

# USTC-CG/2024 课程作业 实验报告

## 实验 10

## Character Animation

马天开

PB21000030 (ID: 08)

Due: 2024.04.29

Submitted: 2024.04.29

## 原理概述 Theory

### 骨骼动画 Skeletal Animation

骨骼动画通常由如下几个部分构成:

- 骨骼 *Skeleton*: 由一系列关节 *Joint* 构成, 用于描述模型的骨架结构。
- 蒙皮 *Skinning*: 将模型的顶点与骨骼关联, 使得骨骼的运动能够影响模型的形状。
- 绑定 *Binding*: 定义网格顶点如何受到骨骼关节影响, 通过 *weight* 表示。

### 骨骼关节

骨骼动画的核心在于如何处理关节之间的变换关系。每个关节有自己的局部变换 *Local Transformation*, 并且受到父节点的变换影响, 两者合并形成世界变换 *World Transformation*。

每个关节的局部变换可以用一个  $4 \times 4$  的矩阵  $L$  表示, 包括平移、旋转和缩放:

$$L = T \cdot R \cdot S$$

其中  $T$  为平移矩阵,  $R$  为旋转矩阵,  $S$  为缩放矩阵。

关节的世界变换是其局部变换与父节点的世界变换的乘积:

$$W' = L \cdot W_0$$

### 蒙皮运动

正如上面提到的, 蒙皮顶点的位置是由骨骼关节的变换确定的, 这个过程也被称作蒙皮 *Skinning*. 常用的蒙皮方式是线性混合蒙皮 (Linear Blend Skinning, LBS). 在 LBS 中, 每个顶点的位置  $\mathbf{v}$  由所有关联的骨骼关节的变换加权求和得到:

$$\mathbf{v} = \sum_{i=1}^n w_i T_i B_i^{-1} \mathbf{v}_0$$

## 功能实现 Features Implemented

### 世界变换计算

```
void Joint::compute_world_transform() {
    if (parent_) {
        world_transform_ =
            parent_>get_world_transform() *
            get_local_transform().GetTranspose();
    }
}
```

```
    } else {  
        world_transform_ = get_local_transform().GetTranspose();  
    }  
}
```

## 蒙皮顶点更新

```
void Animator::update_mesh_vertices() {  
    const auto &jointIndices = skel_>jointIndices;  
    const auto &jointWeight = skel_>jointWeight;  
  
    int n = mesh_>vertices.size();  
    int m = jointIndices.size() / n;  
    for (size_t i = 0; i < n; ++i) {  
        GfMatrix4f transform;  
        transform.SetZero();  
        for (size_t j = 0; j < m; ++j) {  
            size_t jointIndex = jointIndices[j + i * m];  
            transform += joint_tree_.get_joint(jointIndex)-  
>get_world_transform() *  
                joint_tree_.get_joint(jointIndex)  
                ->get_bind_transform()  
                .GetTranspose()  
                .GetInverse() *  
                jointWeight[j + i * m];  
        }  
  
        mesh_>vertices[i] =  
            transform.GetTranspose().TransformAffine(mesh_>vertices[i]);  
    }  
}
```