TopoNode说明

1. ConfigManager 和 DataManager

用来存放图源的文件。

目前的产品大致分为地上部分和地下部分。

ConfigManager存储的是地上部分的数据（公区，户型）

DataManager存储的是地下部分的数据（地库）

1. Topo节点

说明： 一种结构，用于存储数据间的关系,每个Topo节点有属于自己的Key和type。其中每个Topo节点的Key都是唯一且与其它节点不同的。Type则是这个topo节点所代表的东西叫什么名字，后续会提到。

Topo节点的定义 ：TopoNode\* a;

1. 结构

Topo节点的孩子有两类分别为childtoponodes(子topo节点)和childs(子图源)

//取子topo节点

std::vector<std::string> children\_keys = a ->childToponodes;

std::vector<std::string>为该Topo节点下所有子Topo节点的Key.

可以通过这些key值去取到该Topo节点。

举例，取第一个TOPO节点

TopoNode\* child\_topo = DataManager::instance()->getTopoNode(child\_keys[0]);

其中DataManager可替换为ConfigManager。具体看上面的应用范围。

//取子图源(此方法并不常用，了解即可)

std::vector<std::string> childkeys = a->childs;

GeometricPrimitives\* gp = DataManager::instance()->primitives[childkeys[0]];

1. Topo关系

常用的topo节点父子关系

1.户上部分

第一级 层topo节点

第二级 层topo节点下有公区topo节点 和 户型topo节点

第三级 公区节点下的房间（如走道，水暖井等）

户型节点下的房间（如客厅，卧室等）

2.地下部分

第一级 地库topo节点

第二级 地库人防 和地库非人防topo节点

第三级 地库非人防下的房间（如换热站 泵房等），

人防节点下目前没有任何东西

注：各节点下的具体房间和名称见

你的工作目录\需求\建筑专业\智能化中英文路径下的房间名称中英文表

1. Topo节点的一些具体应用

TopoNode\* room\_to;

1. 获取智能化节点下的设备、管线等（以内轮廓举例）

std::map<std::string,GeometricPrimitives\*>border\_=room\_to>getChildByType("border\_");

1. 从父类节点找指定类型的子节点（例：从户型下找到客厅）

//户型Topo节点

TopoNode\* httopo;

//获得所有子节点

std::vector<std::string> childekes = httopo->childToponodes;

//遍历

for (const std::string& roomkeys : childekes) {

//取子节点

TopoNode\* roomtopo = ConfigManager::instance()->getTopoNode(roomkeys);

//判断节点类型是否是客厅（起居室）

if (roomtopo->topoType == "LivingRoom") {

//若进来则证明此节点是客厅

}

}

1. 创建topo节点和TOPO节点维护关系
2. 创建Topo节点(以户型举例)

//调用系统函数生成一个不重复的KEY

std::string key= adBuildingModing::BodyModing::instance()->makeKey();

//第一个参数是类型（目前户型的类型为“HT + 数字”）第二个参数是key

TopoNode\* room = ConfigManager::instance->createTopoNode(“HT1”,key);

1. 添加父，子节点

room -> childToponodes.push\_back(childtopokey);

room -> parentToponodes.push\_back(parenttopokey);

1. 往topo节点里添加图源（以圆举例）

osg::ref\_ptr<adPrimitive::Circle> circle = new adPrimitive::Circle;

TopoNode\* room;

room ->childs.push\_back(circle->getName());

DataManager::instance()->addPrimitive(circle);

room->init(DataManager::instance()->primitives);