总体方案设计说明

1. 软件开发环境：

IDE：vscode，g++编译环境

系统：macOS

运行环境：终端

1. 总体结构图：
2. 模块结构图

图示

描述已自动生成

1. 模块清单

|  |  |  |
| --- | --- | --- |
| 编 号 | 模 块 名 称 | 模 块 标 识 |
| 1 | 主模块 | main |
| 2 | 系统启动模块 | StartSystem |
| 3 | 初始化模块 | Initialization |
| 4 | 导航模块 | Navigation |
| 5 | 查询模块 | Inquire |
| 6 | 时钟模块 | Clock |
| 7 | 算法模块 | ShortestPath |
| 8 | 日志模块 | Log |

1. 模块功能描述：
2. 模块1功能

|  |  |  |
| --- | --- | --- |
| 模块编号：1 | 模块名称：主模块 | 模块标识符：main |
| 输 入 | 处 理 | 输 出 |
| 无 | 1.调用初始化模块初始化校园及用户信息。  2.初始化模块初始化班车时刻表  3.调用启动模块启动系统。 | 无 |

1. 模块2功能

|  |  |  |
| --- | --- | --- |
| 模块编号：2 | 模块名称：启动模块 | 模块标识符：StartSystem |
| 输 入 | 处 理 | 输 出 |
| 1. 命令行输入用户起点 2. 命令行输入目的地数量 3. 命令行搜索目的地，根据输出提示选择目的地。 | 1.初始化时钟模块，默认设置其实时间为8:00  2.用户首次启动系统时，读取用户的起点位置信息，对用户进行初始化。初始化后将不需再次询问用户所在位置，系统维护用户的位置信息。  3．读取用户命令行输入，判断用户目的地和起点是否在同一校区，分别调用单校区导航接口与跨校区导航接口  4.用户目的地数为零时退出系统。  5.支持一定程度上的差错处理 | 1.当遇到非法输入时输出提示。  2.用户搜索地点时输出可选地点列表  3.调用查询模块输出用户附近节点的信息 |

1. 模块3功能

|  |  |  |
| --- | --- | --- |
| 模块编号：3 | 模块名称：初始化模块 | 模块标识符：Initialization |
| 输 入 | 处 理 | 输 出 |
| 无 | 1. 初始化模拟校区一和模拟校区二的建筑物服务设施及道路信息。 2. 初始化定点班车时刻表 | 无 |

1. 模块4功能

|  |  |  |
| --- | --- | --- |
| 模块编号：4 | 模块名称：导航模块 | 模块标识符：Navigation |
| 输 入 | 处 理 | 输 出 |
| 从启动模块接收校区节点道路信息，用户的起点位置信息，目的地列表，以及时钟信息 | 1. 根据用户选择的目的地数量选择单目的地导航与途径目的地导航。 2. 命令行读取用户的策略选择，根据用户选择的导航策略新建重构图。 3. 根据新建图调用算法模块，计算最优路径。 4. 途径目的地导航只能给出一条较优路径。 5. 当用户选择导航开始，模拟用户在时间系统下的运动，输出到命令行并同步到日志文件。 6. 用户可以随时通过敲击键盘的方式暂停系统，暂停时支持（1）用户中断当前导航并重新选择目的地与导航策略。（2）用户查询。（3）结束暂停继续当前的导航。同时将用户暂停时的操作写入日志文件。 7. 时刻维护用户的位置信息。 | 1.输出策略列表供用户选择。  2.输出最优路径及其长度  3.用户选择开始导航后在时钟下模拟用户沿着导航路线的前进，每秒输出一次用户的位置信息与动态信息。 |

1. 模块5功能

|  |  |  |
| --- | --- | --- |
| 模块编号：5 | 模块名称：查询模块 | 模块标识符：Inquire |
| 输 入 | 处 理 | 输 出 |
| 从上层接收用户位置信息，校区节点与道路信息。 | 1. 根据用户所在位置信息更新图结构。用户所在地如果为某一建筑前，则连接一条长度为零的有向边从用户指向该建筑节点。若用户所在地在某一条道路上，则连接两条有向边，从用户分别指向该道路的起点与终点。 2. 根据新建图，调用算法模块，计算从用户节点，到全部节点的最短路径。 3. 设定500m范围内的建筑物与服务设施位于用户附近，将全部满足条件的节点以及与用户之间的最短距离输出。 4. 输出结束后将新加入的边删除。 | 1. 输出与用户之间最短距离在一定范围内的节点的集合。 2. 输出每一个满足条件的节点与用户之间的最短距离。 3. 写入日志文件。 |

1. 模块6功能

|  |  |  |
| --- | --- | --- |
| 模块编号：6 | 模块名称：时钟模块 | 模块标识符：Clock |
| 输 入 | 处 理 | 输 出 |
| 输入初始化时间，默认初始化时间为8:00 | 1. 处理时间，当用户步行或骑行自行车时，真实一秒模拟系统的两分钟。当用户乘坐公交车或定点班车时，真实一秒模拟系统的十分钟。 2. 暂停处理 | 1.按时间轴以分钟为单位输出时间 |

1. 模块7功能

|  |  |  |
| --- | --- | --- |
| 模块编号：7 | 模块名称：算法模块 | 模块标识符：ShortestPath |
| 输 入 | 处 理 | 输 出 |
| 1. 单目的地，从上层接收有向图全部的边，起点信息。 2. 多目的地，从上层接收全部目的地信息，接收任意两点之间的最短距离，起点信息。 | 1. 单目的地，使用ford算法计算单源最短路径，同时维护最短路径上以每个节点为终点的边。即维护了到每一个节点之间的最短路径。 2. 多目的地，使用贪心算法，每次找到距离当前节点最近的目的地，当遍历了全部途径点后从最后一个途径节点走到最终的终点。 | 1. 单目的地，向上层返回一个以起点为源的到任意一个节点的最短路径和最短路径的距离。 2. 多目的地，向上层返回一个较优的途经路径。 |

1. 模块8功能

|  |  |  |
| --- | --- | --- |
| 模块编号：8 | 模块名称：日志模块 | 模块标识符：Log |
| 输 入 | 处 理 | 输 出 |
| 无 | 1. 在每次用户键入信息时和时钟定时器结束时调用日志文件。 2. 将用户键入信息 3. 将用户实时位置信息写入文件 | 1.更新日志文件。 |

1. 接口设计

文本

描述已自动生成

StartSystem，启动系统模块接口

InitOne，一校区初始化接口；InitTwo，二校区初始化接口；

ChangeUVOne，一校区初始化辅助函数，将对应的两条有向边的起点终点互换

ChangeUVTwo，二校区初始化辅助函数，将对应的两条有向边的起点终点互换

InitTimeTable，初始化定点班车时刻表

Inquire，查询模块接口

NavigationInOneCampus，导航模块接口，单校区导航，支持多目的地导航

NavigationInTwoCampus，导航模块接口，跨校区导航

ShortestPath，算法模块接口，执行单目的地的ford算法，计算单源最短路径

ShortestMultiplePath，算法模块接口，执行贪心算法，计算途径策略的较优路径

Pulse，中断响应函数

五．总体执行流程

1. 主模块调用初始化模块，对模拟校区一模拟校区二的建筑物服务设施进行初始化，对定点班车时间表进行初始化。
2. 主模块调用系统启动模块，启动系统。
3. 系统启动模块初始化时钟，默认初始化时间为8:00
4. 首次启动系统，询问用户所在起点，根据用户所在起点对用户进行初始化，要求用户起点必须在两校区中的某一确定的节点上，不能在道路上。同时对不存在的地点输出提示，直到用户选择一个确定的节点。
5. 用户初始化完成后进入导航请求阶段，此阶段用户需要键入目的地数量，当用户输入目的地数量为零时认定用户无导航需求，退出。当用户输入目的地数目后，搜索并选择目的地。系统判断目的地校区，起点校区，目的地数量是否满足系统导航要求，且本系统不支持跨校区的途经最短路径策略。
6. 系统启动模块经过判断，选择调用单校区导航接口或跨校区导航接口。
7. 单校区导航模块，首先询问用户选择最短路径，最短时间，自行车策略，根据用户选择重建图。根据系统启动模块传入的目的地信息判断是单目的地策略或途径目的地策略，分别调用不同的算法接口。
8. 单目的地，调用算法模块单目的地函数接口，计算出最短路径后输出。
9. 多目的地，首先使用warshall算法计算出此校区中任意两节点之间的最短距离，后调用算法模块中的多目的地函数接口，计算出较优路径。遍历较优路径上的全部边，调用单源单目的地函数接口，计算出每条路径的起点与终点之间的最短路径。遍历完成后则计算出了完整的路径。
10. 用户键入选择开启导航。系统按照固定的时钟推进，将用户运动信息实时输出并写入日志文件。
11. 用户敲击键盘暂停程序，进入中断程序，用户可以选择中断导航，继续导航和查询操作。系统根据用户输入做出响应。