

MADS用例模型

MADS用例模型

- 1. 系统参与者
- 2. 用例概览
 - 版本与配置
 - 使用帮助
 - 使用日志
 - 数据文件操作
 - 数据可视化
 - 基本数据结构和基本操作
- 3. 系统用例
 - UC1: 以给定格式读取网格数据文件并写入内存(2.4.1)
 - UC2: 平移(2.6.2.1)
 - UC3: 旋转(2.6.2.2)
 - UC4: 对称翻转(2.6.2.3)
 - UC5: 缩放(2.6.2.4)
 - UC6: 合并点(2.6.2.5)
 - UC7: 删除点、边、面和体(2.6.2.6)
 - UC8: 增加点、边、面和体(2.6.2.7)
 - UC9: 计算面的数量(2.6.5.1)
 - UC10: 点的度数(2.6.5.2)
 - UC11: 2.4.4 以给定格式将内存中网格数据写入文件
 - UC12: 2.6.4 迭代Kernel Function

时间	修改内容	备注
2021.3.26	提交了文档的框架以及少部分详细用例	初始阶段的粗略讨论
2021.3.27	添加了关于计算和Mesh操作的详细用例	
2021.04.10	添加了将内存存储数据结构进行本地保存的详细用例	
2021.04.11	添加了部分前置条件	
2021.05.23	添加了关于用户用类库的方法对自己定义的Kernel Function进行迭代的用例	

1. 系统参与者

- 1.1开发人员：调用MADS类库进行编程的人员
- 1.2类库：指提供底层Mesh操作的类库

2. 用例概览

版本与配置

- 2.1. MADS的版本与配置
 - 2.1.1 本机配置检查
 - 解释：例如检测当前OS架构和版本、编译器版本、是否安装/安装了哪些可用的 网格可视化程序和其他MADS需要使用的第三方类库或软件、网络的状况。
 - 2.1.2 版本更新检查
 - 解释：启动时检查MADS版本，询问是否升级（考虑到后续可能会打补丁什么的）

使用帮助

- 2.2. MADS的使用帮助
 - 2.2.1 查询使用教程（帮助文档）
 - 解释：虽然听起来怪怪的，但是对于第一次使用MADS的用户，这个过程是应该包含进“用户的软件使用过程中”。
 - 2.2.2 查询选项帮助
 - 解释：例如Linux上的很多软件会提--help，选项指程序编译、链接、运行等过程中使用的-参数。

使用日志

- 2.3 用户使用日志的记录和查询
 - 解释：例如用户使用此功能查询上次操作的文件、异常关闭前文件的状态。

数据文件操作

- 2.4. MADS与网格数据文件
 - 2.4.1 检查网格数据文件内容的（格式）正确性
 - 2.4.2 将特定字节串转换为特定数据类型
 - 2.4.3 以给定格式读取网格数据文件并写入内存
 - 2.4.4 以给定格式将内存中网格数据写入文件

数据可视化

- 2.5. MADS与可视化（似乎是专业网格可视化软件的工作）
 - 2.5.1 显示静态网格数据
 - 2.5.2 显示动态网格数据
 - 解释：例如一块材料温度随时间变化的热力图。

基本数据结构和基本操作

- 2.6. MADS类库操作（或许根本没有基本数据类型，所有网格数据的建立都是用户用Set、Map、Dat等定义的）
 - 2.6.1 使用系统提供的基本网格数据类型（2D/3D）进行建模（正方体、三角形、球等）
 - 2.6.2 使用MADS提供的通用函数库操作数据
 - 2.6.2.1 平移
 - 2.6.2.2 旋转
 - 2.6.2.3 对称翻转
 - 2.6.2.4 缩放

- 2.6.2.5 合并点
 - 2.6.2.6 删除点、边、面和体
 - 2.6.2.7 增加点、边、面和体
- 2.6.3 使用者使用类库的模板自定义一种网格数据结构。
 - 解释：自定义一种不在基本类型中的单元结构，可以是继承于基本结构、可以是基本结构的组合，也可以直接通过定义Set和Map等来创建结构
- 2.6.4 使用者定义基于网格的操作（查找，遍历，比较，运算等）、核函数并调用类库的函数对Mesh的点和面进行遍历迭代
- 2.6.5 使用者获取Mesh基本的属性数值
 - 2.6.5.1 计算面的数量
 -
 - 2.6.5.2 点的度数

3. 系统用例

UC1：以给定格式读取网格数据文件并写入内存(2.4.1)

- 主要参与者：开发人员、类库
- 前置条件：格式合法且网格数据文件存在
- 主成功场景：
 1. 开发人员调用类库的载入函数，传进参数：文件绝对地址、格式类型。
 2. 查询文件是否存在，存在。
 3. 尝试打开文件，成功。
 4. 本函数调用同类中的确认文件格式合法的函数，确认文件格式无误。
 5. 从数据段开始读取。
 6. 读入一个字节串，将其作为参数调用同类中的串转换函数，并将返回值赋给类中对应成员。
 7. 重复5.直至读入结束符。
 8. 关闭文件。
 9. 返回成功值给调用函数
- 扩展：
 - a.文件访问失败

处理过程：

 1. 调用stat函数查询文件状态，获得对应的错误代码。
 2. 根据错误代码返回对应的错误代码给开发人员。
 - b.文件打开失败

处理过程：

 1. 返回对应的错误代码给开发人员
 - c.格式错误

可能情况：

 1. 指定的格式和文件格式不符。
 2. 文件格式部分已经被破坏
 3. 指定的格式不合法

处理过程：

 - 1.返回对应错误代码给开发人员
 - d.数据出错

可能情况：

 1. 数据段残缺
 2. 非法比特串

处理过程：

3. 记录下路径、文件名、出错位置行标号以及错误类型
4. 打印出错误信息
5. 返回错误代码给调用者

UC2: 平移(2.6.2.1)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要平移的Mesh主体。
 2. 编程者输入平移的主体、主体维度信息、平移的方向以及距离等参数，并调用系统类库中的平移函数。
 3. 系统查询类库，并对被操作主体执行平移操作。
 4. 平移成功，修改后的主体被返回，控制台打印出操作成功的信息。

UC3: 旋转(2.6.2.2)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要旋转的Mesh主体。
 2. 编程者输入旋转的主体、主体维度信息、旋转的方向以及角度等参数，并调用系统类库中的旋转函数。
 3. 系统查询类库，并对被操作主体执行旋转操作。
 4. 旋转成功，修改后的主体被返回，控制台打印出操作成功的信息。

UC4: 对称翻转(2.6.2.3)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要翻转的Mesh主体。
 2. 编程者输入旋转的主体、主体维度信息、翻转的对称轴位置、方向等参数，并调用系统类库中的翻转函数。
 3. 系统查询类库，并对被操作主体执行翻转操作。
 4. 翻转成功，修改后的主体被返回，控制台打印出操作成功的信息。

UC5: 缩放(2.6.2.4)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要缩放的Mesh主体。
 2. 编程者输入缩放的主体、主体维度信息、缩小或放大的选择、缩放中心点的位置、缩放倍率等参数，并调用系统类库中的缩放函数。
 3. 系统查询类库，并对被操作主体执行缩放操作。
 4. 缩放成功，修改后的主体被返回，控制台打印出操作成功的信息。

UC6: 合并点(2.6.2.5)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要合并的点集。
 2. 编程者输入该点集、点集所在的维度信息、合并后点的位置等参数，并调用系统类库中的合并点函数。
 3. 系统查询类库，并对被选中的点集执行合并操作。
 4. 合并成功，返回合并点，控制台打印出操作成功的信息。

UC7: 删除点、边、面和体(2.6.2.6)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要删除的点、边、面或体（为方便描述，统称为被删除主体）
 2. 编程者输入被删除主体、该主体所在的维度信息等参数，并调用系统类库中删除主体的函数。
 3. 系统查询类库，并对被删除主体执行删除操作。
 4. 删除成功，控制台打印出操作成功的信息。

UC8: 增加点、边、面和体(2.6.2.7)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要增加的点、边、面或体（为方便描述，统称为新添主体）。
 2. 编程者输入新添主体、该主体所在的维度信息、主体将要放置的位置和方位信息等参数，并调用系统类库中新添主体的函数。
 3. 系统查询类库，并对新添主体执行添加操作。
 4. 添加成功，控制台打印出操作成功的信息。

UC9: 计算面的数量(2.6.5.1)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要计算的主体（维度在2维及以上，包含一个或多个面）。
 2. 编程者输入主体、主体所在的维度信息等参数，并调用系统类库中计算面数量的函数。
 3. 系统查询类库，并根据主体内部数据结构信息执行计算其面的数量的操作。
 4. 返回计算结果，该结果输出到编程者指定的文件中或者直接输出到控制台。

UC10: 点的度数(2.6.5.2)

- 主要参与者：开发人员、类库
- 前置条件：记录网格数据的文件存在
- 主成功场景：
 1. 编程者确定/选中需要计算的主体（包含一个或多个顶点）。

2. 编程者输入主体、主体所在的维度信息等参数，并调用系统类库中计算点的度数的函数。
3. 系统查询类库，并执行计算主体点的度数的操作。
4. 返回计算结果，该结果输出到编程者指定的文件中或者直接输出到控制台。

UC11: 2.4.4 以给定格式将内存中网格数据写入文件

- 主要参与者：开发人员、类库
- 前置条件：操作系统的文件系统正常
- 主成功场景：
 1. 编程者处理好文件的绝对路径格式、文件名和文件类型，以参数形式调用类库的存储函数。
 2. 类库的存储函数以只读形式打开或者创建文件，以覆盖的形式更新文件内容。
 3. 读出成功，返回正常存储的代码。
- 扩展：
 - a.对指定路径的文件夹没有操作权限
 - 1.判断错误并打印错误信息到终端
 - 2.函数返回错误代码
 - b.存储格式是不支持的类型
 - 1.打印错误信息到终端。
 - 2.函数返回错误代码

UC12: 2.6.4 迭代Kernel Function

- 主要参与者：开发人员、类库
- 前置条件：类的实例已存在且Kernel Function已经定义
- 主成功场景：
 1. 开发人员定义并实现Kernel Function、给对应的Kernel Function参数分配好内存空间并确定遍历迭代的序列数组。
 2. 开发人员将Kernel Function、Kernel Function参数、迭代的元素的名称和序列数组的地址作为参数，调用迭代函数。
 3. 开发人员将迭代函数的返回值（Kernel Function的返回值）赋给自己定义的变量。
- 扩展
 - a.没有序列数组
 - 1.迭代函数将按默认序号从小到大的顺序迭代