function3\_7\_5：gx，20220309

法规法律来源：机动车驾驶人安全文明操作规范7.1：7.1.2 遇下列情况，纵向安全距离应每项增加1s，并逐项累加：b) 在夜间、恶劣气象和不良道路条件下行驶时；

场景3.7-5：在恶劣气象行驶且同车道前方有车

输入：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 参数名称  （英文） | 参数名称（中文） | 参数类型 | 单位 | 说明 | 备注 |
| 1 | p1 | 自车后轴中心点坐标（x,y,z） | Point | - |  |  |
| 2 | p2 | 目标车后轴中心点坐标（x,y,z） | Point | - |  |  |
| 3 | heading1 | 自车航向角 | float | rad |  |  |
| 4 | heading2 | 目标车航向角 | float | rad |  |  |
| 5 | \* length |  | float | m | 该指针的输入是为了通过算法改变length的值，length为自车到目标车距离 |  |
| 6 | visibility | 能见距离 | float | - |  | 需作为关键参数输出 |
| 7 | snow | 雪天 | int | - |  | 需作为关键参数输出 |
| 8 | hail | 冰雹 | int | - |  | 需作为关键参数输出 |

输出：

1. 返回值含义：是否为满足3.7-5（在恶劣气象行驶且同车道前方有车）的场景。

返回值1：满足

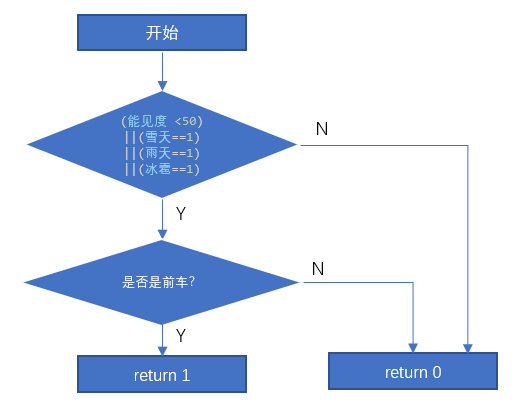
返回值0：不满足

1. \* length ：自车到目标车距离
2. 关键参数：输入的visibility（能见距离）、snow（雪天）、hail（冰雹）

算法逻辑：

（1）判断是否属于恶劣气象：雨/雪/雾（能见度小于50）/冰雹是恶劣气象

（2）判断是否有前车



函数调用示例：

//common

  OpenDrive::OdrManager manager;

  bool flag = manager.loadFile("D:\\gx\\evaluate\\roadInfos\\data\\Germany\_2018.xodr");

  if (flag)

    cout << "load file successful." << endl;

  else

    cout << "load file wrong." << endl;

  OpenDrive::Position\* pos = manager.createPosition();

  manager.activatePosition(pos);

  getPosInfo posManager;

  //function3\_7\_5example

  float pi=4\*atan(1);

  Point p1(5948.24,-3000.27,0);

  Point p2(5909.09,-2967.52,0);

  float heading1=3\*(pi/4);

  float heading2=3\*(pi/4);

  float \* length= new float;

  float  visibility=40;

  int snow=0;

  int rainy=0;

  int hail=0;

  bool function3\_7\_5flag=posManager.function3\_7\_5(p1, p2, heading1,heading2,length, visibility, snow, rainy, hail);