

# DS Project Report

姓名：王越越 学号：21307130429 专业：信息安全

注：完成的是bonus

## 1 **bvh\_parser** 字符串处理方案

将**stringstream**与**getline**结合使用具有强大的分隔字符串并实现类型转换的功能：

- (1) 利用**getline** (**file,string**) 顺序逐行读出文件中一行的数据存入字符串**string**中；
- (2) 定义一个**stringstream**对象**ss** (**string**)，在默认状态下以空格分隔字符串，接着**ss >> object**，则可一次将这一行的数据的值赋给变量**object**，并随着变量**object**类型的转化而转化。

## 2 **bvh\_parser** 结构体存储构思

因为**getline**是逐行读取数据，所以对于文件字符串的处理从上至下依次读取然后存进**joint**和**META**结构体中，我们可以用两个函数，分别是**LoadHierarchy**和**LoadMotion**来实现，分别实现文件中**HIERARCHY**与**MOTION**两部分数据的读取与存储。

### ● **LoadHierarchy**函数

在上述处理字符串的方法的基础上，如何读取数据以递归的形式存放入结构体中，此处我利用一个工作栈来实现。

首先将已建的**joint**结构体**root**的指针压入栈中，设置**joint**指针变量**t**用来表示当前处理的**joint**结构体，初始指向**root**，接着逐行处理数据：

- (1) 如果一行的数据第一个字符串是**ROOT**，即设置当下**joint**的**joint\_type**为**ROOT**，接着读入第二个字符串即为**name**；
- (2) 如果第一个字符串为**OFFSET**或**CHANNELS**，即设置当下**joint**的**offset\_x**, **offset\_y**, **offset\_z**和**channels**成员变量的值；
- (3) 如果第一个字符串为**JOINT**，说明当前结构体有孩子，则创建新的**joint**，设置其**joint\_type**及**name**成员变量的值并将其压入栈中，接着使**t**指向当前该孩子；
- (4) 如果第一个字符串为**End**，说明为叶子节点，在创建新的**joint**、压入栈中和使**t**指向当前该**End**之外，设置**joint\_type**自然为**End**，而设置**name**时需取栈顶结构体，其**name**添加**\_End**；
- (5) 如果第一个字符串为**}**，说明该分支结束，进行回退，栈顶不为空即退栈，**t**指向新的栈顶；
- (6) 如果第一个字符串为**MOTION**，说明**HIERARCHY**部分数据处理完毕，循环结束。

### ● **LoadMotion**函数

利用上述处理字符串的办法读取前两行数据存入结构体变量**meta\_data**中；接着将**frame**行的数据依次存入结构体**joint**的**motion**成员变量中，每一行的数据处理方式相同，利用循环可实现，针对一行的数据，相当于遍历一遍**joint**中的成员，可以利用一个工作栈来实现。

读入一行数据，进入循环：

(1) 先将根节点入栈；

(2) 当栈不为空时，进入真正读取数据的循环；

(3) 在读取数据的循环中，每次先设置当前处理的joint指针变量p指向栈顶元素并出栈，接着将joint的所有孩子倒序入栈，以此实现像bvh文件中从上至下读取的顺序；然后将与当前joint通道个数相同数量的数据push进vector中，此vector最后push进motion中。

在main函数中调用LoadHierarchy和LoadMotion两个函数即可完成将数据存入结构体中。

### 3 Struct2Json 数据类型转化与输出方式

此处我调用了开源的cJSON库来输出json格式的数据，利用CreateMeta和CreateJoint两个函数将存在结构体joint和META的数据输出，然后利用Struct2Json函数完成整个过程，最后只需要在main函数中调用即可。

输出joint中的数据时我采用了编写递归函数的方式：对于当前处理的结构体，输出其type、name、offset、channels、motion成员变量的值之后根据孩子数量递归输出其孩子的数据。

### 4 收获与感悟

最大的收获就是首次学会如何去调用开源的库！之前只是听说有这样的方式，但是作为小白没有参加过项目并不知道实际怎么操作，这里也是奔着想看怎么调用库的方式才选择了调用库输出JSON格式数据的方式，而不是暴力输出。其中过程也很简单，但是实现接入，就要仔细阅读对方的规则，cJSON是一个比较经典的库，所以能在网上找到很详细的资料；用别人封装的模块来实现自己的需求，我想是之后团队合作必须要学会的技能，也是现在开源的魅力所在吧。

对于如何将数据存入结构体以及如何输出数据，我想到有建树、编写递归函数和利用栈的方式，最后利用栈来存储、递归来输出则是因为对于建树的知识掌握的并不熟练，“书到用时方恨少”，在实践中发现了自己的薄弱点，之后准备花时间加强下对树的建立、遍历和其他操作的熟练度，然后利用新的方式再来实现下这次的pj。

在debug过程中最容易出现的错误就是segmentation default，最后往往发现是指针越界的问题，敲代码的时候要考虑全面。

还有一点就是对于makefile的改写和ICS课接上了，学来就用，学来能用，感觉很棒！