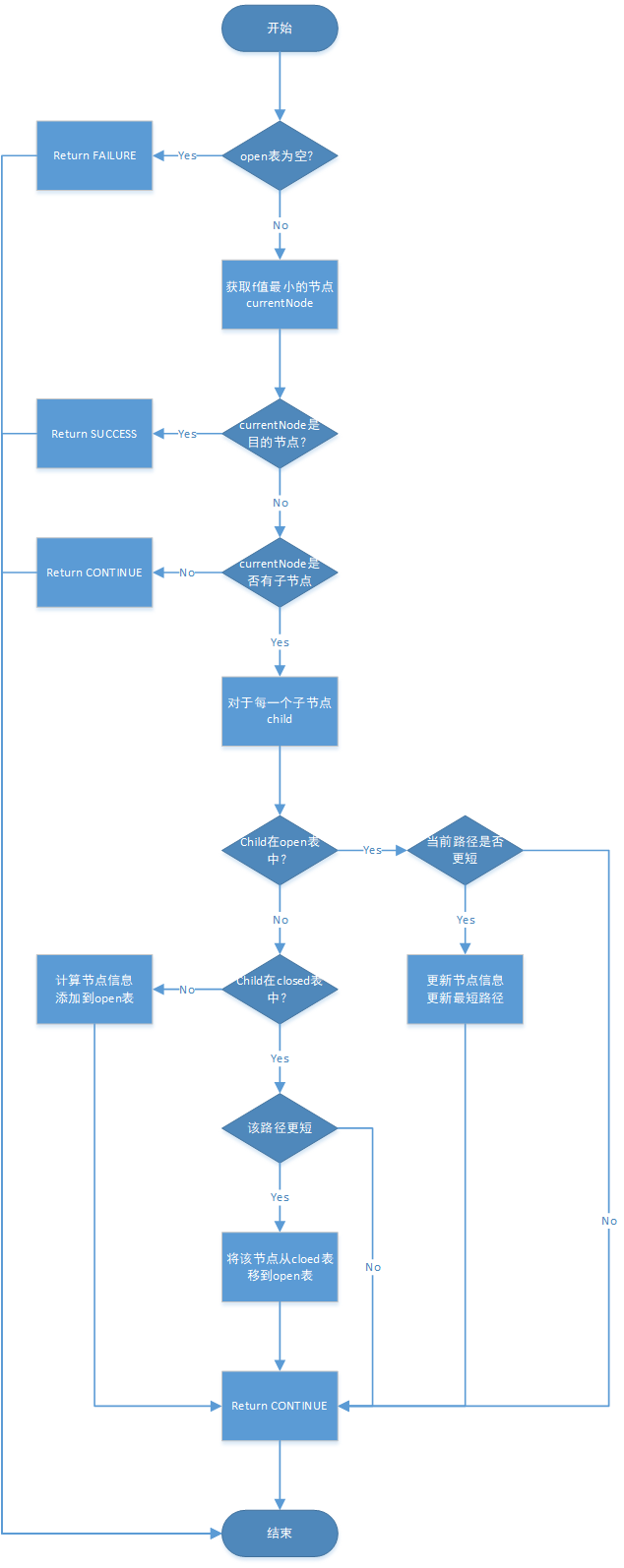


图2.3(b) 算法流程图

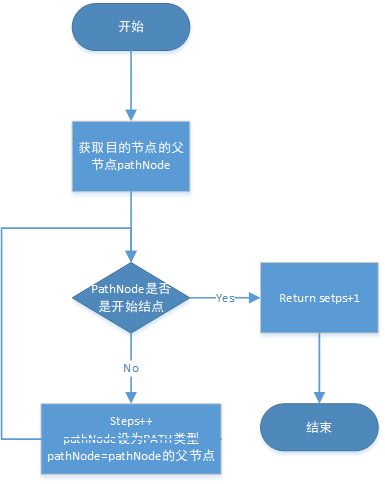


图2.3(c) 获取最短路径流程图

二、启发函数h(n)的定义

h(x)定义为当前节点到目的节点的无障碍距离，即不考虑墙的阻隔，也就是当前节点与目的节点的横纵坐标之差的绝对值之和（|Ax-Bx|+|Ay-By|）。因为不考虑墙的阻隔，所以该值一定小于等于实际值。