



OptiTrack

OptiTrack 运动捕捉系统

使用说明(下)

Version 2.0.3
October 2017

目录

5 数据应用.....	1
5.1 NatNet SDK	1
5.1.1 SDK 内容.....	2
5.1.2 延时监控.....	4
5.1.2.1 Motive 中延时	4
5.1.2.2 NatNet 中延时.....	5
5.2 Animation.....	7
5.2.1 实时展示.....	7
5.2.1.1 软件要求	8
5.2.1.2 MotionBuilder Plugin 安装.....	8
5.2.1.3 MotionBuilder Plugin 介绍.....	8
5.2.1.4 Motive 设置.....	9
5.2.1.5 OptiTrack Optical 使用	10
5.2.1.5 OptiTrack Skeleton 使用	15
5.2.2 已记录数据应用.....	19
5.2.2.1 MotionBuilder 内数据应用	19
5.2.2.2 3ds Max 内数据应用	23
5.2.3 MotionBuilder 角色化	25
5.2.3.1 角色动画流程.....	25

5.2.3.2 角色模板	26
5.2.3.4 角色化.....	30
5.2.4 VCS 使用	34
5.2.4.1 VCS 系统硬件连接.....	34
5.2.4.2 VCS 系统软件设置.....	36
5.3 Virtual Reality.....	39
5.3.1 Active Marker 跟踪.....	39
5.3.1.1 硬件设置	41
5.3.1.2 软件设置	43
5.3.2 HMD Tracking.....	45
5.3.2.1 Passive Marker	45
5.3.2.2 Active LED HMD bracket.....	48
5.3.2.3 HMD Calibration	49
5.3.3 Unity Plugin	50
5.3.3.1 Motive 设置.....	51
5.3.3.2 Unity 设置.....	52
5.3.4 Unreal Engine Plugin	55
5.3.4.1 Motive 设置.....	56
5.3.4.2 Unreal Engine 4 设置.....	56
5.4 Movement Sciences.....	60
5.4.1 eSync.....	60

5.4.2 AMTI 力学平台	60
5.4.3 Bertec 力学平台	61
5.4.4 NI-DAQ.....	61

5 数据应用

OptiTrack 运动捕捉系统中 ,Motive 用于实时捕捉 Rigid Body / Skeleton 运动数据 , 将运动捕捉数据实时发送到其它应用程序 , 或者将记录的运动捕捉数据导出为通用格式数据并在其它应用程序中打开。根据运动捕捉数据应用的不同情况 , OptiTrack 运动捕捉系统可细分为 4 个不同的应用方向 : Animation、Virtual Reality、Robotics、Movement Sciences。

数据使用方式分为两种情况 实时传输、数据导出。实时传输需要用到 NatNet 或 Plugin , NatNet 可以将实时或已记录的数据传输到客户端应用程序中 , 用户可以通过 NatNet 中提供的示例程序自定义客户端应用程序以接收数据 ; Plugin 是根据 NatNet 开发的适用于特定应用程序的插件 , 不同的应用程序对应特定的插件。数据导出可以查看 Data Export , 根据客户端应用程序选择相应的格式。

5.1 NatNet SDK

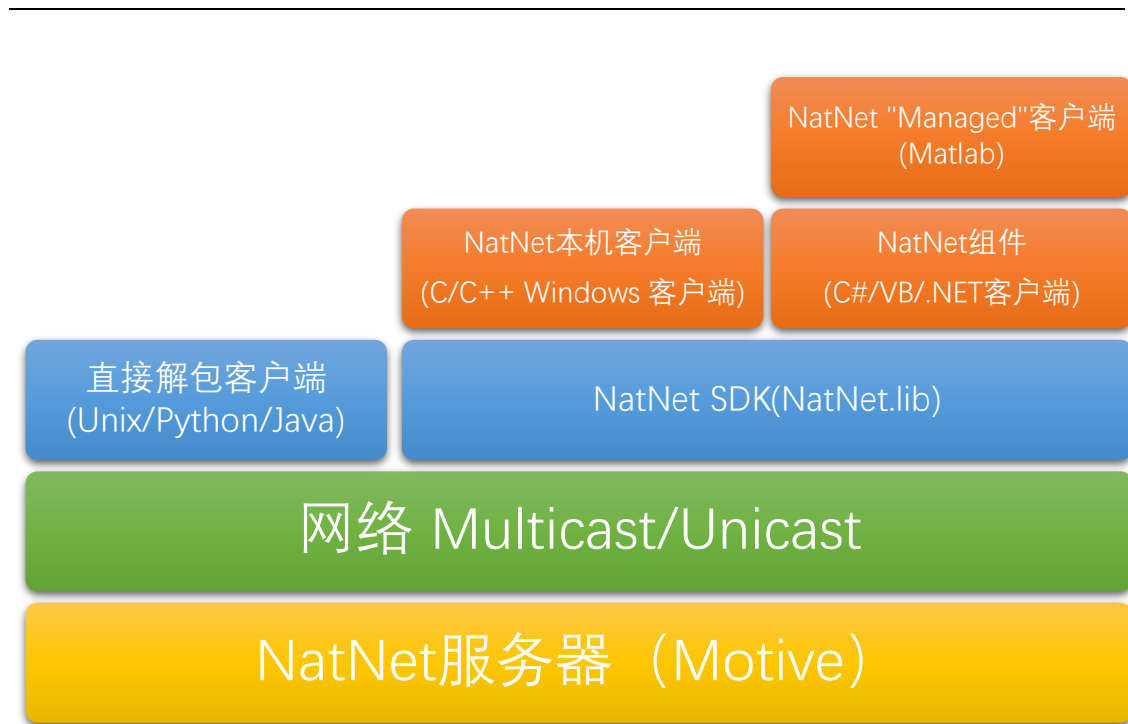
NatNet SDK 是通过网络接收 Motive 数据的网络软件开发包 , 可以实时或记录的运动捕捉数据从服务器 (Motive) 传输到客户端应用程序 , 通过 NatNet SDK 可以开发自定义客户端应用程序以接收运动捕捉数据



并且发送远程命令到 Motive。NatNet 使用 UDP 协议 , 通过 Unicast 或 Multicast 方式发送、接收数据。

NatNet SDK 下载地址 : <http://url.cn/57VirvD>。

下图概述了一个典型的 NatNet 网络设置的主要构成 , 以及在 NatNet 中如何建立服务器和客户端之间的连接。



5.1.1 SDK 内容

- NatNet 库 :本机 C++库 ,包括静态链接库(.lib) ,动态链接库 (.dll)和对应的头文件 ;
- NatNet 组件 : .NET 组件 (NatNetML.dll) ;
- NatNet 示例 :提供一些示例项目和编译的可执行文件。

附加信息 :

- 一个 NatNet 服务器有两个线程和两个 sockets ,一个用来发送跟踪数据至客户端 ,一个用来发送接收指令 ;
- NatNet 服务器和客户端可以存在于同一台/不同的机器 ;
- 多个 NatNet 客户端可以连接到同一服务器 ;
- 当 NatNet 的服务器配置为 IP 组播 ,数据将对播组只广播一次 ;

库头文件 :

包含在 NatNet SDK\include

文件	描述
----	----

\include\NatNetTypes.h	NatNetTypes.h 包含所有通过 NatNet 协议通信的数据格式的类型声明。
\include\NatNetClient.h	NatNetClient.h 包含 NatNet 类的声明。 必须初始化此项目以便接收数据。
\include\NatNetServer.h	NatNetServer.h 包含 NatNetServer 类的声明, 用于 NatNetClient 应用的基础测试, 如需要虚拟发送数据测试。 实际跟踪时, Motive 为 NatNet 服务器。
\include\NatNetRequests.h	NatNetRequest.h 包含一系列用于发送至服务器端 (SendMessageAndWait) 获得数据的指令集。
\include\NatNetCapi.h	NatNetCapi.h 包含 NatNet API 帮助函数的声明。这些函数只用于本机客户端程序。
\include\NatNetRepeater.h	仅供内部使用。

动态链接库文件：

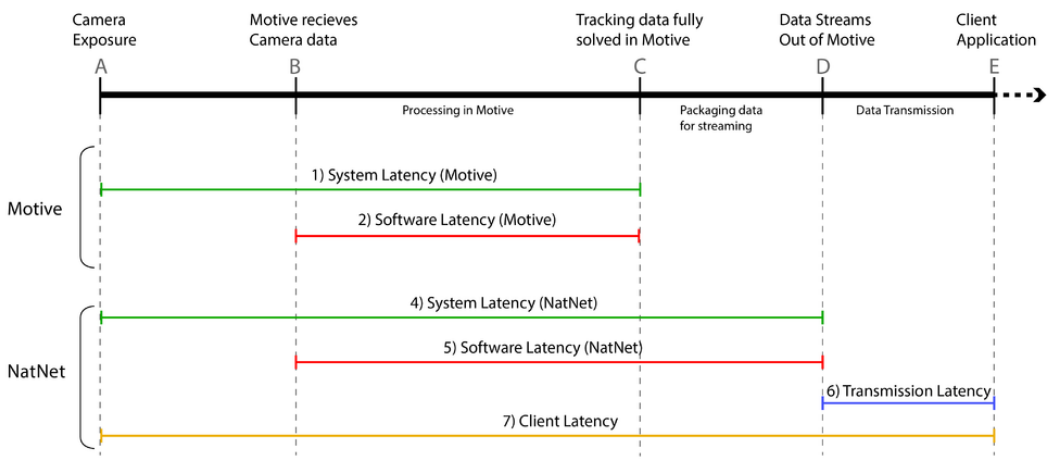
包含在 NatNet SDK\lib

文件	描述
\lib\	包含 NatNet SDK 32 位库文件；
\lib\x64	包含 NatNet SDK 64 位库文件；
\lib\NatNetLib.dll \lib\x64\NatNetLib.dll	Native NatNet library for 32-bit and 64-bit platform architecture. These libraries are used for working with NatNet native clients.
\lib\NatNetLib.lib \lib\x64\NatNetLib.lib	Import library files for 32-bit and 64-bit platform architecture.
\lib\NatNetML.dll \lib\x64\NatNetML.dll	Managed NatNet assembly files for 32-bit and 64-bit platform architecture. These libraries are used for working with NatNet managed clients, including applications that use .NET assemblies. Note that this assembly is derived from the native library, and to use the NatNetML.dll, NatNetLib.dll must be linked as well.
\lib\NatNetML.xml \lib\x64\NatNetML.xml	Includes XML documentations for use with the NatNetML.dll assembly. Place this alongside the DLL file to view the assembly

	reference.
--	------------

5.1.2 延时监控

Motive 传输数据的整个流程如下图所示。

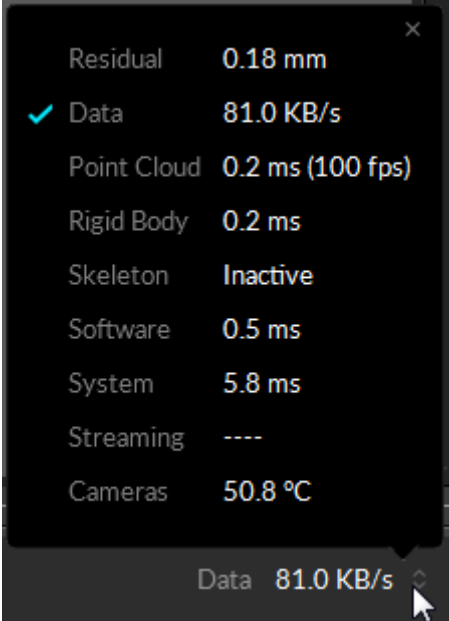


节点	描述
A	相机曝光时刻
B	软件获得相机 2D 数据
C	软件解算完成时刻
D	数据准备被传出时刻
E	客户端接收数据

5.1.2.1 Motive 中延时

Motive 中测量的系统或软件延时会在状态面板中显示，如图所示（仅适用于以太网系统相机）。

- **System** : 表示当前系统延时, 从相机曝光到软件解算完成时刻总的时间延迟。
- **Software** 表示 Motive 处理捕捉数据每一帧所花费的时间, 包括将 2D 数据重建为 3D 数据, 标记和建模为可跟踪资产, 在视口中显示以及在 Motive 中配置的其他流程所需的时间。



Residual	0.18 mm
✓ Data	81.0 KB/s
Point Cloud	0.2 ms (100 fps)
Rigid Body	0.2 ms
Skeleton	Inactive
Software	0.5 ms
System	5.8 ms
Streaming	---
Cameras	50.8 °C

Data 81.0 KB/s

5.1.2.2 NatNet 中延时

在 NatNet3.0 中, 存在新的数据类型允许用户监控流程中特定点的时间戳。NatNet 客户端接收 `sFrameOfMocapData`, 包含下列时间戳:

- 点 A: `sFrameOfMocapData::CameraMidExposureTimestamp`.
- 点 B: `sFrameOfMocapData::CameraDataReceivedTimestamp`.
- 点 D: `sFrameOfMocapData::TransmitTimestamp`.

这些时间戳由主机提供时间, 在计算延时, 需要按时钟频率来划分时间戳值, 以便获得数值。时钟频率包含在服务器包 `sServerDescriptionPacket::HighResClockFrequency`. 为了计算时间, 可以调用 `NatNetClient::SecondsSinceHostTimestamp`.

系统延时 (NatNet)

只适用于以太网系统。系统延时可能不适用于所有配置, 因此建议将此计算包含在条件语句中。

由硬件和软件引入的延时。从相机曝光到数据开始传输的时间间隔。

该数值需要来自 NatNet 协议的两个数值相减获得。

```
const bool bSystemLatencyAvailable = data->CameraMidExposureTimestamp != 0;
```

```
// From SampleClient.cpp (522 ~ 531)

if ( bSystemLatencyAvailable )
{
    const uint64_t systemLatencyHostTicks = data->TransmitTimestamp - data->CameraMidExposureTimestamp;

    const double systemLatencyMillisec = (systemLatencyHostTicks * 1000) / static_cast<double>(g_serverDescription.HighResClockFrequency);
}

```

软件延时 (NatNet)

- 该数值需要来自 NatNet 协议的两个数值相减获得。
- 该延迟表示处理数据所需的时间，同时包含了数据封装时间。

```
// From SampleClient.cpp (502~503)

const uint64_t softwareLatencyHostTicks = data->TransmitTimestamp - data->CameraDataReceivedTimestamp;

const double softwareLatencyMillisec = (softwareLatencyHostTicks * 1000) / static_cast<double>(g_serverDescription.HighResClockFrequency);

```

传输延时

- 该数值来自 NatNet 协议。
- 该延时为 Motive 传输数据包到客户端接收到数据时的时间差。
- 该数值可以通过使用 `sFrameOfMocapData::TransmitTimestamp` 作为输入调用 `SecondsSinceHostTimestampmethod` 获得。

```
// From SampleClient.cpp (507)

const double transitLatencyMillisec = pClient->SecondsSinceHostTimestamp( data->TransmitTimestamp ) * 1000.0;

```

客户端延时 (NatNet)

- 该数值来自 NatNet 协议。

- 该延时为相机曝光时刻到 NatNet 接收到数据时的时间差。为动捕系统的总延时。
- 该数值可以通过使用 `sFrameOfMocapData::CameraMidExposureTimestamp` 作为输入调用 `SecondsSinceHostTimestamp` 获得。

```
// Sample code

const bool bSystemLatencyAvailable = data->CameraMidExposureTimestamp != 0;

if ( bSystemLatencyAvailable )

{

    const double clientLatencyMillisec = pClient->SecondsSinceHostTimestamp(data->CameraMidExposureTimestamp) * 1000.0;

}
```

5.2 Animation

OptiTrack 运动捕捉数据可以应用于影视动漫制作方面，通过记录 Rigid Bodies/Skeleton 的运动数据，导出为其它格式，在影视动漫制作软件中为这些数据添加 Character，驱动虚拟模型运动。在影视动漫制作中，可以实时应用运动捕捉数据，也可以将已记录的运动捕捉数据应用于制作软件中。



5.2.1 实时展示

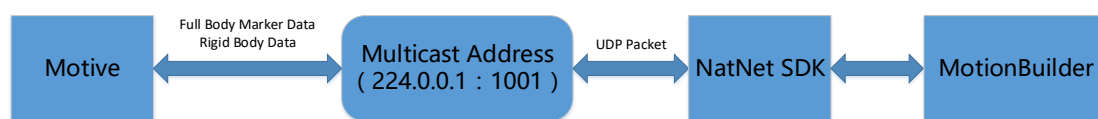
OptiTrack 提供将跟踪的 Rigid Bodies、Skeleton 数据实时传输到 MotionBuilder 的插件，MotionBuilder Plugin 是 MotionBuilder devices, scripts 以及在 MotionBuilder 应用运动捕捉数据示例的集合，允许用户将光学运动捕捉数据映射到 MotionBuilder 动画角色中。

5.2.1.1 软件要求

请在下面地址下载 MotionBuilder Plugin : <http://url.cn/51QGUwA>。

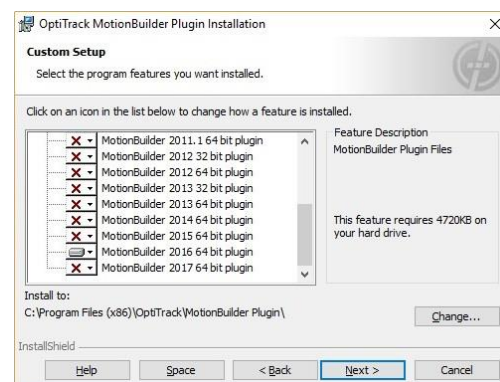
- MotionBuilder Plugin V3.1.0 需要 Motive 1.7+ 版本 ;
- MotionBuilder Plugin V3.1.0 需要 MotionBuilder 2014-2017 (64-bit) 或 MotionBuilder 2009-2013 (32-bit & 64-bit) 等版本。

MotionBuilder 插件客户端/服务器架构如下所示 :



5.2.1.2 MotionBuilder Plugin 安装

- 双击运行 MotionBuilder Plugin 安装包 ;
- 按照提示安装 MotionBuilder Plugin , 建议在默认目录中安装 MotionBuilder Plugin ;
- 安装完成。



注意 :安装时选择与 MotionBuilder 版本对

应的插件 , 建议先安装 MotionBuilder。

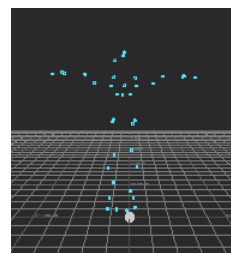
5.2.1.3 MotionBuilder Plugin 介绍

- 打开 MotionBuilder。
- 安装完相应版本的 MotionBuilder Plugin 之后 , 在 Resources → Asset Browser → Templates → Devices 窗口中会显示三个 OptiTrack Devices :OptiTrack VCS、

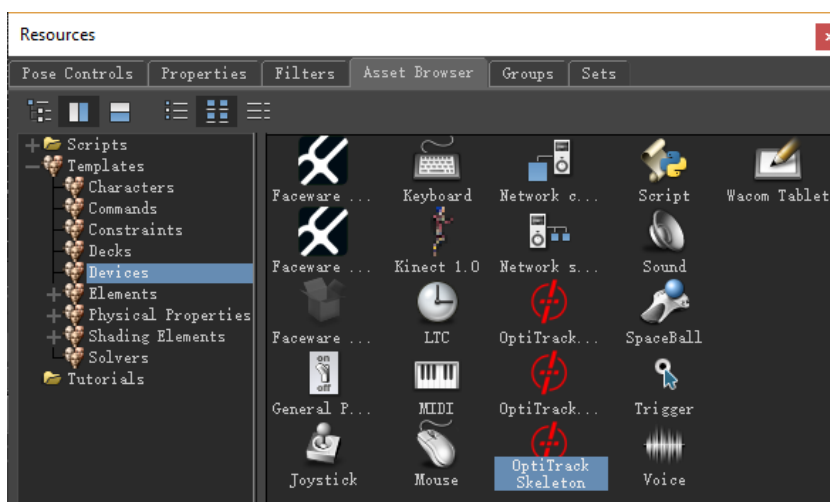
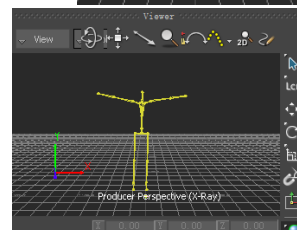
OptiTrack Optical、OptiTrack Skeleton。

OptiTrack VCS : OptiTrack VCS 用于在 MotionBuilder、Maya 中创建虚拟摄像机。

OptiTrack Optical : 允许用户将 OptiTrack 获得的 Marker 点数数据传输到 MotionBuilder 中，映射到 Actor 上。表示 OptiTrack MarkerSet 与 MotionBuilder Actor 之间的单一映射关系。



OptiTrack Skeleton : OptiTrack Skeleton 允许用户将 Motive 6DOF Skeleton 关节角度数据直接映射到 MotionBuilder Character 上。



5.2.1.4 Motive 设置

- 在实时采集模式 (Live Mode) 下，创建需要捕捉的 Assets。
- 打开 Streaming 窗口。
- 按照如所示参数进行设置。

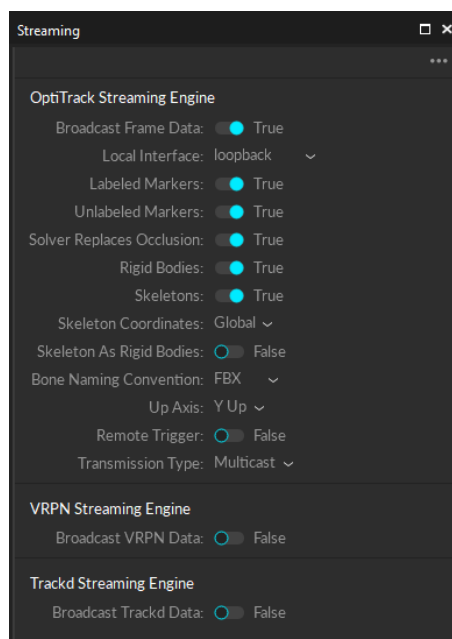
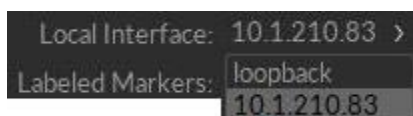
注意：

Broadcast Frame Data 设置为 True。、Rigid Bodies 设置为 True。

Skeleton 设置为 True。

Bone Naming Convention 设置为 FBX。

如果在本机传输数据 Local Interface 设置为 loop back ; 如果在局域网中传输数据 Local Interface 设置为 PC 在局域网中的 IP 地址。



5.2.1.5 OptiTrack Optical 使用

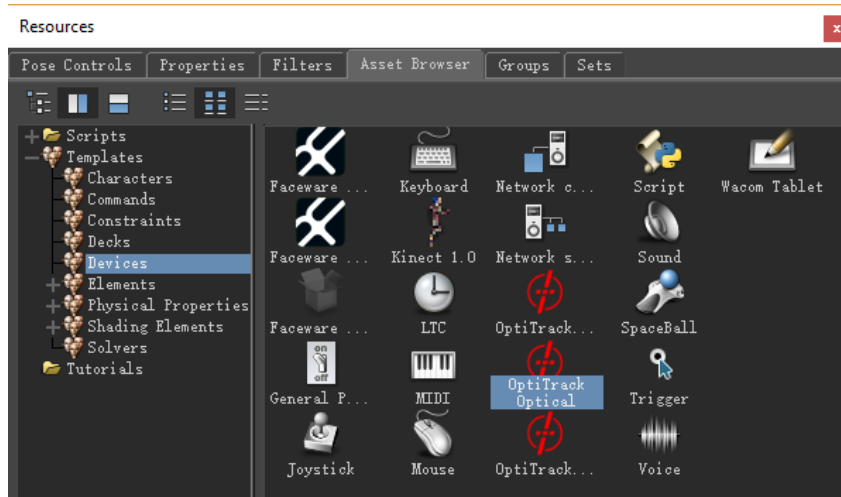
为驱动 MotionBuilder 中人物模型运动需创建如下数据流：

Mocap Marker Data → MotionBuilder Actor → Skeleton Data → MotionBuilder Character。

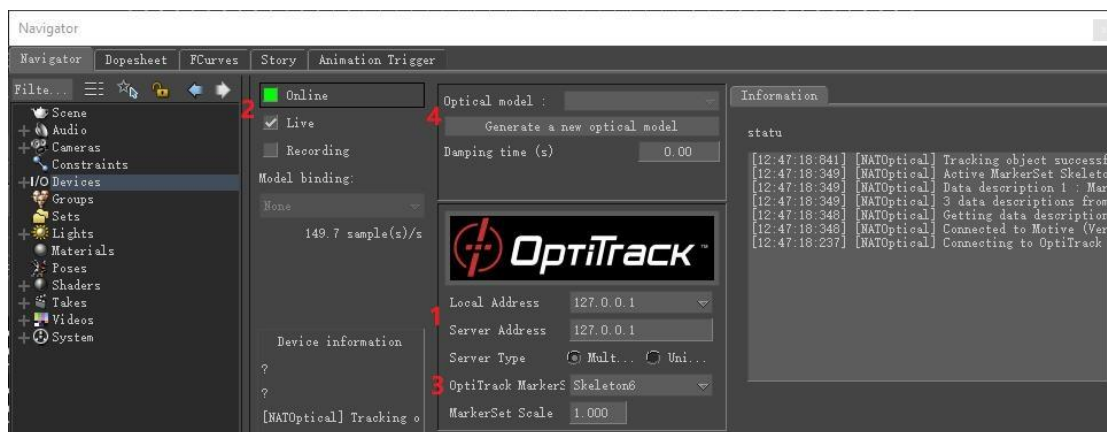
Mocap Marker Data → MotionBuilder Actor 步骤将运动捕捉数据 (Markers) 映射到 MotionBuilder Actor 上。MotionBuilder Actor 是从 Marker Data 创建关节角度的 Skeleton Solver。

MotionBuilder Actor → Skeleton Data → MotionBuilder Character 步骤是 MotionBuilder 特有的，将 MotionBuilder Actor 映射到最终的 Character Skeleton 上。

- 1) MotionBuilder 中，在 Resources → Asset Browser → Templates → Devices 窗口，将 OptiTrack Optical 拖拽到 Viewer 窗口 (主显示界面)。



2) 在 Navigator → I/O Devices → I/O OptiTrack Optical 中。



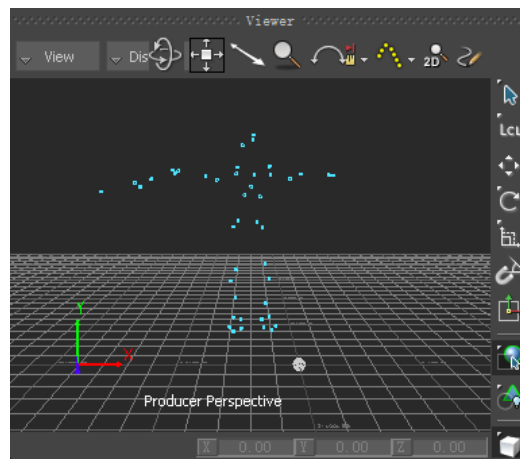
- 设置 Local \ Server Address。如果 Motive 中 Local Interface 设置为 loopback , Local \ Server Address 均设置为 127.0.0.1 ; 如果 Motive 中 Local Interface 设置为局域网 IP 地址 , Local 设置为运行 MotionBuilder 的 PC 的局域网 IP 地址 , Server 设置为运行 Motive 的 PC 的局域网 IP 地址。
- 勾选 Online 以及 Live , 开始接收数据。红色代表网络连接断开 ; ; 黄色代表有网络连接 , 但没有接收到数据 ; 绿色代表成功接收到数据。
- 设置 OptiTrack MarkerSet。建议设置为 All , 传输所有 Asset 的 Marker 点信息。

- 点击 Generate a new optical model / Update the current optical model.

将当前的 MarkerSet 从 OptiTrack 添加/更新到 MotionBuilder Opticals 模型列表。

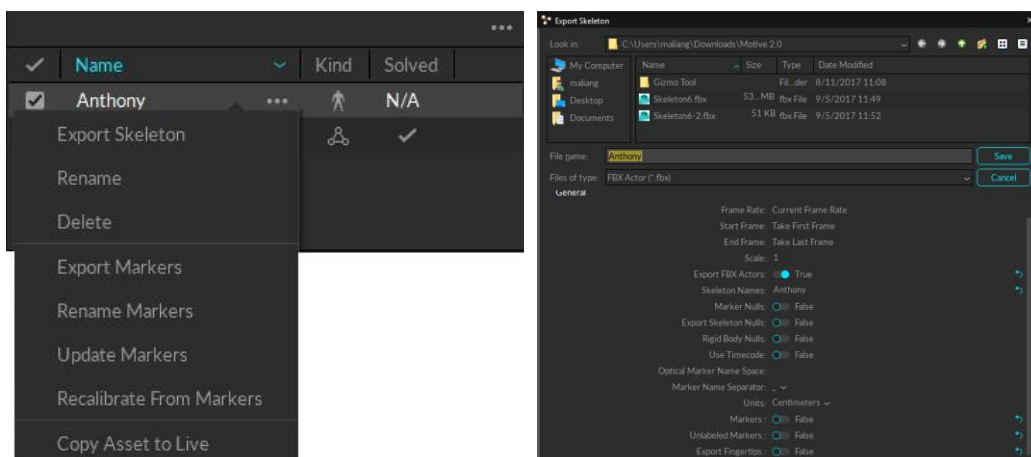
在 Viewer 中会显示接收到的 Marker 点数据信息。

在 MotionBuilder 中 Viewer 视图中会看到由多个方块点构成的人体的数据模型，实时显示时这些数据模块会随着人体的运动而运动，这些点就是在演员身上贴的 Marker 点，每个方块对应一个 Marker 点。

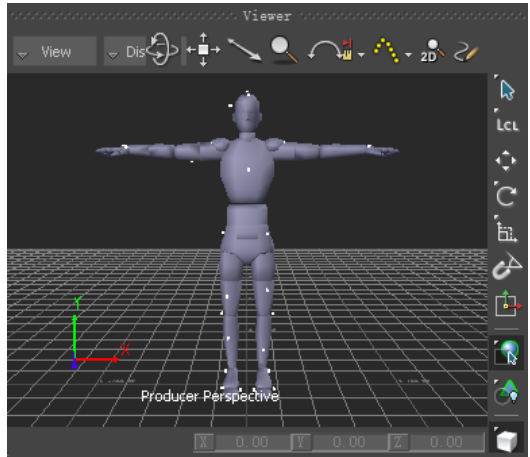


- 3) 将 Motive 中对应的 Skeleton Asset 导出为 FBX 格式，设置如下。

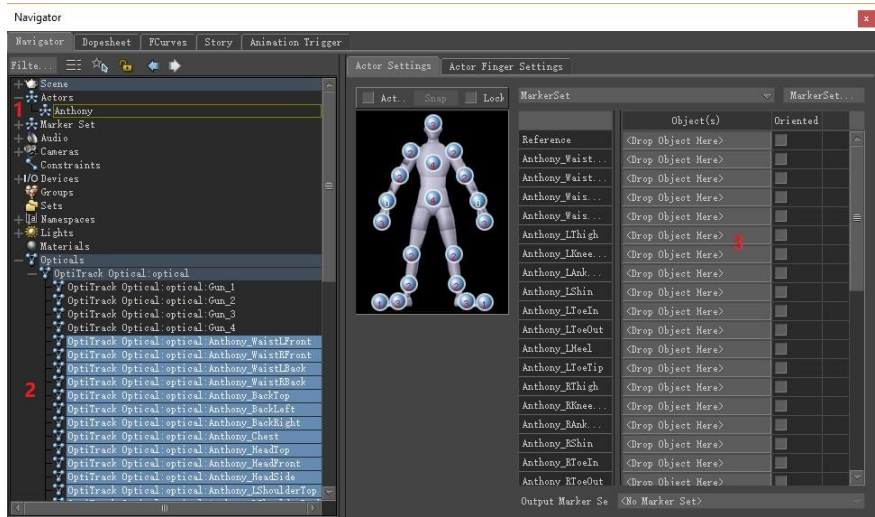
注意：设置 Export FBX Actor 为 True。将 Skeleton Asset 导出为 Actor 资产。



- 4) 在 MotionBuilder 软件中，点击 File → Merge，在弹出的窗口中选择刚导出的 Skeleton Asset，在 Navigator 窗口中 Actors 下会相对应的 Actor 名称。



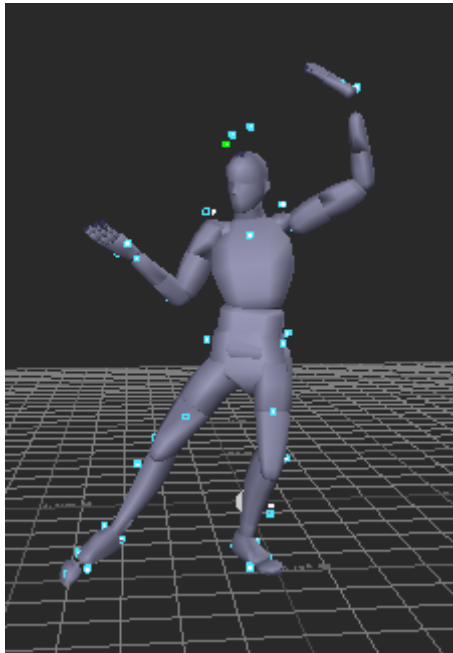
- 5) 在 Navigator 窗口中 展开 Opticals → OptiTrack Optical : optical 以及 Actors , 双击导入的 Actor , 在右侧 Actor Settings 包含的 Markerset , 全选在 Opticals 中对应改 Actor 的 Markerset , 拖拽到 Object 区域<Drop Object Here>。



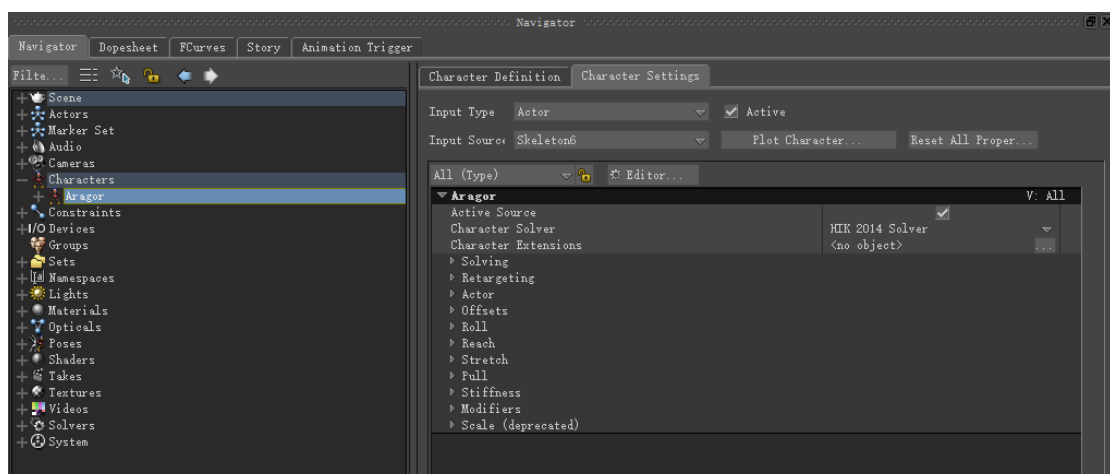
使得 Object 中的名称与左侧名称一致，如下图所示。

	Object(s)
Reference	<Drop Object Here>
Anthony_WaistLFront	OptiTrack Optical:optical.Anthony_WaistLFront
Anthony_WaistRFront	OptiTrack Optical:optical.Anthony_WaistRFront
Anthony_WaistLBack	OptiTrack Optical:optical.Anthony_WaistLBack
Anthony_WaistRBack	OptiTrack Optical:optical.Anthony_WaistRBack
Anthony_LThigh	OptiTrack Optical:optical.Anthony_LThigh
Anthony_LKneeOut	OptiTrack Optical:optical.Anthony_LKneeOut
Anthony_LAnkleOut	OptiTrack Optical:optical.Anthony_LAnkleOut
Anthony_LShin	OptiTrack Optical:optical.Anthony_LShin
Anthony_LToeIn	OptiTrack Optical:optical.Anthony_LToeIn
Anthony_LToeOut	OptiTrack Optical:optical.Anthony_LToeOut
Anthony_LHeel	OptiTrack Optical:optical.Anthony_LHeel
Anthony_LToeTip	OptiTrack Optical:optical.Anthony_LToeTip
Anthony_RThigh	OptiTrack Optical:optical.Anthony_RThigh
Anthony_RKneeOut	OptiTrack Optical:optical.Anthony_RKneeOut
Anthony_RAnkleOut	OptiTrack Optical:optical.Anthony_RAnkleOut
Anthony_RShin	OptiTrack Optical:optical.Anthony_RShin
Anthony_RToeIn	OptiTrack Optical:optical.Anthony_RToeIn
Anthony_RToeOut	OptiTrack Optical:optical.Anthony_RToeOut

勾选 Actor Settings 中的 Act 选项。即可看到 Viewer 窗口中 Actor 与 Optical Marker 关联在一起，跟随 Optical Marker 一起运动。



- 6) 为 Actor 添加 Character。在 MotionBuilder 软件中，点击 File->Merge，在弹出的窗口中选择要导入的角色模型文件，将角色模型角色化 (Characterize)，在 Navigator → Characters 中双击完成角色化的角色模型，Character Settings 中，设置 Input Type 为 Actor，Input Source 设置为对应的演员名称，勾选 Active。

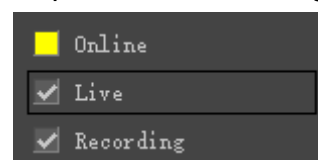


然后即可看到在 Viewer 窗口中。导入的人物角色模型与 Actor 绑定在一起，跟随 Actor 一起运动。



角色化 (Characterize) 操作会在后面介绍。

- 7) 录制数据。Navigator → I/O Devices → I/O OptiTrack Optical 中选 Recording 选项，然后点击 Transport Controls 下的播放按钮。



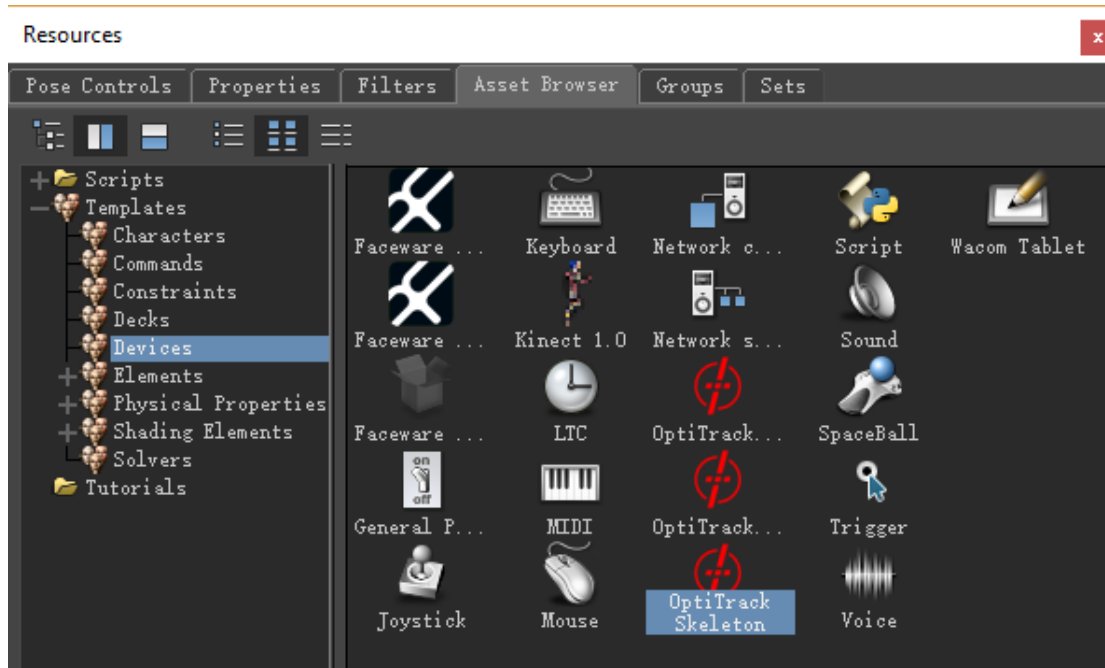
- 8) 回放录制数据。Navigator → I/O Devices → I/O

OptiTrack Optical 中取消勾选 Live 选项，然后点击 Transport Controls 下的播放按钮。

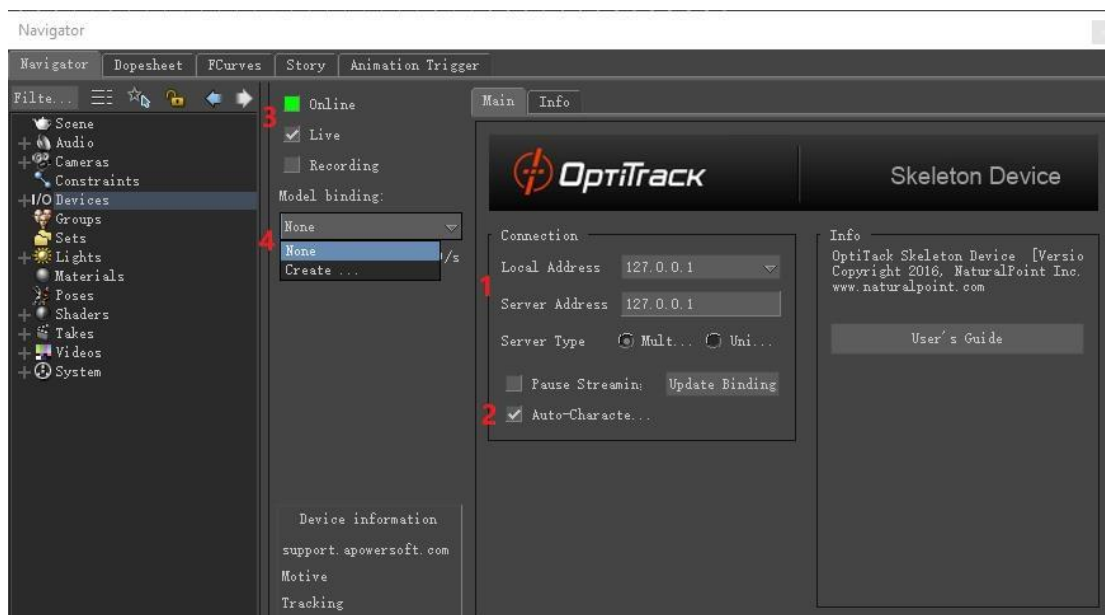
注意：如果是已经记录的数据，请现在 Motive 中播放然后再进行 MotionBuilder 中的操作。

5.2.1.5 OptiTrack Skeleton 使用

- 1) MotionBuilder 中，在 Resources → Asset Browser → Templates → Devices 窗口，将 OptiTrack Skeleton 拖拽到 Viewer 窗口 (主显示界面)。



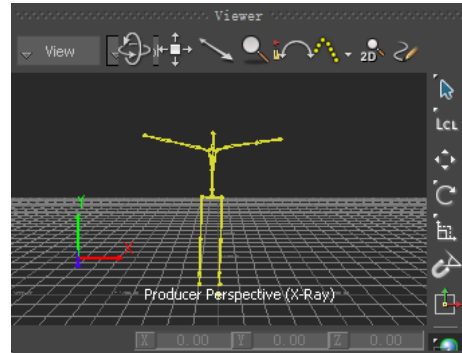
2) 在 Navigator → I/O Devices → I/O OptiTrack Optical 中。



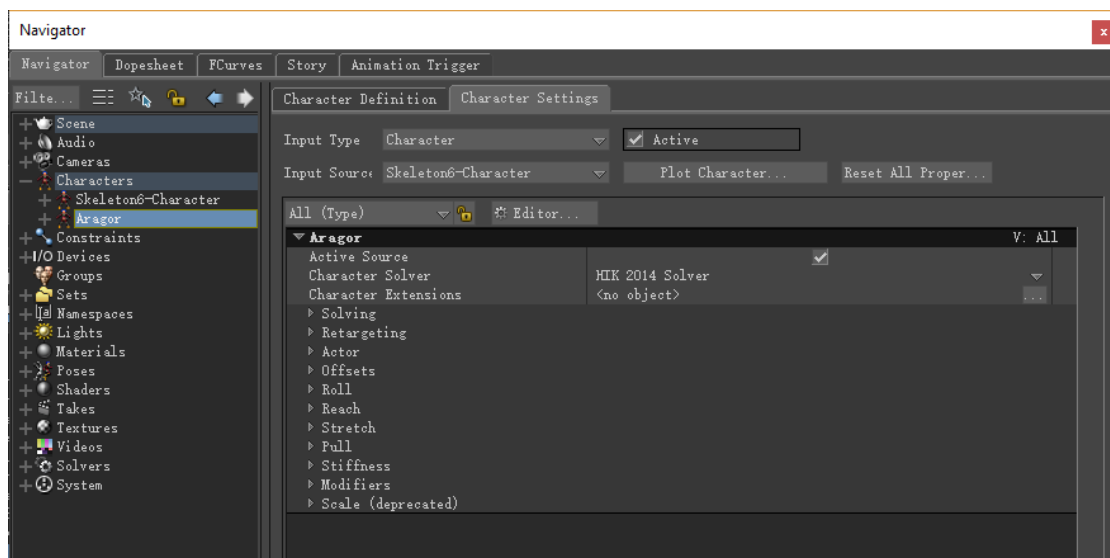
- 设置 Local \ Server Address。如果 Motive 中 Local Interface 设置为 loopback , Local \ Server Address 均设置为 127.0.0.1 ; 如果 Motive 中 Local Interface 设置为局域网 IP 地址 , Local 设置为运行 MotionBuilder 的 PC 的局域网 IP 地址 , Server 设置为运行 Motive 的 PC 的局域网 IP 地址。

- 勾选 Auto-Character 选项。
- 勾选 Online 以及 Live，开始接收数据。红色代表网络连接断开；黄色代表有网络连接，但没有接收到数据；绿色代表成功接收到数据。
- 点击 Model Binding 下 Create 选项。

在 Viewer 中会显示接收到的角色化后的演员 Skeleton 数据信息，同时会在 Navigator → Characters 显示该 Skeleton 角色化后的名称。



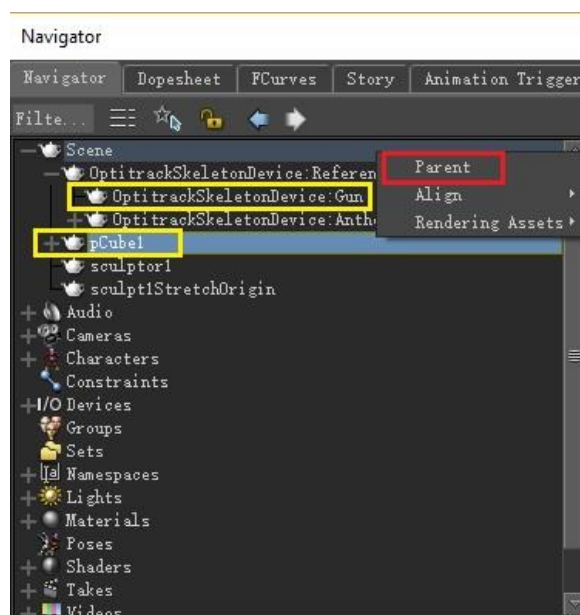
- 3) 添加人物角色模型。在 MotionBuilder 软件中，点击 File->Merge，在弹出的窗口中选择要导入的人物角色模型文件，将角色模型角色化 (Characterize)，在 Navigator → Characters 中双击完成角色化的角色模型，Character Settings 中，设置 Input Type 为 Charecter，Input Source 设置为对应的 Skeleton 角色化后的名称，勾选 Active。



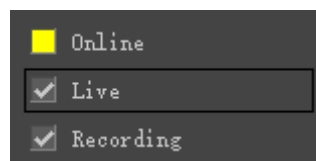
然后即可看到在 Viewer 窗口中。导入的人物角色模型与 Skeleton 绑定在一起，跟随 Skeleton 一起运动。



- 4) 添加道具模型。在 MotionBuilder 软件中，点击 File->Merge，在弹出的窗口中选择要导入的道具模型文件，在 Navigator → Scene 中选中道具模型，推拽到 OptiTrack Skeleton Device 下刚体资产上(与 Motive 中 Rigid Body 名称相同)，选择 Parent。在 Viewer 窗口中将道具模型的质心与刚体的质心移动到一起。



- 5) 录制数据。Navigator → I/O Devices → I/O OptiTrack Optical 中选 Recording 选项，然后点击 Transport Controls 下的播放按钮。



- 6) 回放录制数据。Navigator → I/O Devices → I/O OptiTrack Optical 中取消勾选

Live 选项，然后点击 Transport Controls 下的播放按钮。

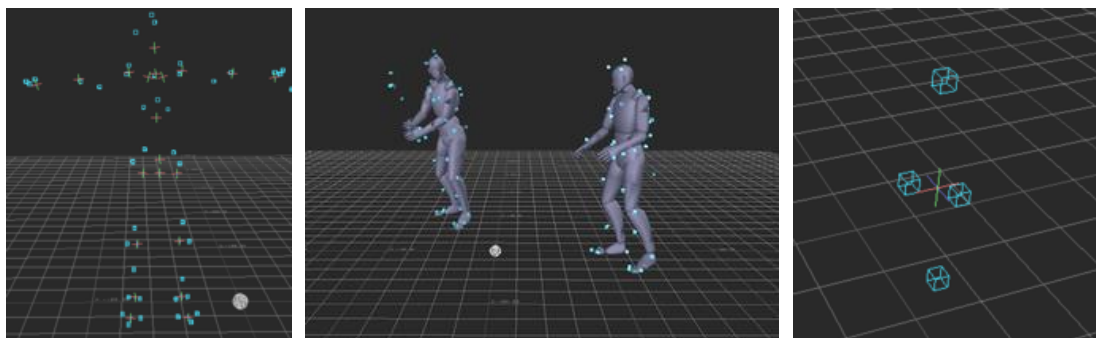
注意：如果是已经记录的数据，请现在 Motive 中播放然后再进行 MotionBuilder 中的操作。

5.2.2 已记录数据应用

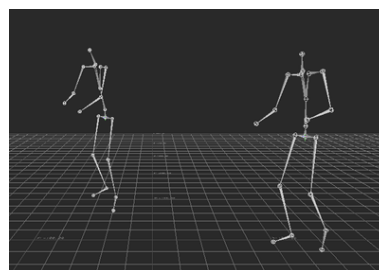
5.2.2.1 MotionBuilder 内数据应用

Motive :Body 可以将跟踪数据导出为 FBX 文件，FBX 文件有两种类型：Binary FBX、ASCII FBX，FBX 格式的数据文件常用于 MotionBuilder 中。

导出的 ASCII 格式的 FBX 文件可以包含重建的每个 Marker 点的坐标数据以及每个涉及的 Asset 的 6 自由度数据，取决于导出设置。可以通过文本编辑器打开编辑 ASCII 文件。



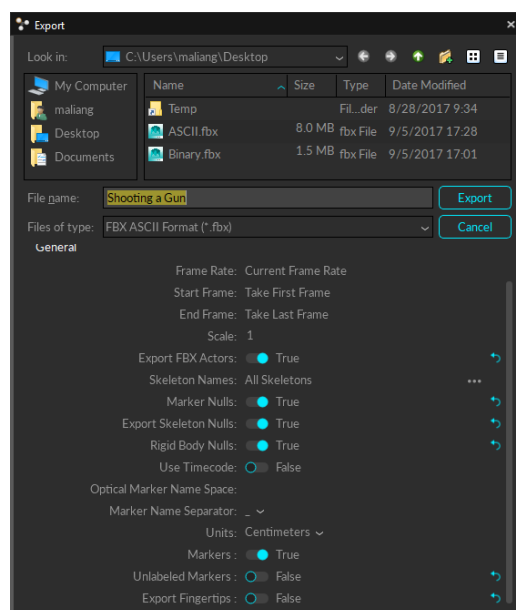
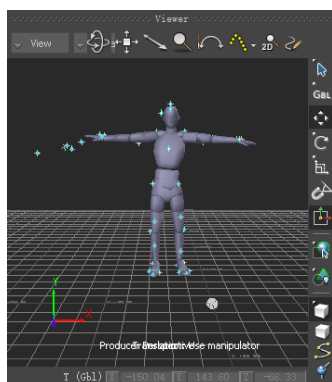
二进制 FBX 文件比 ASCII FBX 文件更精简。此文件类型中不包含重建的 3D Marker 数据，但通过保存对应的关节角度和段长度导出选定的骨架。刚体将会输出起始处的位置和方向。



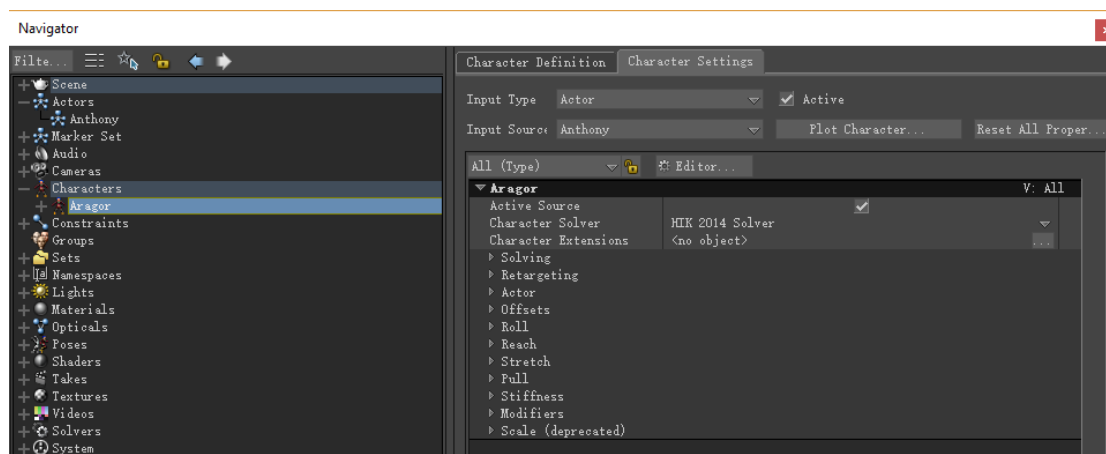
1) ASCII 数据应用

- Motive 中导出 ASCII 格式的 FBX 文件，导出时参数设置如图所示。

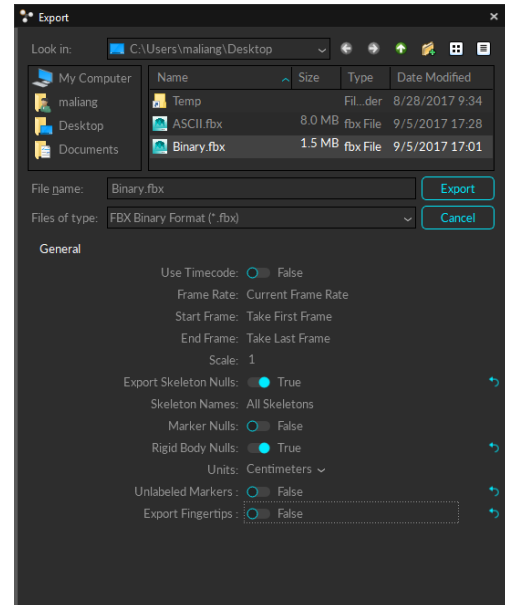
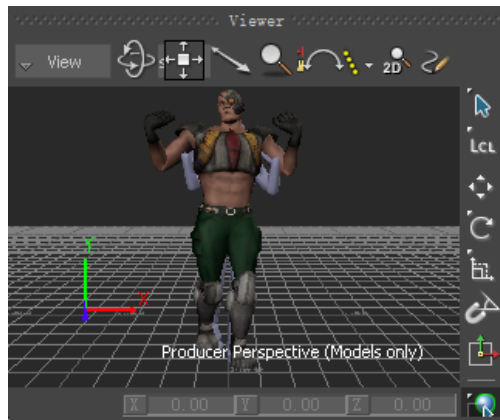
- MotionBuilder 导入 ASCII 格式的 FBX 文件。在 MotionBuilder 软件中，点击 File → Merge，在弹出的窗口中选择刚导出的 ASCII 格式的 FBX 文件。



- 为 Actor 添加 Character。在 MotionBuilder 软件中，点击 File->Merge，在弹出的窗口中选择要导入的角色模型文件，将角色模型角色化 (Characterize)，在 Navigator → Characters 中双击完成角色化的角色模型，Character Settings 中，设置 Input Type 为 Actor，Input Source 设置为对应的演员名称，勾选 Active。

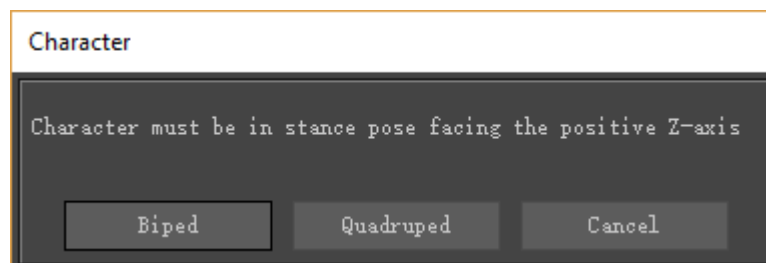
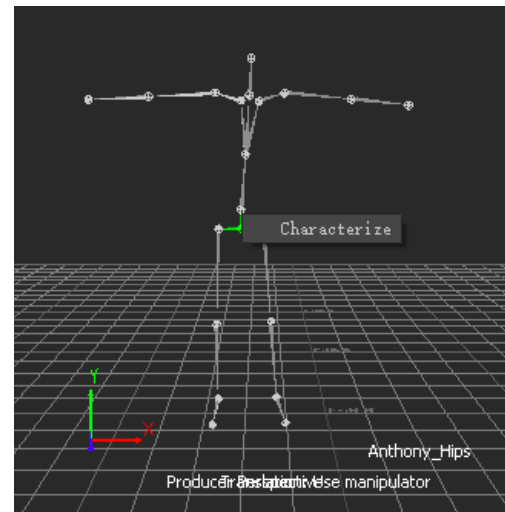


然后即可看到在 Viewer 窗口中。导入的人物角色模型与 Actor 绑定在一起，跟随 Actor 一起运动。



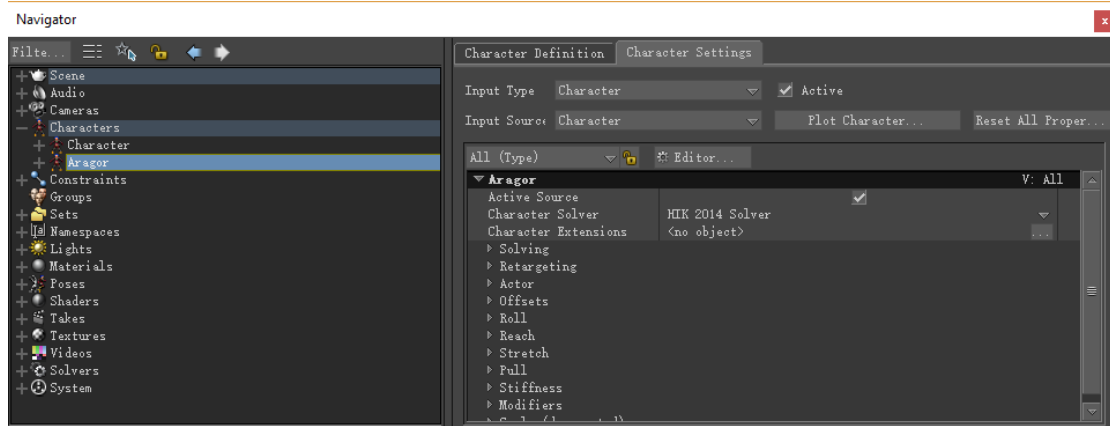
2) Binary 数据应用

- Motive 中导出 Binary 格式的 FBX 文件，导出时参数设置如图所示。
- MotionBuilder 导入 Binary 格式的 FBX 文件。在 MotionBuilder 软件中，点击 File → Merge，在弹出的窗口中选择刚导出的 Binary 格式的 FBX 文件。
- 将 Skeleton 数据 Characterize。将 Resources → Template → Characters 中，拖动 Character 图标到 Viewer 视图中 Skeleton 中的 Hips 上，点击 Characterize。在弹出的窗口中选择 Biped。



- 添加人物角色模型。在 MotionBuilder 软件中，点击 File->Merge，在弹出的窗口中选择要导入的人物角色模型文件，将角色模型角色化 (Characterize)，在

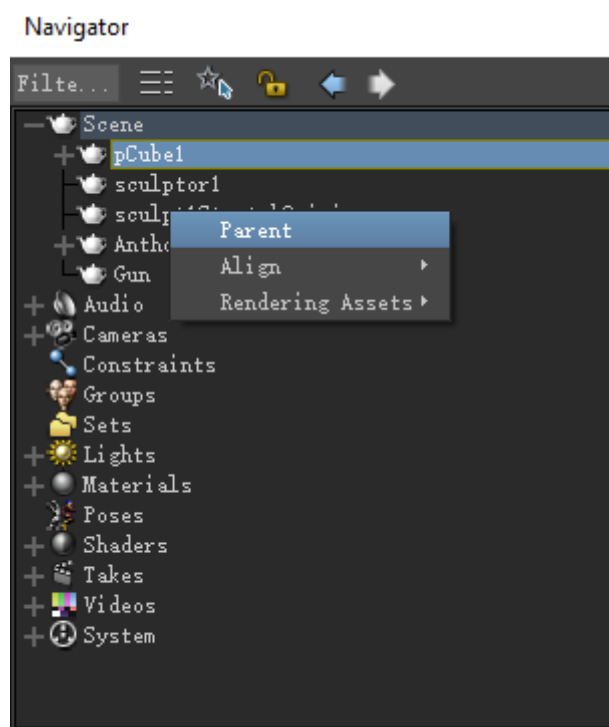
Navigator → Characters 中双击完成角色化的角色模型 ,Character Settings 中 ,
设置 Input Type 为 Charecter , Input Source 设置为对应的 Skeleton 角色化后
的名称 ,勾选 Active。



然后即可看到在 Viewer 窗口中。导入的人物角色模型与 Skeleton 绑定在一起 ,
跟随 Skeleton 一起运动。



- 添加道具模型。在 MotionBuilder 软件中 , 点击 File->Merge , 在弹出的窗口中选择要导入的道具模型文



件 ,在 Navigator → Scene 中选中文具模型 ,推拽到 OptiTrack Skeleton Device
下刚体资产上 (与 Motive 中 Rigid Body 名称相同) ,选择 Parent。在 Viewer 窗

口中将道具模型的质心与刚体的质心移动到一起。

5.2.2.2 3ds Max 内数据应用

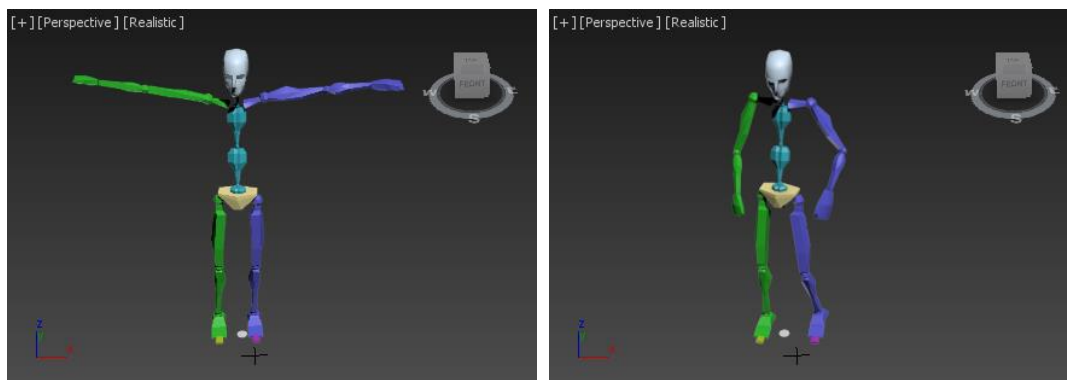
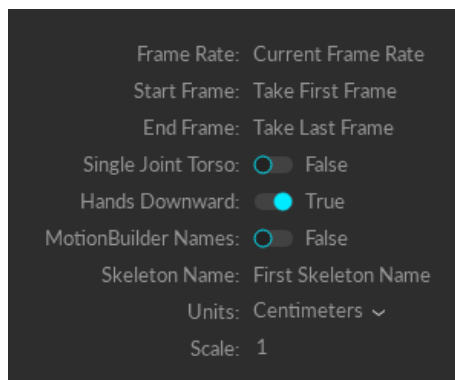
Motive 可以将跟踪数据导出为 BVH (Bio Vision Hierarchy) 文件。BVH 文件中不包含单个 Marker 点的数据。采用分层式骨骼段关系导出选定的 Skeleton，即在 BVH 文件中，导出主要骨架段 (Hips) 的位置信息，并通过关节角度和骨骼段参数来记录后续骨骼的数据。

每个 BVH 文件只导出一个骨架，包含了在其他软件中表征骨架所需的基本定义。BVH 文件可以在 3ds Max

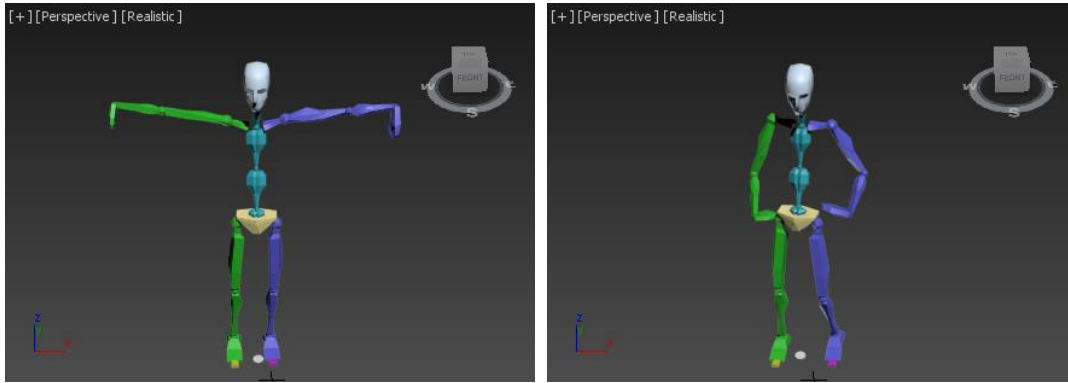
- 1) BVH 文件可以导入到 Autodesk 3ds Max 中，数据导出时，需做如图所示设置才能保证数据在 3ds Max 正常使用。

需注意两点：

- Hands Downward 需设置为 True。如果设置为 False，在 3ds Max 中 Skeleton 的手部动作将会没有姿态信息，如下图所示：



Hands Downward 设置为 True

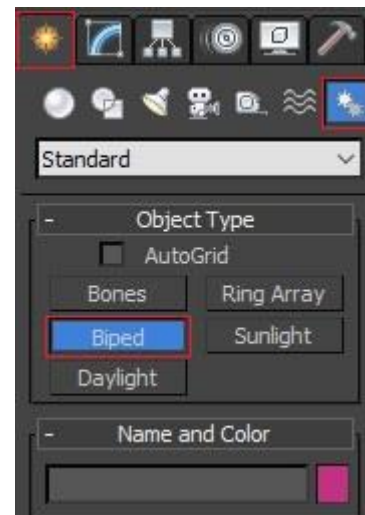
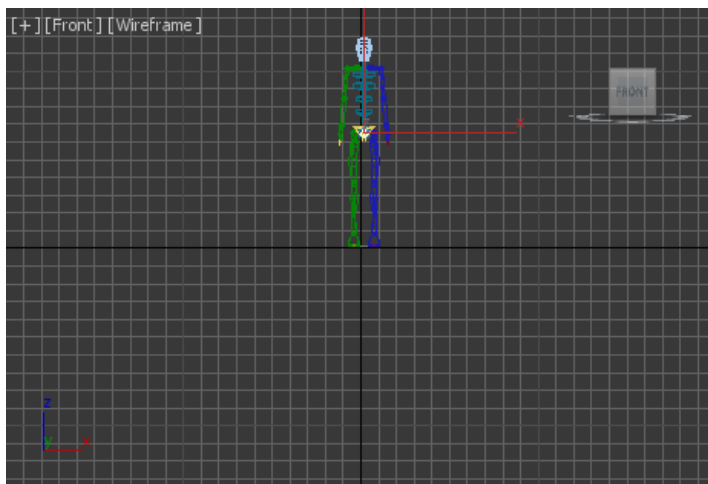


Hands Downward 设置为 False

- MotionBuilder Names 需设置为 False。如果设置为 True , BVH 数据文件会因命名方式不同不能导入到 3ds Max。

2) 3ds Max 中创建 Skeleton。切换到如图所示界面，点击图中 Biped。

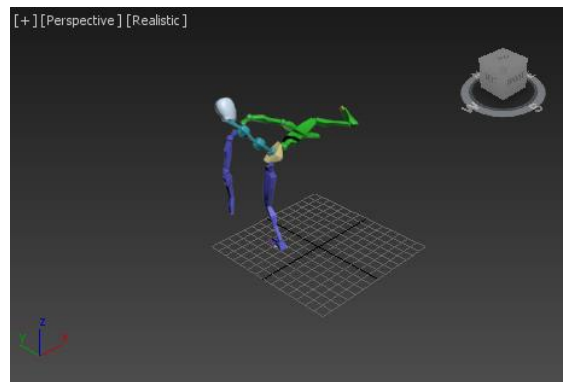
在 Front 视图中点击鼠标左键，并拖动，创建如下的 Skeleton。



3) 载入 Motion Capture 数据文件。选中 Skeleton 的 Hips , 切换到如下界面，点击下图
中 Load Motion Capture Files。



在弹出的窗口中选择刚导出的 BVH 文件。即可将数据加载到 Skeleton 上，驱动 Skeleton 运动。



5.2.3 MotionBuilder 角色化

5.2.3.1 角色动画流程

MotionBuilder 基于实时架构、使用智能角色动画技术来表现角色动画，其主要的工作流程包括下面三个方面：

- 角色导入
- 角色化处理

- 设置角色驱动方式

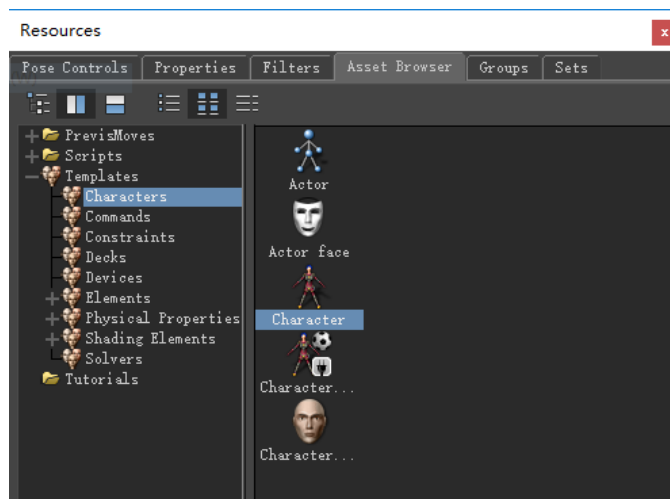
首先，在角色导入方面。由于 MotionBuilder 没有建模能力，因此角色及其骨架创建通常都是在其它的外部软件(比如 Lightwave，Maya 和 3dsMax 等)中完成，然后再通过 FBX 插件对角色进行导入处理。

其次，在角色导入场景以后，为了让角色骨架能够和 MotionBuilder 的角色动画系统产生匹配连接，需要使用系统内置的角色模板来对角色骨架进行角色化处理。在这里，角色模板实际上就是系统基于自身角色动画系统缺省定义的一套骨骼节点。而角色化处理，就是系统连接角色模板上的缺省节点到角色骨架的过程。在这个过程中，系统能够自动根据角色模板上各个节点的命名来检查角色骨骼。如果角色上各骨骼的命名都能够和角色模板上所定义的节点名称对应，则导入的角色骨架可以正确的和 MB 角色动画系统相连。

最后，在角色骨架完成角色化处理后，系统提供几种角色驱动方式来设置角色动画。比如，可以使用骨骼控制装置操控角色，使用数据映射方式映射运动捕捉数据到角色等。

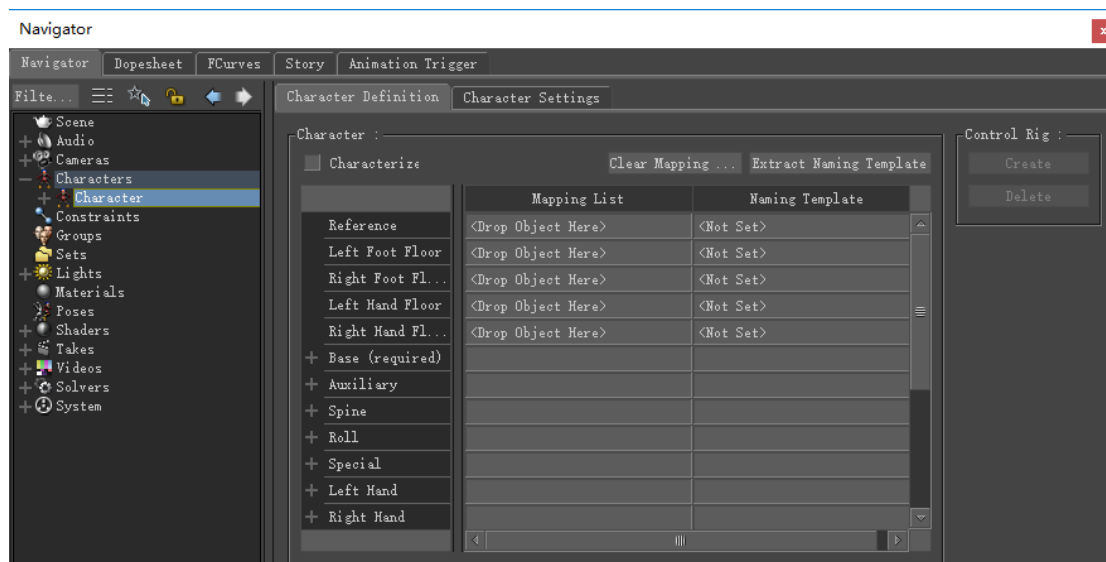
5.2.3.2 角色模板

Layout→Editing 模式下，角色模板位于 Resources → Template → Characters 文件夹中，如图所示：

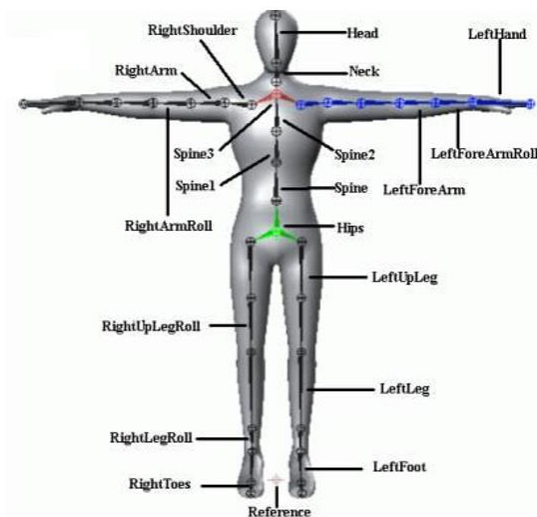


要使用角色模板，可以在文件夹里选择 Character 图标后，按住鼠标左键拖动到场景视图中。这样，系统就会在 Navigator → Characters 元素类型里自动生成命名为 Character 的角色模板设置。同时，对应的模板设置显示在面板右边。

如果要观看角色模板上的各骨骼节点定义，可以在面板上切换到 Character Definition 选项卡。如下图所示，在选项卡中，模板上的所有节点定义都显示在节点列表窗口中。当系统没有进行角色化处理时，在节点列表右边的 Model 模型映射栏中，所有节点对应的角色骨骼都显示为<No Model>。反之，则 Model 栏里会显示和模板上的节点相应的角色骨骼或者物体。



任何时候，在为角色绑定骨架时，最好都应根据角色模板的这套骨骼节点定义来配置骨架结构和为骨骼命名。但这并不意味着 MotionBuilder 会强迫用户创建骨架时就要完全包含模板上所定义的骨骼节点。与之恰好相反的是，可以根据需要来为角色配置骨架，比如，如下图所示，图中显示的角色骨架也能够正确进行角色化处理，每个骨骼都使用角色模板中相应节点的命名原则。然而，由于角色模型没有手指部分，可以不用为手指部分创建相应的骨骼。那么，在模板中也不用指定和手指节点相对应的骨骼。关于这些节点和角色骨骼的对应关系，可以参考下面的说明：



- Reference (参考节点) : 定义角色骨架层级的根部节点，可以控制整个角色层级的位移，旋转等运动变换。

- Left/Right Foot Floor(脚部地面节点)和 Left /Right Hand Floor(手部地面节点) : 指定场景地面。这影响到角色在动画过程中，手和脚在接触地面时的反应。如果没有为它们进行任何指定，系统使用 Y 轴零点位置作为地面。

- Base(基本节点) : 是必须为进行指定的角色骨骼节点设置。主要包括 Hip(臀部骨骼)，LeftUpLeg/RightUpLeg(左 / 右大腿骨骼)，LeftLeg/RightLeg(左 / 右小腿骨骼)，LeftFoot/RightFoot(左/右脚骨骼)，Spine(脊椎)，LeftArm/RightArm(左/右手上臂骨骼)，LeftForeArm/RightForeArm(左/右手前臂骨骼)，LeftHand/RightHand (左/右手骨骼) 和 Head(头部骨骼)。

- Auxiliary(辅助节点) : 增强角色肩部，脚趾和手指等部位运动的控制骨骼节点。主要包括 LeftToeBase/RightToeBase(左/右脚趾控制骨骼)，LeftShoulder/RightShoulder(左/右肩控制骨骼)，Neck(颈部控制骨骼)，LeftFingerBase/RightFingerBase(左/右手指控制骨骼)。

- Spine (脊椎节点) : 定义角色脊椎骨骼，最多可以设置 10 个。为了让脊椎运动产生最好的效果，应该在绘制角色脊椎骨骼时让每根骨骼都具有相同长度。

▪ Roll(转动节点) : 定义角色手臂，腿部用于产生扭动效果的支关节。在为角色创建骨骼时，这些关节最好能够同时创建。这样，通过控制相应的支关节，可以为角色的手臂和腿部增加一些真实的肌肉扭动效果。对应骨骼主要包括 LeftUpLegRoll / RightUpLegRoll (控制大腿扭动)，LeftLegRoll / RightLegRoll (控制小腿扭动)，LeftForeArmRoll / RightForeArmRoll (控制手臂前臂扭动)，LeftArmRoll / RightArmRoll (控制手臂上臂扭动)等。

▪ Left Hand / Right Hand(左右手指节点) : 角色包含手指动画控制时，定义手指各关节。在列表中每个手指都有 4 个节点来定义相应的骨骼，节点名称的每一个数值和相应的手指关节对应。比如标记为 1 和 3 的数值代表手指的第一个关节和最后一个关节。但要注意，最后一个节点定义的是手指末端骨骼，而不是手指的末端关节。

▪ Left Foot / Right Foot(左右脚趾节点) : 和手指节点类似，定义左右脚脚趾关节

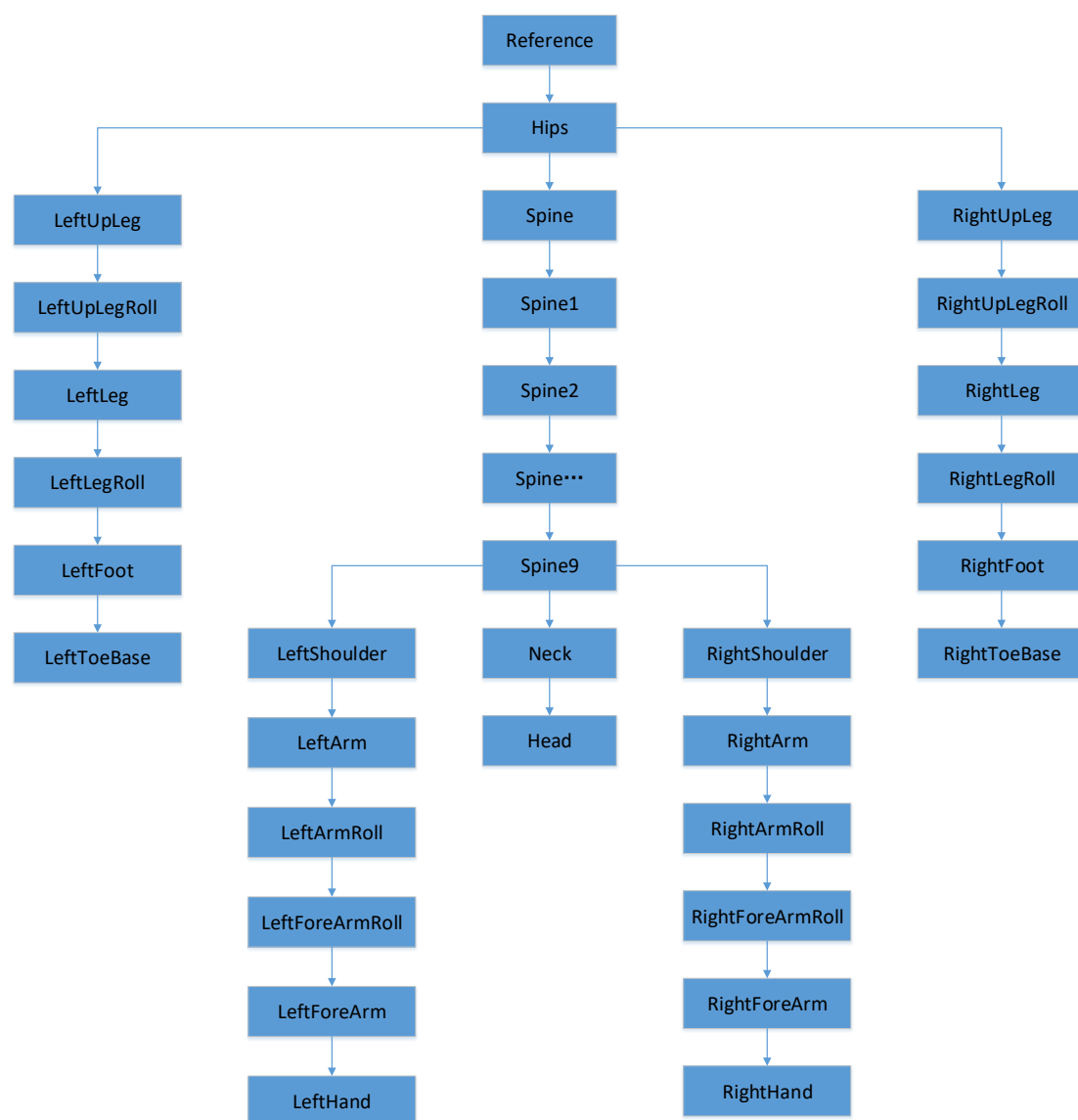
▪ Left In-Hand / Right In-Hand 和 Left In-Foot/Right In-Foot : 通常用于定义动物角色的爪部，连接各个指关节和 Finger / Toe Base 骨骼的附加骨骼，如下图 ABC 所示。



▪ Neck (颈椎节点) : 控制角色颈部运动的骨骼节点。通常两足动物的颈椎骨骼只有一个，而四足动物可包含更多的颈椎骨骼定义。

▪ Props (道具节点) : 为角色增加 5 个附加节点去控制角色的道具使用。比如，像武器等角色可以使用的物体。

注意上面列举的节点定义，各节点间也具有相应的层级关系。如下图显示角色在不包括手指骨骼节点时各个节点间的层级关系。另外，如果是动物角色设置，可以把四足动物的前腿当作角色的手，把四足动物的后腿当作角色的腿，并且这些部位的骨骼命名和层级结构同样使用和角色定义相同的原则。



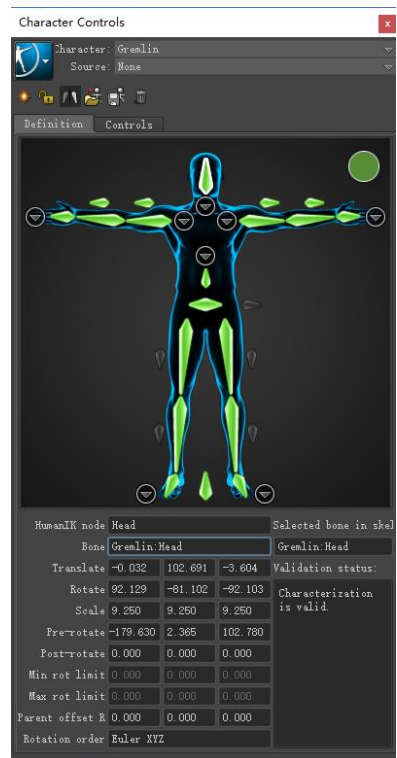
5.2.3.4 角色化

在进行角色化之前，需要确认角色是否符合要求，默认情况下，两足角色的静止姿势应

该为 T-Pose，具体表现为角色面向视图 Z 轴向正极，手臂伸展和地面平行，脚部和地面垂直相交，脊椎伸直。四足动物角色的静止姿势应该表现为面向视图 Z 轴向正极的同时，前后腿以及所有脚趾都沿着地面处于完全伸展状态。确保角色符合要求之后，才能进行角色化。

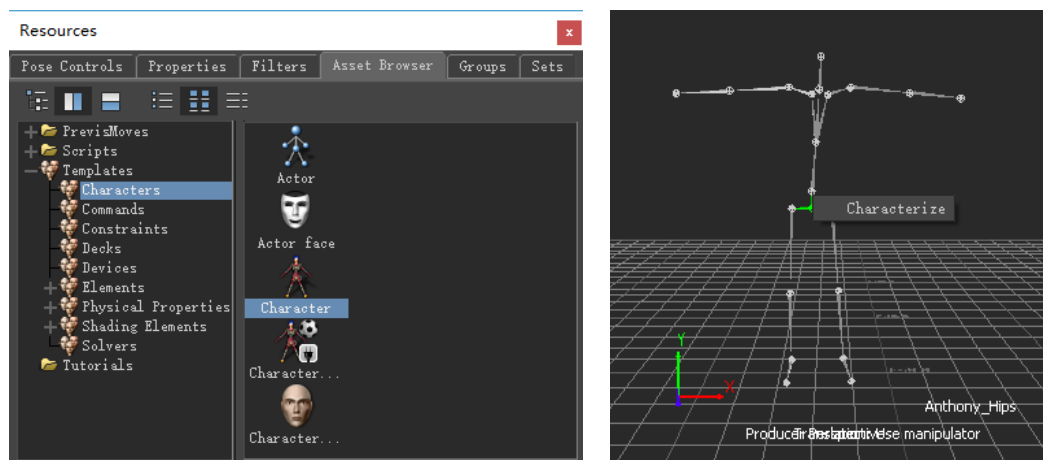
对角色骨架进行的角色化处理可以分为自动和手动两种方法来。大多数情况下，使用自动角色化处理，简便、快捷。

完成角色化之后在 Character Controls->Definition 中，应如图所示



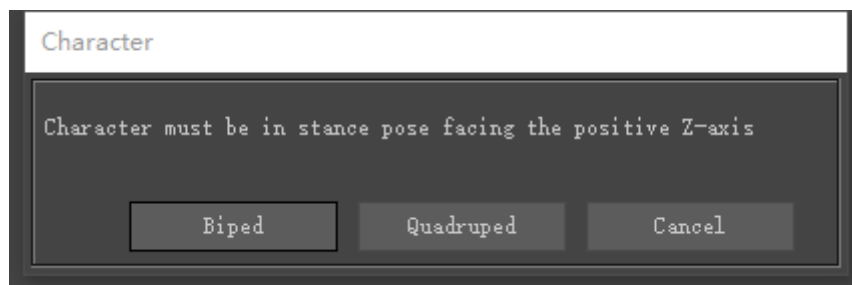
1. 自动角色化

在自动角色化处理中，操作步骤将 Resources → Template → Characters 中，拖动 Character 图标到 Viewer 视图中角色骨骼 Hips 上，从打开菜单中选择 Characterize 角色化选项，如下图所示。然后，系统根据当前角色的骨架结构和骨骼命名，可能出现下面两种情况：



如果角色骨骼使用角色模板的节点命名原则，系统会弹出一个如下图所示的对话框，提示必须让场景中的角色模型处于静止姿势并面向 Z 轴向正极。同时，在对话框下方，包括

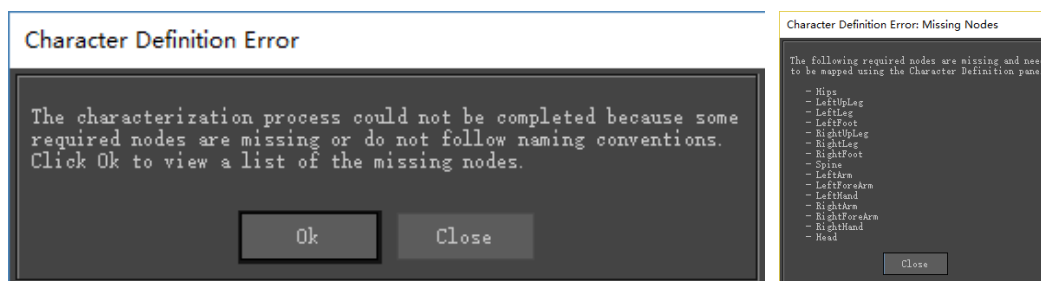
Biped (两足) 或 Quadruped (四足) 两个按钮，可以用来选择是对两足还是四足角色进行角色化处理。



当选择相应的按钮后，系统开始角色化。在角色化的过程中，有时系统也会弹出信息警告框，提示某个骨骼，其旋转轴角度产生偏移。假如不认为该骨骼的旋转偏移会影响角色动画设置时，可以选择忽略 (Ignore) 选项继续让系统进行角色化。

2. 手动角色化

如果角色骨骼没有遵循角色模板的节点命名原则时，则在角色化时系统首先会弹出如下图所示的警告框，提示由于某些骨骼不能和角色模板上的节点相对应，角色化失败。

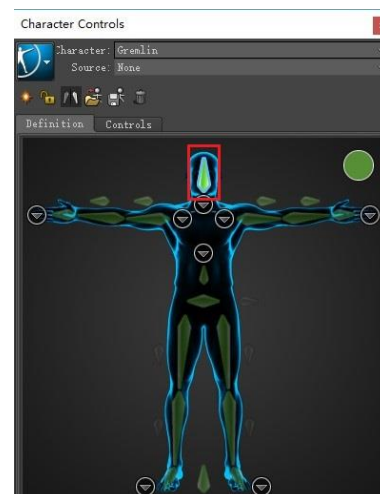


点击 OK 之后，会提示丢失的骨骼节点

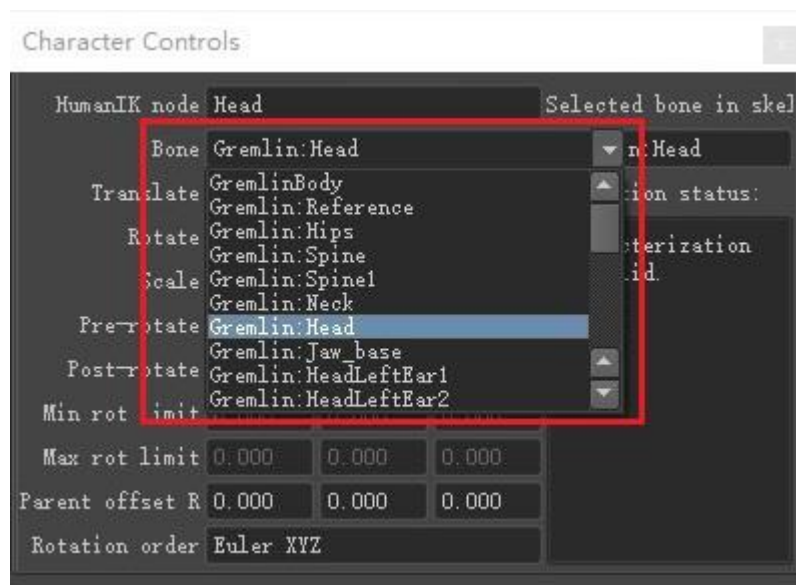
这时，需要使用手动方法来设置角色化。

可以通过以下步骤手动设置角色化

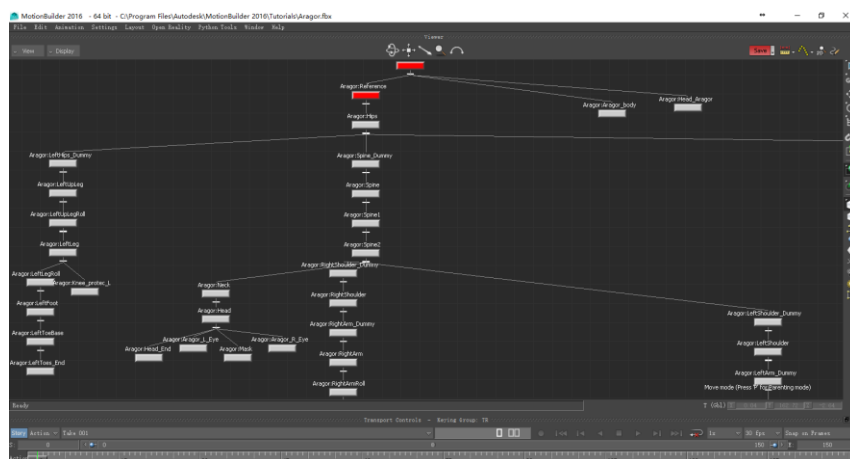
- 1) 打开 character controls，选择对应的 character，在 Definition 窗口，双击未匹配的骨骼（如下图所示），然后点击 Viewer 中相对应的骨骼，即完成匹配，一次完成剩余骨骼。



2) 可以在下方同样可以修改所匹配的骨骼，如下图所示：



3) 在 Viewer 视图下，按下 Ctrl+W 键，切换试图，按照 标准骨骼命名方式重命名骨骼名称，然后重新对模型进行自动角色化。

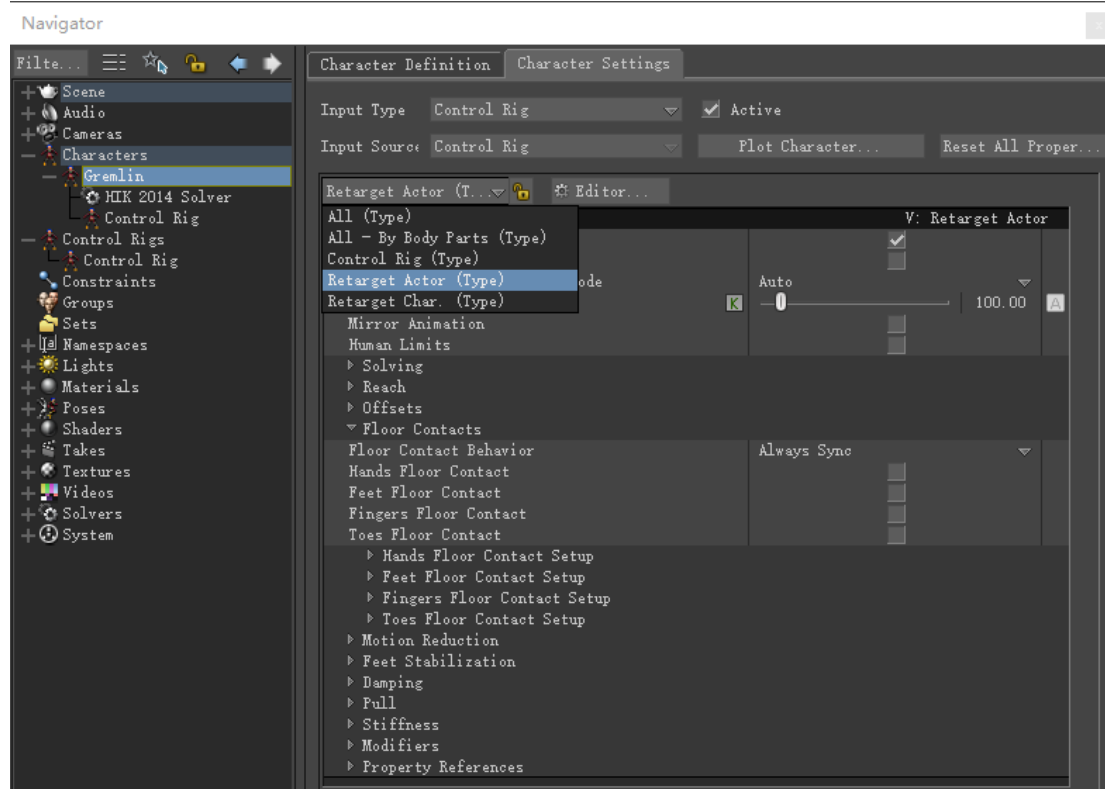


或者也可以按住 Alt 键将重命名之后的骨骼节点拖拽到 Navigator->Character 下 Character Definition 选项卡中对应的骨骼节点中，完成之后勾选 Characterize。

锁定脚

角色做动作时，有时脚会插在地面之下或者浮于空中。为使角色看上去稳定、真实，可进行下面的处理：

Character->Character Settings->Floor Contact->Feet Floor Contact，勾选脚部地面约束选项，使角色在行走过程中计算地面约束影响。

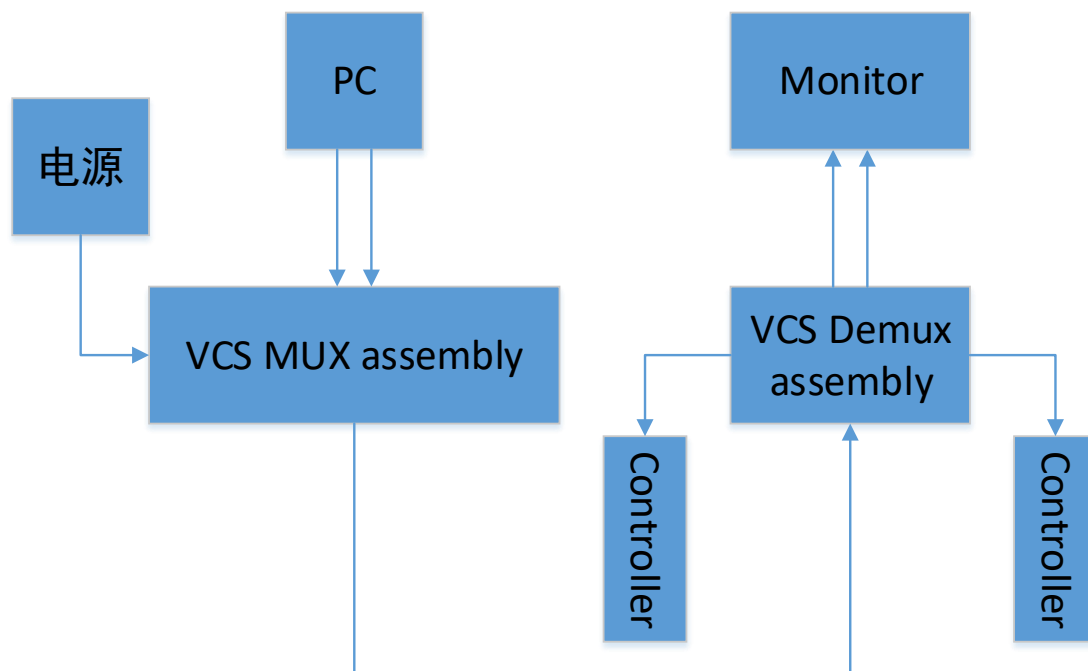


5.2.4 VCS 使用

OptiTrack VCS 用于在 MotionBuilder、Maya 中创建虚拟摄像机。同时监控虚拟摄像机的画面。在影视拍摄过程中可以实时监测拍摄的经过渲染的虚拟画面，方便摄像师取景。

5.2.4.1 VCS 系统硬件连接

VCS 硬件系统硬件连接如下图所示。



- VCS Mux assembly：虚拟摄像机转接盒

一侧包含三个接口：

USB 接口：与电脑 USB 接口连接。

HDMI：与计算机显卡连接。

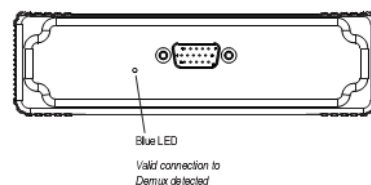
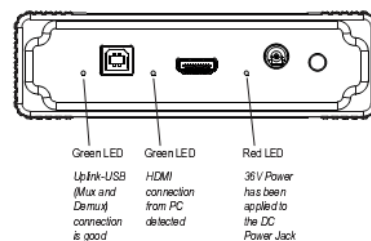
电源接口。

另一侧只有一个 VCS 接口，通过 VCS 线和 VCS

Demux assembly 连接。



VCS Mux assembly



- VCS Demux assembly: VCS 多用转接盒。

转接盒一侧通过 VCS 线缆连接 VCS Mux assembly。

另一侧有四个 USB 接口、一个 HDMI 接口、一个电源接

口

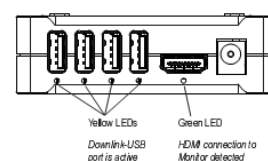
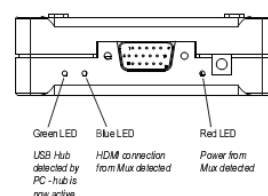
USB 接口：与两个控制器连接。

HDMI 接口：与 Monitor 连接。

电源接口：连接显示器电源。



VCS Demux assembly



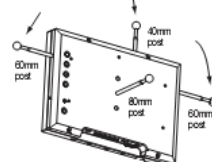
VCS Mux assembly 与 VCS Demux assembly 之间通过一条 15 Pin 线缆连接，该线缆上负载着供电、数字图像传输、USB 控制信号传输三个功能。

- **Controller**：控制器，每个控制器上面有一个方向控制摇杆和 7 个按钮。

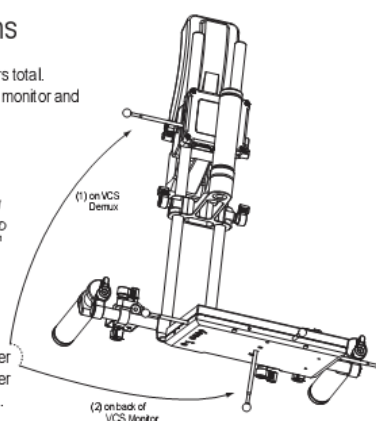
- **Monitor**：监视器，可以监视 PC 机上的内容，Monitor 上面需要安放标志点，标志点的安放如图所示。

* Marker Configurations

The VCS:Pro rig needs four markers total. Place one marker on top of the HD monitor and one on either side.







The fourth (60mm) marker may be placed in either of these two locations.

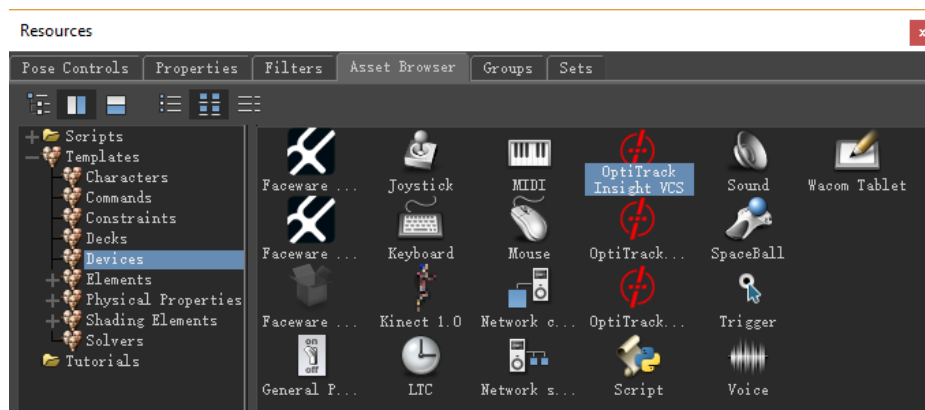


5.2.4.2 VCS 系统软件设置

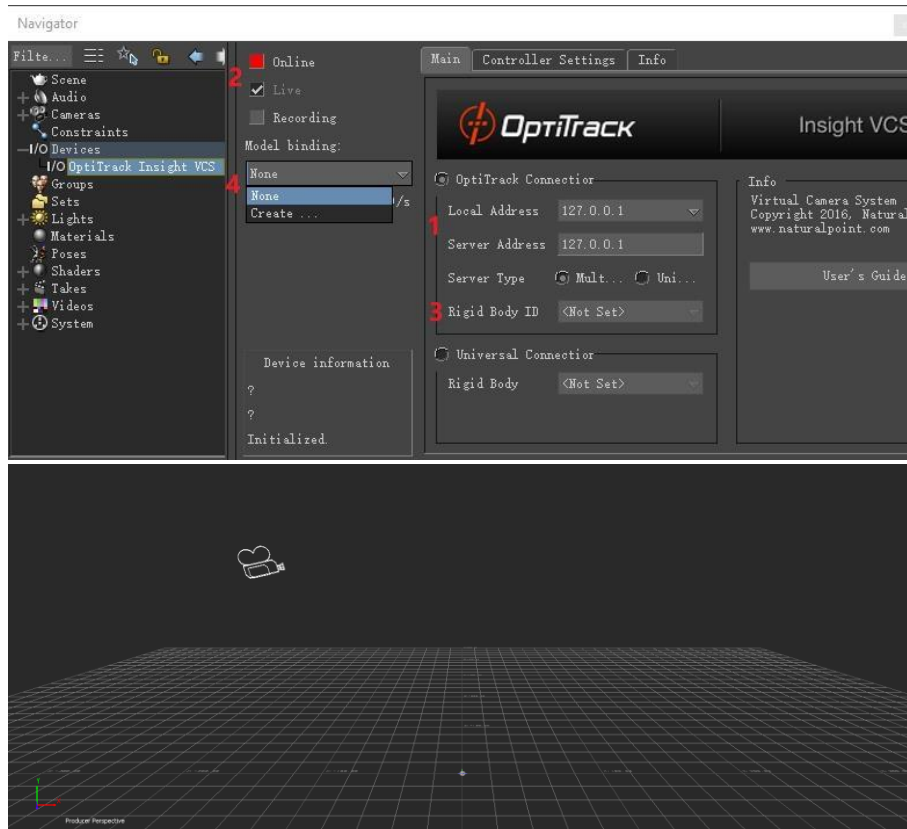
- 1) 在安装的 MotionBuilder Plugin 中包含 VCS 插件。
- 2) MotionBuilder 中 VCS 插件需要有效的 License 才能运行，参考 Motive 激活方式，激活 VCS 插件的 License。
- 3) 在 Motive 中创建代表 VCS 的 Rigid Body，创建刚体时按照下面提示的朝向放置 VCS。

Motive 1.5 (朝向+Z 轴)	Motive 1.7+ (朝向-Z 轴)
	
	

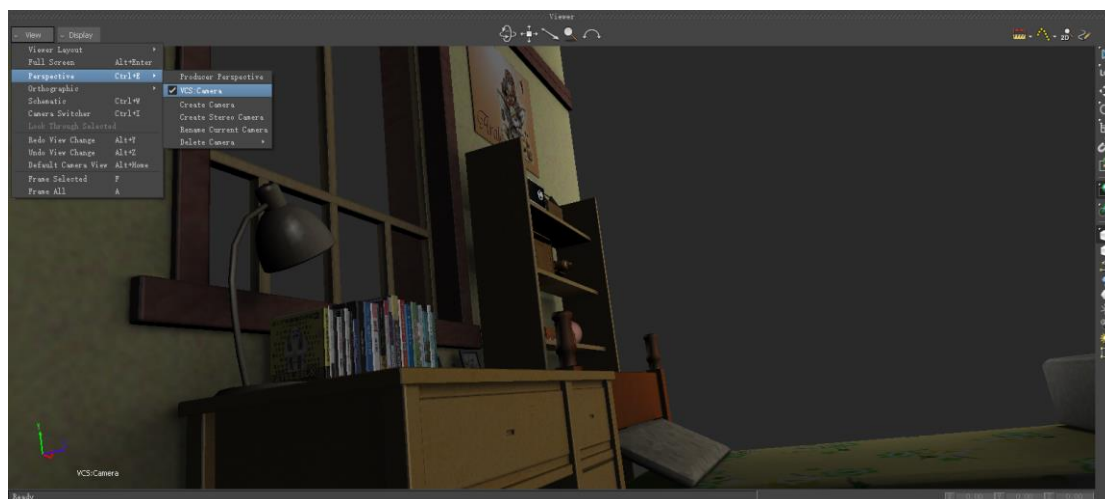
- 4) 在 Motive 中打开 Streaming 窗口。其中 , Rigid Bodies 设置为 True ; 如果在本机传输数据 Local Interface 设置为 loop back ; 如果在局域网中传输数据 Local Interface 设置为 PC 在局域网中的 IP 地址。
- 5) MotionBuilder 中载入虚拟场景。
- 6) MotionBuilder 中 , 在 Resources → Asset Browser → Templates → Devices 窗口 , 将 OptiTrack VCS 拖拽到 Viewer 窗口 (主显示界面) 。



- 7) 在 Navigator → I/O Devices → I/O OptiTrack VCS 的 Main 窗口中。
 - 设置 Local \ Server Address. 如果 Motive 中 Local Interface 设置为 loopback , Local \ Server Address 均设置为 127.0.0.1 ; 如果 Motive 中 Local Interface 设置为局域网 IP 地址 , Local 设置为运行 MotionBuilder 的 PC 的局域网 IP 地址 , Server 设置为运行 Motive 的 PC 的局域网 IP 地址。
 - 勾选 Online 以及 Live , 开始接收数据。红色代表网络连接断开 ; ; 黄色代表有网络连接 , 但没有接收到数据 ; 绿色代表成功接收到数据。
 - 设置 Rigid Body ID , 名称与 Motive 中表征 VCS 的 Rigid Body 名称相同。默认情况下会自动选择 Rigid Body ID , 如果 Motive 中有多个 Rigid Body 需要手动选择 Rigid Body ID。
 - 点击 Model binding 中的 Create。会在 Viewer 中创建虚拟摄像机。

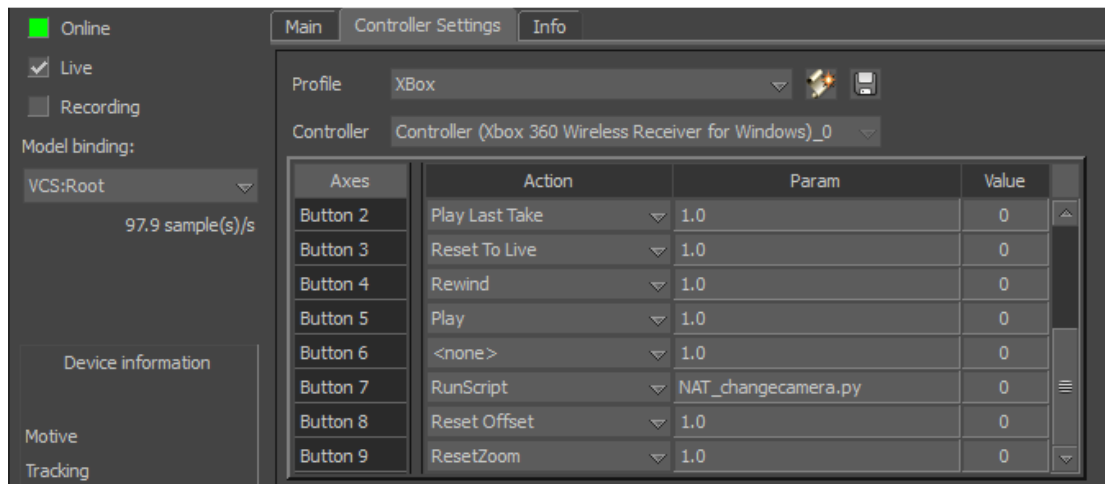


- 8) 选择 Viewer → View → Perspective → VCS : Camera , 即可切换到 VCS 视角 , 查看虚拟摄像机中的内容。



- 9) 在在 Navigator → I/O Devices → I/O OptiTrack VCS 中的 Controller Settings 窗口中 , 可以对 Controller 按钮功能进行设置 , Controller 上面的按钮序号 :

蓝	黑	白	红	底部按钮	摇杆按钮	滚轮按钮
0	1	2	3	4	5	6



5.3 Virtual Reality

OptiTrack 运动捕捉系统可以将捕捉到的 HMD、道具的位置和姿态信息通过插件实时传输到虚拟现实引擎中，虚拟显示引擎根据获得的数据信息驱动场景中道具，并在头显中显示相对应画面。常用的虚拟现实引擎有：Unity 3D、Unreal Engine 4，OptiTrack 针对这两款软件提供相应的插件以获得实时捕捉的运动数据。



5.3.1 Active Marker 跟踪

OptiTrack Active 技术主要应用于虚拟现实应用中，作为被动跟踪的替代方案，Active 有如下好处：

- 搭配 Slim 13E 使用，降低系统成本。
- 可以同时捕捉空间中创建多个相同结构的刚体。
- 通过同步发光技术可以在 Motive 中标记单个



Active Marker。

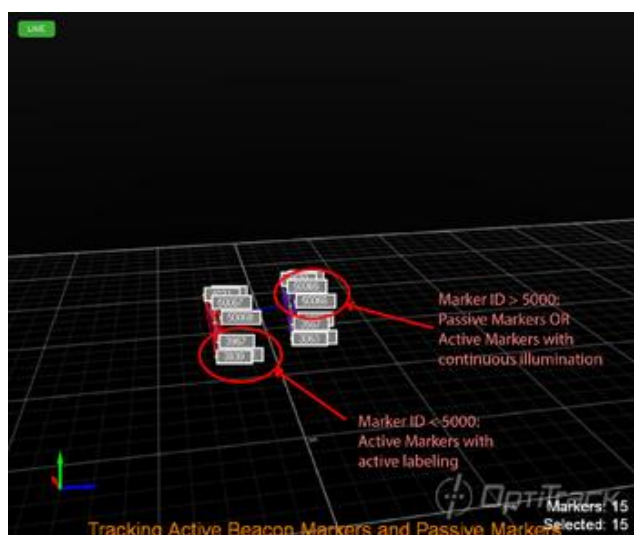
- 能够极大程度降低外界环境对系统造成的干扰。

OptiTrack Active Marker Kit 允许使用 OptiTrack 运动捕捉系统同步跟踪 Active LED Marker。Active Marker Kit 由 Base Station、Tags、Active LED markers 组成，同时需要 OptiTrack 网口相机系统。

完成设置之后，Base Station 将会成为相机系统以及 Active Marker 的同步控制器。通过 RCA 以及射频波发出同步信号，精确同步相机快门与 Active LED Marker 的发光。

注意：Active Marker Kit 需要与网口相机系统（Prime 系列、Slim 13E）配合使用。不支持 USB 相机系统（Flex 系列）。

使用 OptiTrack Active Marker Kit 时，Active Marker 可以通过其独特的发光方式进行单独标记（labeled），单独标记 Active Markers 允许创建多个结构类似的 Rigid Body 资产。Motive 中每个通过 active labeling 方式标记的 Active Marker 的 ID 都小于 5000。



兼容性说明：固件版本（Firmware Version）

强烈建议跟新 Active 跟踪组件（Base Station、Tag）上的固件。

Firmware v0.8 Base Station 与 Tags

- 2017 年 9 月之前发货的 Active 组件使用 v0.8 版本固件。
- v0.8 Base Station 仅适用于 v0.8 Tags。

- 固件 v0.8 的组件需要搭配 eSync 共同使用 ,才能同步 Mocap 系统与 Base Station。
- v0.8 Base Station 重新上电时 v0.8 Tags 也必须重新上电
- v0.8 Base Station 与 v1.0 Tags 不兼容。
- v0.8 Base Station 只能发出 180Hz 的同步信号

Firmware v1.0 Base Station 与 Tags


- 2017 年 9 月之后发货的 Active 组件使用 v1.0 版本固件。
- 不需要使用 eSync 同步 Base Station 与 Mocap 系统。
- 支持用户自定义相机帧率。
- 允许更改发光方式的深度，允许更多数量的 Active Markers。
- Active LED 发光与相机曝光时间同步。
- 仅与 Motive 2.0 beta2 或更高版本兼容。
- V1.0 Base Station 仅与 v1.0 Tags 兼容。



v0.8 Tags 与 v1.0 Base Stations

- 每次关闭并重新启动 Motive 时 v0.8 Tags 需要重新上电。
- 每次 Motive 更改系统帧率时 v0.8 Tags 需要重新上电

5.3.1.1 硬件设置

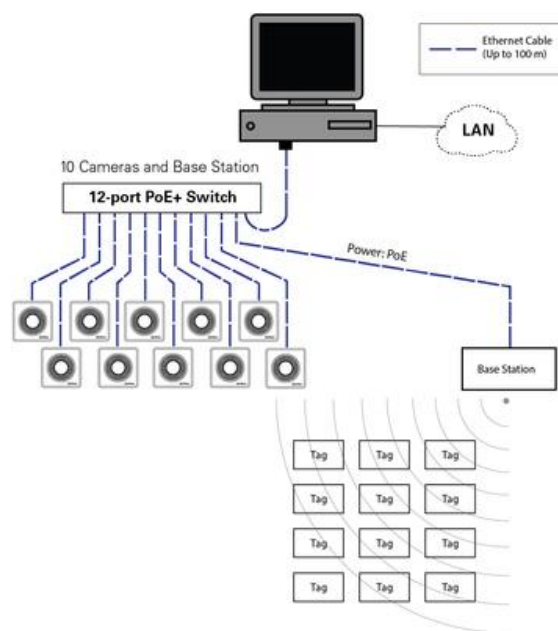
1) 需要组件

名称	描述	图像
Base Station	相机系统以及 Active Marker 的同步控制器，通过 RCA 以及射频波发出同步信号，精确同步相机快门与 Active LED Marker 的发光。	

Tag	Tags 接收来自基站的射频信号，同步与之连接的 Active Markers 的发光	
Active LEDs	<p>发射 850 nm 红外光，每组包含 4 个 Active LED，每个 Tag 最多连接两组。</p> <p>（每个 Tag 有 8 个 Active LED）</p> <p>尺寸：5 mm (T1¾) 塑料封装，半角±65°，typ. 100mA 时 12 mW / sr。</p>	

2) 硬件连接

- 相机系统：Active Marker 只支持以太网相机系统，使用 Active Marker 之前需搭建相机系统，并完成标定。



- Base Station：连接到以太网相机系统的 PoE 交换机上；为了获得最好跟踪效果，将 Base Station 靠近捕获空间的中心放置，使用过程中 Tag 需要放置在视线无障碍干扰的位置，尽管无线信号能够穿过许多类型的障碍物，但仍然存在由于干扰而导致的使用范围减小的可能性，特别是从金属和其它致密材料；不要在 Base Station 附近放置外

部电磁或射频设备。

Base Station 信号

Base Station 通过发出的射频信号 (radio frequency waves) 同步 Active Markers ,

下面是 RF 信号规格 :

- 802.15.4 @ 2.4 GHz
- Channel 20
- Base Station 在相机系统的每 10 帧发出一次射频信号。
- Tag + Active Markers : 将两组 Active Marker 连接到 Tag 上 ; 为 Tag 供电 , 使用 Micro USB 线缆将 Tag 连接到电源 , 电源要求在 2.7V-5.5V 之间 , 经测量 100mA@5V 支持所有 LED 在 180Hz 下每帧工作 2ms ;LED 灯显示红色 ,可以在 2D Camera View 视图中看到 LED 灯。



5.3.1.2 软件设置

1) Motive 设置 (Live Reconstruction)

打开 Edit → Application Settings 界面 , 切换到 Live Reconstruction 窗口。

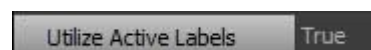
- 将 Marker Labeling

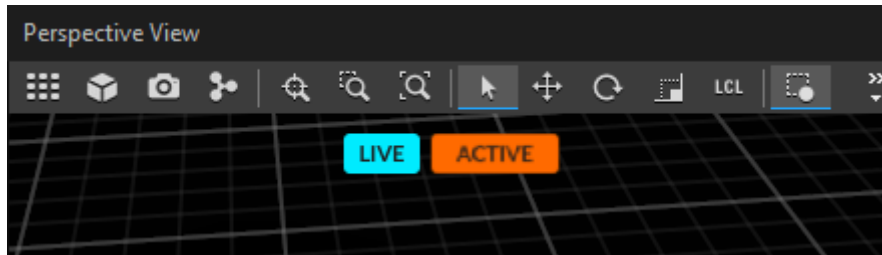
Mode 设置为 Active and

Passive Markers 或者 Active Marker Only。



- 将 Utilize Active Labels 设置为 True。

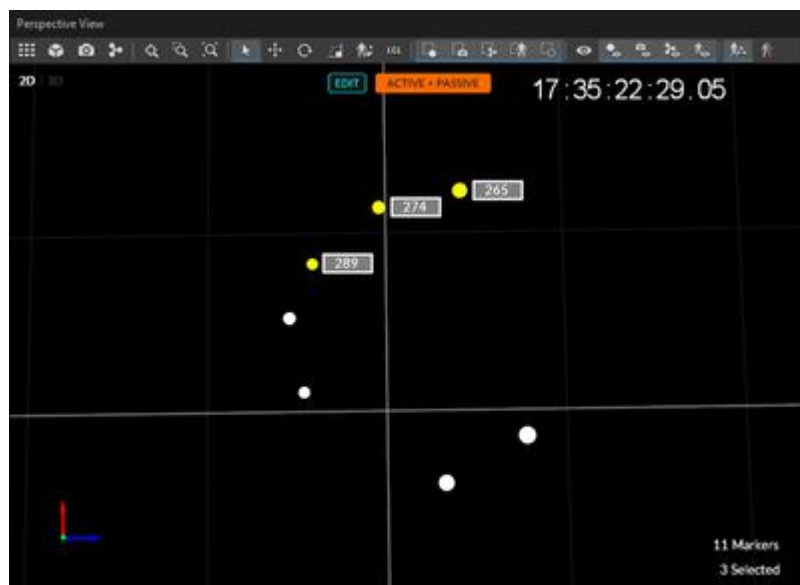




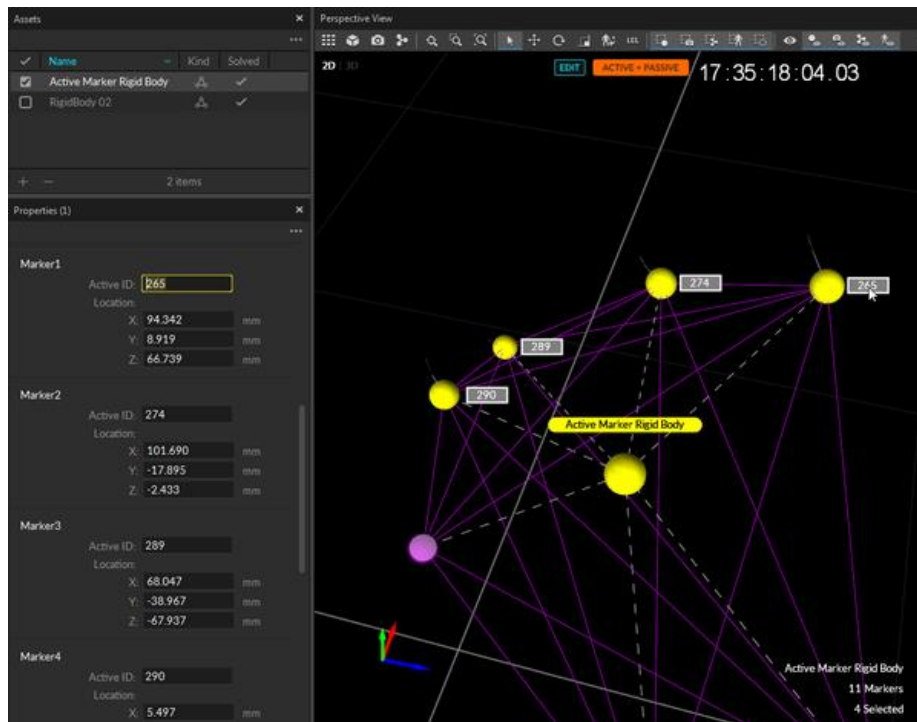
- Active Patten Depth : 默认设为 12 , 设置 Active Marker 发光模式的复杂性 , 跟踪更多刚体时需要设置为更大值 , 如果设置过低会产生重复的 Active ID。

2) Active Labels

完成上述设置之后 , Motive 将重建并跟踪 Active Marker。根据独特的发光模式 , 每个 Active Marker 都将被单独命名 , 根据不同的发光模式 , Motive 会分配给每个重建的 Active Marker 唯一的 ID , 可在 Perspective View 中查看。



如果使用 Active Marker 重建刚体 , 对应的 Active Marker ID 将保存在相应的刚体属性下。为了跟踪刚体 , 除了相匹配标记位置之外 , 还需要捕捉空间中重建并跟踪匹配 Active Marker 点 ID。



5.3.2 HMD Tracking

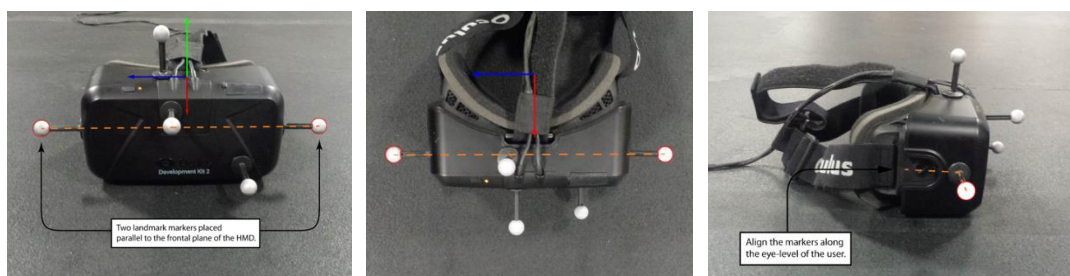
5.3.2.1 Passive Marker

1) HMD Marker 点安装

HMD(Oculus)刚体创建时布置在 HMD 上 Marker 点可参照下图。建议使用宽 20mm 高 30mm 的 M4 螺纹底座，并使用丙烯酸粘胶剂固定在 HMD 上。

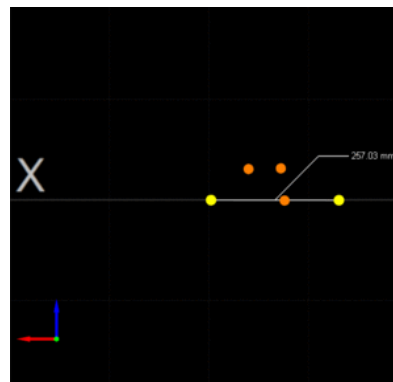
两侧的 Marker 点应与眼睛同高，同时保证两侧的 Marker 点连线平行于 HMD，最重要的是两侧的 Marker 点需精确的对称，且使用相同的底座安装。

如存在多个 HMD，需保证其余 Marker 点的放置不一致。



2) HMD Rigid Body 创建

创建 HMD 刚体时，需要将 HMD 平放在地面上，调整 HMD 位置，使得在 Motive 中 HMD 两侧的 Marker 点与 X 轴平行。然后创建 Rigid Body。

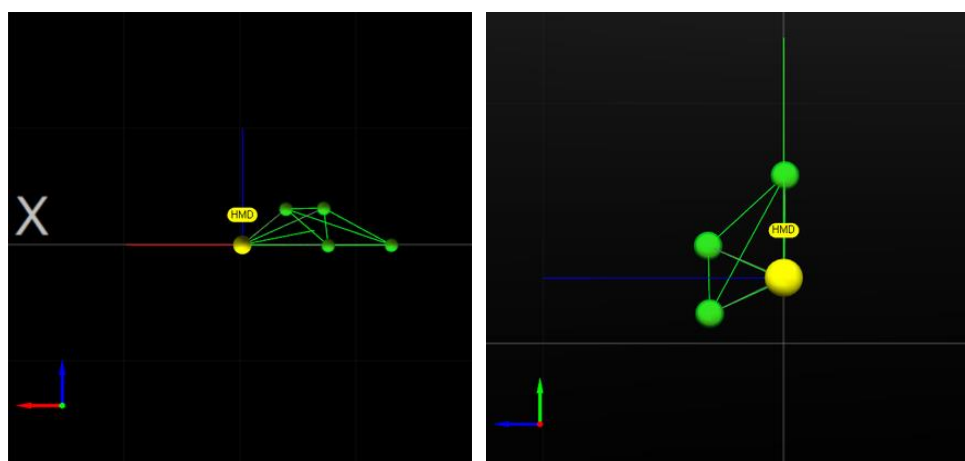



3) HMD 质心调整

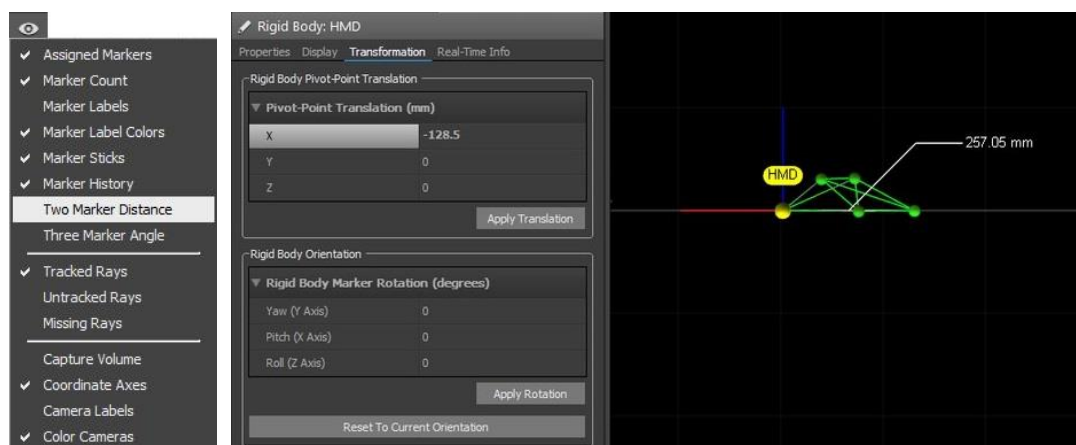
为了良好的跟踪定位 HMD 位置，需要仔细设定 HMD 的质心位置。需满足下面三个原则。

- 质心点应在两眼之间的中点处；
- 刚体的坐标轴应与 HMD 对齐；
- 刚体 X 轴需平行于地面并垂直于 HMD 的显示。

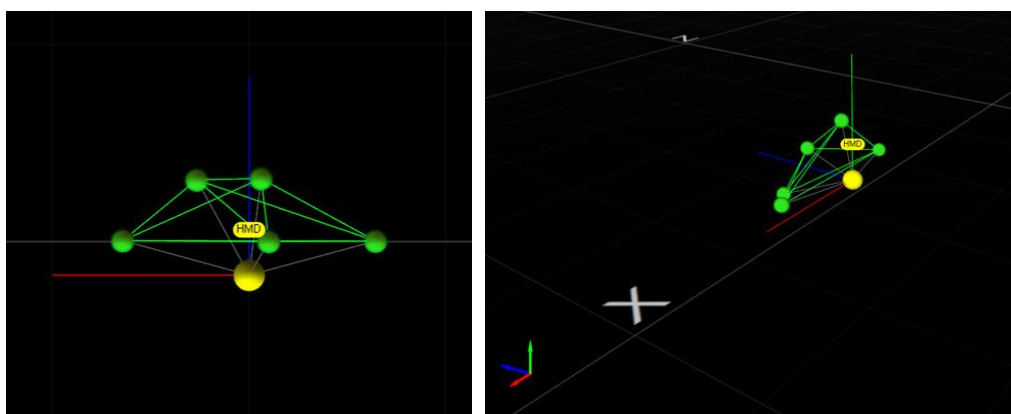
首先需要将质心点移至两侧的 Marker 点处（选中该刚体质心，同时按住 Ctrl 选中需要将质心移至的 Marker 点处，右键 Rigid Body → Set Pivot Point to Selected Marker），使得质心点处在眼睛所在高度。



其次，将质心点移至连线中点处。在 Perspective 窗口中点击  选择 Two Marker Distance，选中两边 Marker 点，可以获得两点间距离，然后在 Rigid Body 的 Transformation 中设置沿 X 轴方向移动间距的一半，将质心移动到两点中间。



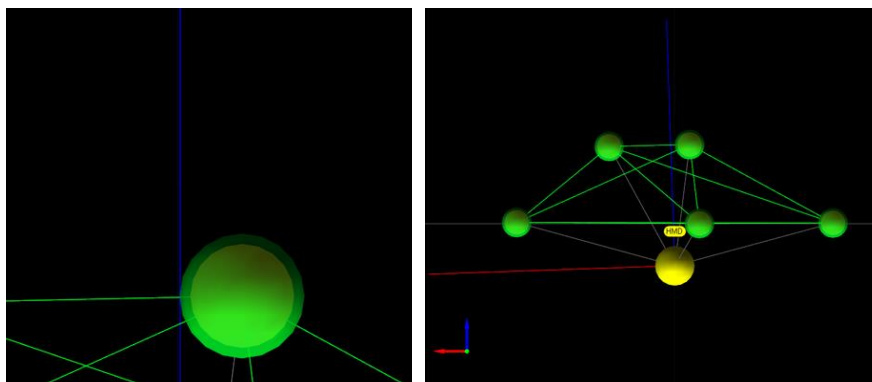
然后，将质心点移至鼻根处。为了获得最准确的位置，需要准确测量两 Marker 点至鼻根处的 Z 轴距离，平移后完成。



最后，对刚体的坐标轴进行旋转。将两侧 Marker 点平行于 X 轴放置，大致将质心点放置在 Z 轴上，进行旋转调节，使得刚体坐标轴与 Motive 网格线平行。

在 Unreal Engine 中：HMD 刚体的 X 轴必须向前。

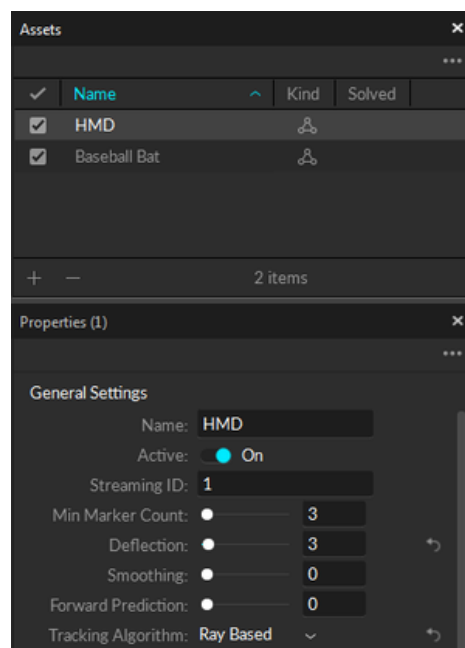
在 Unity 中：HMD 刚体的 Z 轴必须向前



4) HMD 刚体属性设置：

对于刚体的属性需注意以下几点：

- Streaming ID: 每一个刚体的识别号，在外部程序中识别每一个刚体（包括 HMD），引擎接收数据时主要的依据；
- Min Marker Count: 刚体的最少所需的 marker 点数，默认是 3。为了防止刚体混淆，可能需要将其增加至 4。当跟踪多个 HMD 时，如果该值为 3，则可能出现 HMD 识别混淆的错误。
- Deflection：刚体上 Marker 点可能从预期位置偏离的大小，默认值为 4mm，对于 HMD 应用，建议降低该值的大小至 3 或 2。会减少计算量，使得整体延迟降低；
- Tracking Algorithm: 使用 Ray Based 算法。这个设置决定在 Motive 中解算刚体采用的协议，推荐使用 Ray Based 算法，允许在跟踪过程中不易受遮挡影响，比较适合 HMD 跟踪。



5.3.2.2 Active LED HMD bracket

1) 安装/拆卸

HMD 支架上有安装有 Active LED 和 Tag，可以安装在 HMD 外壳上。将支架的底部滑动到 HMD 上，然后夹住顶部，如图所示。

从 HMD 上拆下支架：拉出支架后面的两个闩锁，并从顶部缓慢拉出。



2) 刚体创建

创建 HMD 刚体时，需要将 HMD 平放在地面上，调整 HMD 位置，使得在 Motive 中 HMD 两侧的 Marker 点与 X 轴平行。然后创建 Rigid Body。

- 确保 Active 硬件正确连接，Motive 软件设置为跟踪 Active Marker，为 HMD 支架供电。
- 在 Motive 中，选择 HMD 上的标记点，创建刚体。
- 打开 Trackables → Rigid Body Refine 窗口优化刚体跟踪结果。
- 如有必要，调整刚体质心于坐标轴，与游戏引擎坐标系匹配。

在 Unreal Engine 中：HMD 刚体的 X 轴必须向前。

在 Unity 中：HMD 刚体的 Z 轴必须向前

5.3.2.3 HMD Calibration

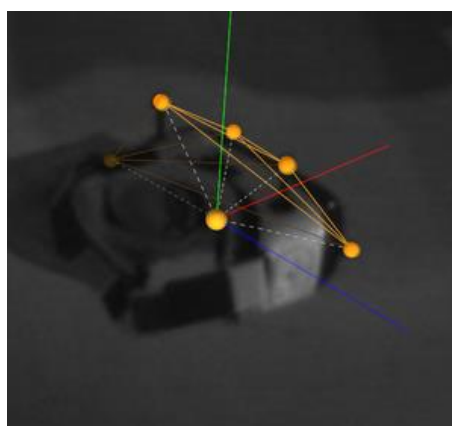
HMD Calibration 功能能够校准 HMD 刚体，使其质心位于 VR 应用的正确位置。利用 OptiTrack Motion Capture 系统和 Oculus Tracker 来精确计算 Oculus HMD 质心的位置。使用此功能时，两个系统必须同时跟踪 HMD 刚体。

- HMD Calibration 只支持 Oculus HMD。
- Oculus Tracker 和 OptiTrack 系统必须同时跟踪 HMD 刚体。HMD 正面正对着 Oculus Tracker。

-
- 此功能仅在 Live 模式下工作。

操作步骤：

- 电脑连接 Oculus Tracker，同时 Oculus 软件在后台运行。
- 在 Motive 中，完成标定，创建/导入 HMD 刚体，开始跟踪 HMD。
- 打开 Trackables → HMD Calibration 窗口，选择 HMD 刚体。
- 设置所需的采样点数量（默认为 750）和坐标轴朝向。
- 点击 Start 开始捕捉采样点，确保此时 mocap 系统和 Oculus Tracker 都跟踪到 HMD。
- 将 HMD 朝向 Oculus Tracker 并缓慢旋转，采集不同的方向。（如果使用 Active LED HMD bracket 会将头显前面的传感器遮盖住，需要将头显底部朝向 Oculus Tracker）
- 采样点捕捉完成之后，将显示 Oculus 检测到的质心的位置与运动捕捉系统确定的质心之间的偏移距离。
- 点击 Apply，校准 HMD 刚体。



5.3.3 Unity Plugin

OptiTrack Unity3D 插件可以将实时或记录的运动捕捉数据从 Motive 传输到 Unity。可以通过插件将来自刚体和骨骼的运动捕捉数据以数据流形式传输，Unity 接收到的数据可用于场景中的对象和角色。插件将数据内容以 unitypackage 格式分发，可以将数据文件简单地加载到 Unity 项目中。导入之后，其中的 C# 脚本可以附加到 Unity 的 Object 上，用于实例化客户端和接收跟踪数据。


通过集成的 Unity 动画系统 (Mecanim), Motive 通过 Unity 插件能够传输全身人体骨骼数据, 不需要中间的重定向软件。来自 Motive 的骨架跟踪数据作为分层骨骼段传输, 并将该数据馈送到 Unity 的 Mecanim 系统中, 该系统允许将不同比例角色做成动画。

Unity Plugin 还可以用于虚拟现实中 HMD 跟踪集成, 支持 HMD : Oculus DK2、Oculus CV1、Gear VR。

Unity 版本 : 5.3+ (5.3.6p1 + , 5.4.0p1 或以上)。Oculus SDK 的版本应与 Unity 版本兼容。

下载地址 : <http://url.cn/5omcUL3>。

5.3.3.1 Motive 设置

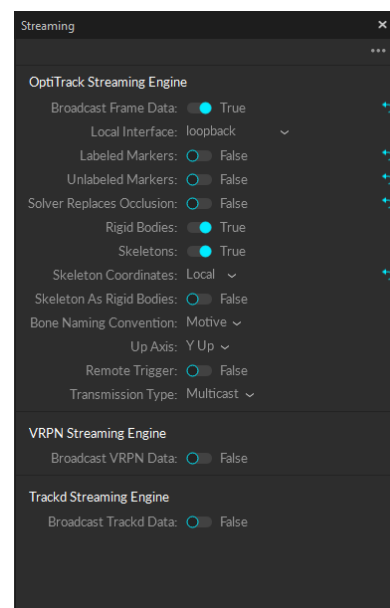
在 Motive 中可以通过实时方式或将已记录的数据传输到其他程序, 可通过 View 选项卡下 Data Streaming Pane 或  打开数据传输窗口, 设置连接、传输数据等选项。为了减小数据包大小, 简化客户端解析过程, 需要禁用不必要的数据传输。确保以下数据能够传输 :

传输 Rigid Bodies 或 HMD

- Stream Rigid Bodies → True

传输 Skeletons

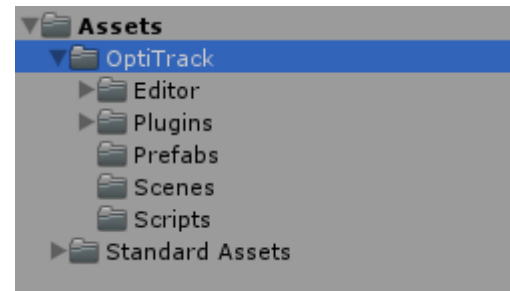
- Stream Local Rigid Bodies → True
- Stream Skeleton → True
- Skeleton As Rigid Bodies → False



5.3.3.2 Unity 设置

1) 导入插件包

在 Unity 项目中 双击插件的 unitypackage 文件，并将插件资源导入到项目中。插件包导入后，项目中将显示以下内容：



文件夹	内容说明
Assets/OptiTrack	所有 Unity 插件内容都包含在此文件夹中。
Assets/OptiTrack/Scripts	包含 C#脚本组件，导入到 Unity Object 中接收数据流。
Assets/OptiTrack/Editor	包含编辑器脚本和用于可视化刚体标记的内容。
Assets/OptiTrack/Plugins	包含插件 lib 文件和头文件
Assets/OptiTrack/Plugins	包含预定义的 Client、Skeleton、HMD 对象。
Assets/OptiTrack/Scenes	包含预配置的客户端，刚体，骨架和 HMD 对象的示例 Unity 场景。

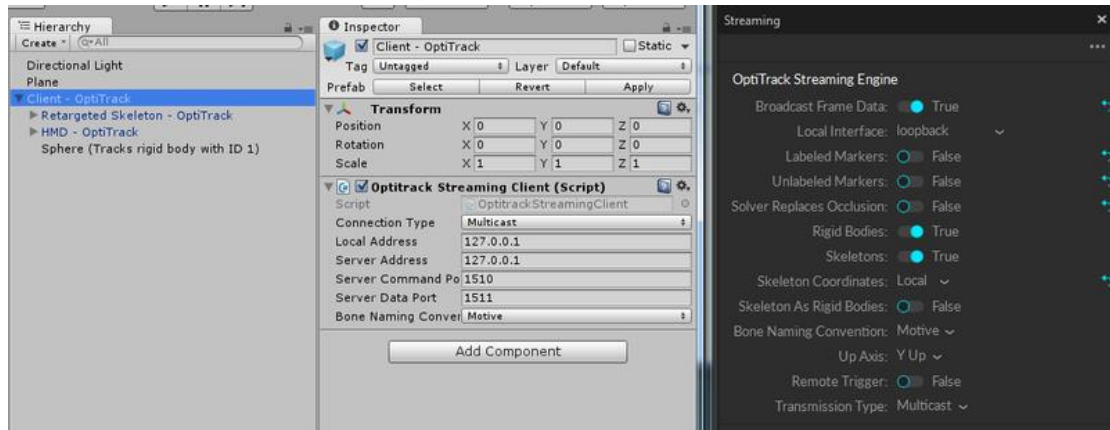
2) 设置客户端

为了从服务器应用程序（Motive）中接收到跟踪数据，需要设置客户端程序。客户端 Object 可以是 Unity 场景中任何 Object，通过附加 OptitrackStreamingClient.cs 脚本指定。此脚本从连接的服务器应用程序（例如 Motive）接收跟踪数据，并使数据应用于场景中。可以将客户端脚本附加到现有对象或空对象上。此外，您可以从 Assets / Optitrack / Prefabs 文件夹导入示例。

- Unity 中 Prefabs 文件夹下，将客户端预定义的 Object 导入到场景中，或者将 OptitrackStreamingClient.cs 脚本附加到现有 Object 上。
- Unity 中 Client Object → OptiTrack Streaming Client 脚本中设置连接传参数，使得 Unity 中 Connection Type、Local Address、Server Address、Server

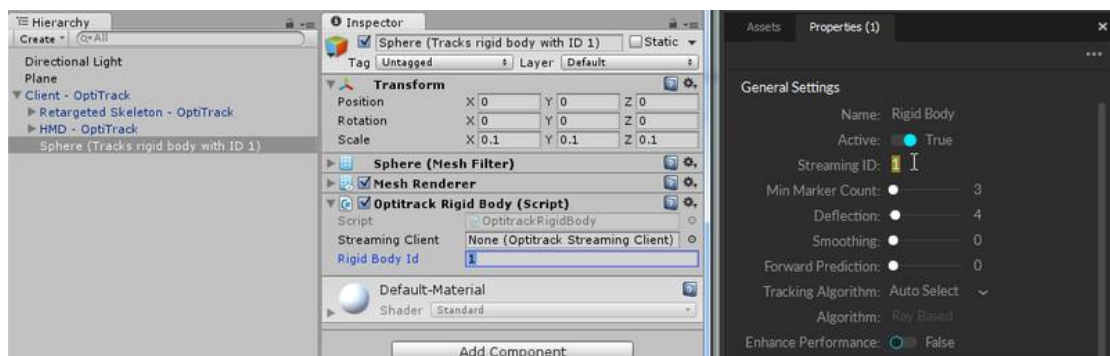
Command Port、Server Data Port 等参数与 Motive 中 Streaming 窗口下的参数相匹配。

- Unity 中设置骨骼命名方式，使其符合 Motive 中设置的命名方式。



3) Rigid Body 动画

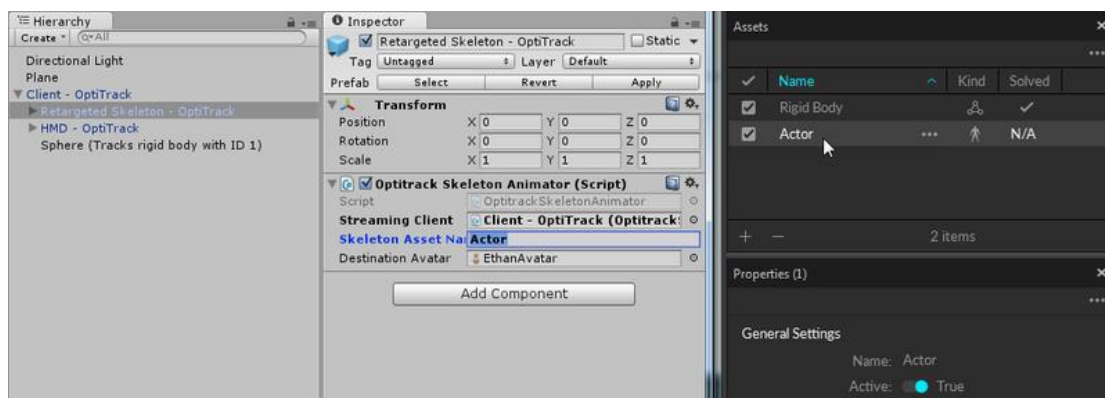
- Unity 中在需要驱动的 Object 上附加 OptitrackRigidBody.cs 脚本。
- 在 Streaming Client 输入窗中，链接客户端脚本所连接的 Unity Client 对象。
- Rigid Body ID 中输入 Motive 中对应的刚体资产的 ID。
- 确保 Motive 中数据流广播打开。
- 播放场景。



4) Skeleton 动画

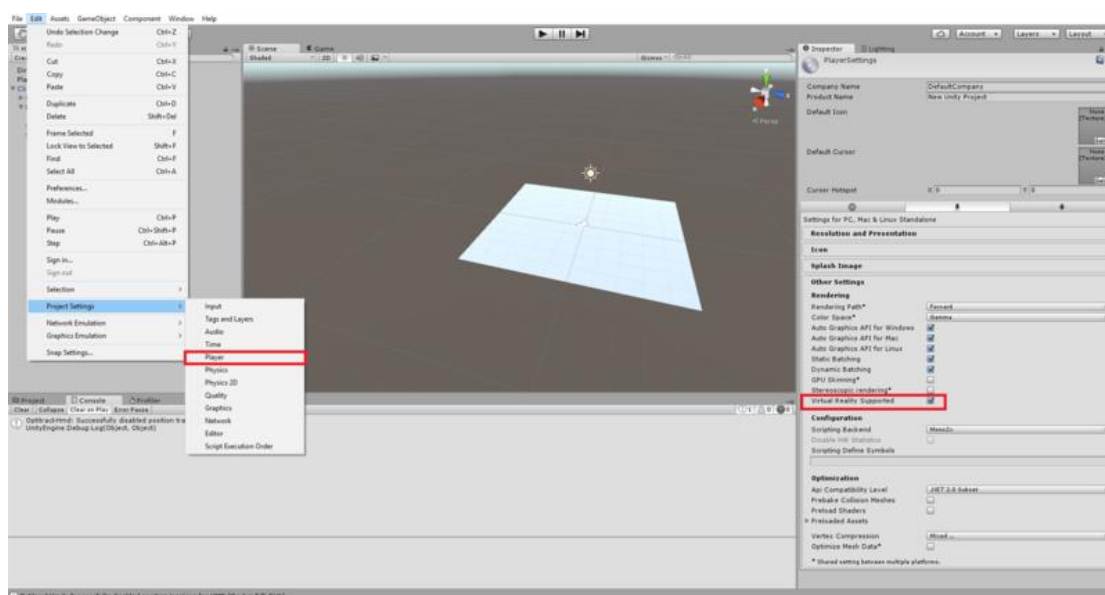
- Unity 中在需要驱动的 Character 上附加 OptitrackSkeletonAnimator.cs 脚本。
- 在 Streaming Client 输入窗中，链接客户端脚本所连接的 Unity Client 对象。

- 在 Skeleton Asset Name 中输入 Motive 中对应的 Skeleton 名称。
- 在 Destination Avatar 中输入加载脚本的 Character。
- 确保 Motive 中数据流广播打开。
- 播放场景。



5) HMD

- 参照 HMD Tracking 放置 Marker 点，并在 Motive 中创建 Rigid Body。
- 确保 HMD 软件正在工作。
- 在 Unity 项目中，打开 Edit → Project Settings → Player，确定 Player Settings 中 Virtual Reality Supported 使能。



- 在导入的插件包中的 Prefabs 文件夹 (OptiTrack / Prefabs) 中，加载 HMD –

OptiTrack 到场景中。

- 附加 OptiTrackHmd.cs 脚本。
- 在 Streaming Client 输入窗中，链接客户端脚本所连接的 Unity Client 对象。
- Rigid Body ID 中输入 Motive 中对应的刚体资产的 ID。
- 确保 Motive 中数据流广播打开。
- 播放场景。

5.3.4 Unreal Engine Plugin

OptiTrack Unreal Engine 4 插件可以将实时或记录的运动捕捉数据从 Motive 传输到 Unreal Engine 4 (简称 UE4) 中。UE4 插件还可以用于虚拟现实 HMD 跟踪集成。将 OptiTrack UE4 插件添加到 UE4 插件目录中，OptiTrack - NatNet Streaming Client 和 OptiTrack - Oculus Rift 插件将在 UE4 项目中可用。NatNet Streaming Client 插件可以将 Motive 中的刚体跟踪数据传输到 UE4 中，Oculus Rift 插件可以在场景中实现 HMD 跟踪。调用 OptiTrack Client Origin，通过 NatNet 协议将工程链接到 Server (Motive) 应用，并导入跟踪数据。Client Origin 导入到场景中，会与 Motive 中 Global 原点关联，接收的跟踪数据将参考 Client Origin。该原点可以放置在场景中任何地方，并缩放到不同大小。

OptiTrack Unreal Engine 4 插件可以将 Motive 中实时刚体和 HMD 跟踪数据传输到 UE4。

版本要求：


Plugin Version	UE4 Version	Oculus SDK
1.0	UE 4.10	0.8
1.1	UE 4.11	1.3

1.2	UE 4.12	Latest
1.3	UE 4.13	Latest
1.4	UE 4.14	Latest
1.5	UE 4.15	Latest
1.6	UE 4.16	Latest

OptiTrack Unreal Engine 4 Plugin 1.3.0 支持 HMD : Oculus DK2、Oculus CV1、Gear VR。

下载地址：<http://url.cn/5Z0Wx5k>。

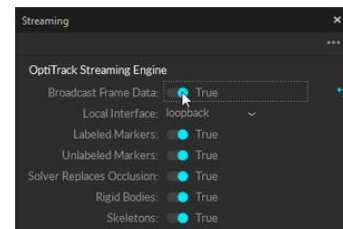
5.3.4.1 Motive 设置

在 Motive 中可以通过实时方式或将已记录的数据传输到其他程序，可通过 View 选项卡下 Data Streaming Pane 或  打开数据传输窗口，设置连接、传输数据等选项。为了减小数据包大小，简化客户端解析过程，需要禁用不必要的数据传输。确保以下数据能够传输：

传输 Rigid Bodies 或 HMD：

- Stream Rigid Bodies → True。

注意：对于 GearVR，必须使用 Unicast 通信。



5.3.4.2 Unreal Engine 4 设置

1) 导入插件包

- 下载 OptiTrack UE4 插件包，解压文件夹，将 OptiTrack NatNet 和 OptiTrack Oculus Rift 文件夹复制到 UE4 插件目录中。默认情况下，该目录安装在 C : \ Program Files \ Epic Games \ 4.11 \ Engine \ Plugins 文件夹中，将 OptiTrack 插件文件夹复制并粘贴到此目录中。

- 打开/创建 UE4 工程，并在插件设备下启用两个 OptiTrack 插件设备。在主命令栏的 Edit 菜单下，打开插件面板，选择虚拟现实部分，并启用添加到文件夹的 OptiTrack 插件 (OptiTrack - NatNet Streaming Client 和 OptiTrack - Oculus Rift)。

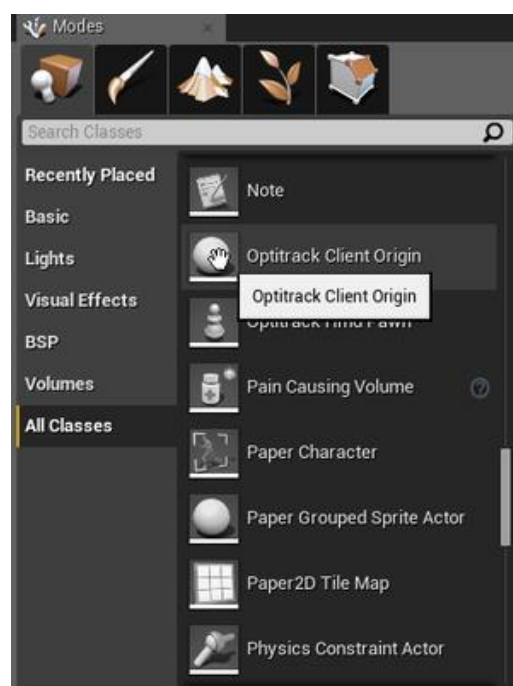


注意：使用 OptiTrack Oculus 插件时，禁用 virtual reality 插件类别下的内置 (Build-In) Oculus Rift (或 Gear VR) 和 Stream VR HMD 插件，因为默认情况下，这些插件的优先级要高于 OptiTrack Oculus 插件，如果启用，将会优先使用。

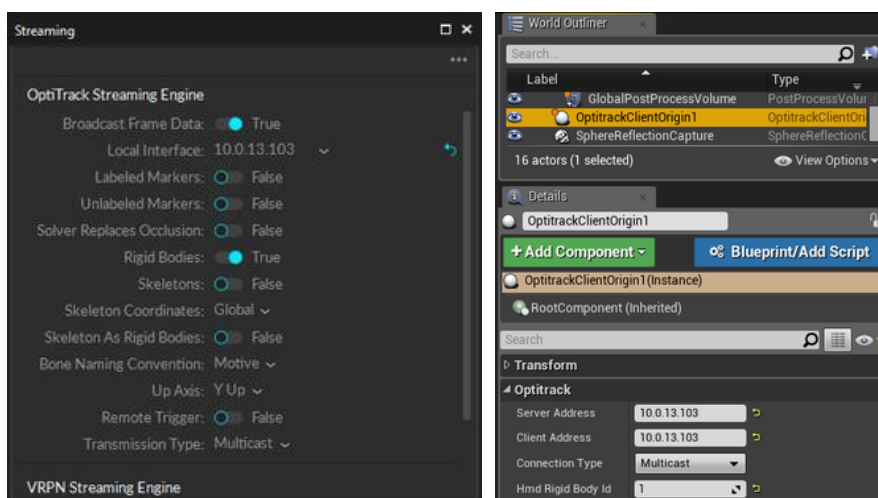
2) Unreal Engine Client 设置

OptiTrack Client Origin 将 Unreal Engine 与 NatNet 连接通信，获取 Motive 中刚体和 HMD 的跟踪数据。在 OptiTrack Client Origin Object 中设置 HMD 通信地址和刚体 ID。将 OptiTrack Client Origin 从 Modes 面板拖动到场景中就可以添加这一 Object。

Global Origin(世界坐标系原点) : UE4 中 OptiTrack Client Origin 的位置与 Motive 中世界坐标系原点对应，跟踪数据将以 UE4 场景中 OptiTrack Client Origin 为基准。



UE4 与 Motive 连接

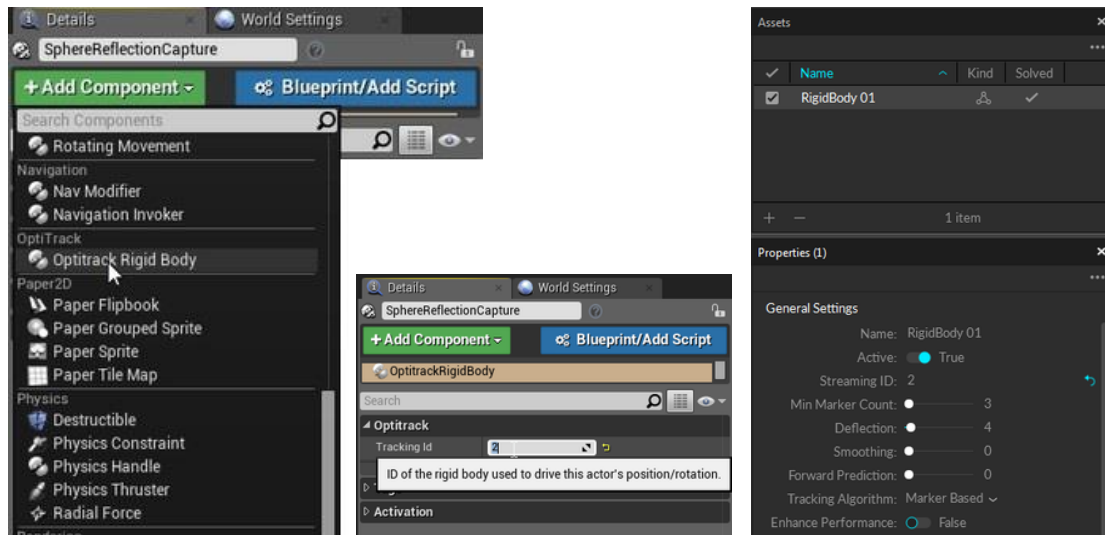


- 确保插件成功加载并使能，将 OptiTrack Client Origin 拖拽到场景中。
- 在 World Outliner 面板中选择实例化的 OptiTrack Client Origin 对象。
- 在 Details 窗口展开 OptiTrack Client Origin 属性，输入与 Motive 匹配的服务器/客户端地址、连接方式以及 HMD 刚体 ID。
- 确保 Motive 中正在广播数据，在 UE4 中播放场景。

3) Rigid Body 动画

UE4 中场景中的道具可以通过 Motive 中刚体的跟踪数据驱动，需要在 Motive 中创建对应的刚体资产，并在 UE4 中使用相同的 ID 连接。

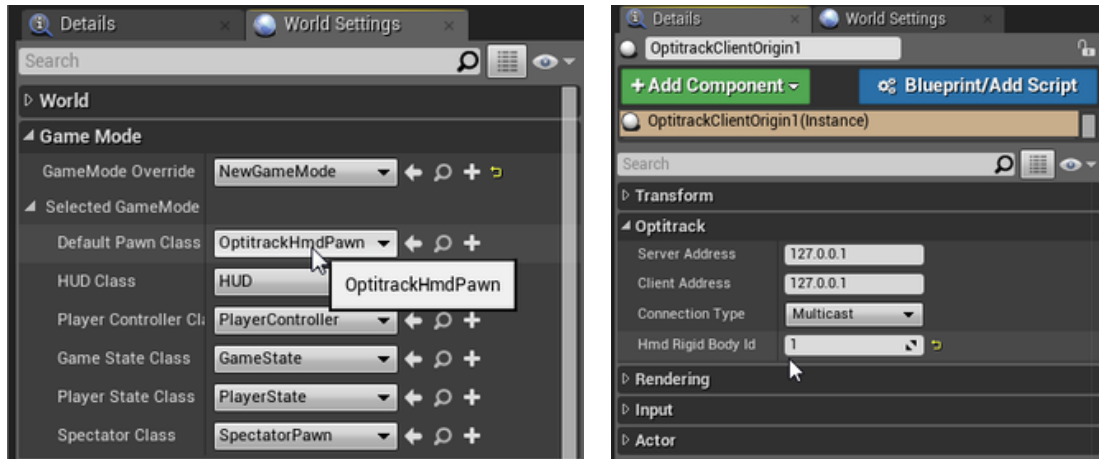
- 在 UE4 场景选择一个 Object。
- 在 Details 窗口，点击 +Add Component 按钮，选择并关联 OptiTrack: OptiTrack Rigid Body 组件。
- 选中 OptiTrackRigidBody 在 Tracking ID 中输入 Motive 中对应的刚体资产的 ID。
- 确保 Motive 中数据流广播打开，在 UE4 中播放场景。



4) HMD (Oculus DK2, CV1)

UE4 工程中调用 OptiTrack Client Origin 之后，HMD 就可以集成到 UE4 中，需确保 UE4 中已经启用 OptiTrack - Oculus Rift 插件。

- 参照 HMD Tracking 放置 Marker 点，并在 Motive 中创建 Rigid Body。
- 确保 HMD 软件正在工作，启用 OptiTrack – Oculus Rift 插件。
- 在 Game Mode 部分添加新的 blueprint，创建新的 GameMode。
- 打开 World Settings 窗口，展开 Game Mode → Selected GameMode，将 Default Pawn Class 更改为 OptiTrackHmdPawn。
- 在 UE4 中，选中 OptitrackClientOrigin，查看 Details 窗口，展开 OptiTrack 列表。
- 在 Hmd Rigid Body Id 中输入 Motive 中对应的刚体资产的 Streaming ID。
- 在 UE4 中播放场景，HMD 会根据位置显示场景。



5.4 Movement Sciences

OptiTrack 运动捕捉系统可以用于生物力学分析，配合力学平台或数据采集设备，提供高精度的生物力学运动捕捉数据。通过设置 OptiTrack 运动捕捉系统可以于力学平台同步，Motive 可以同步采集运动学与动力学数据，用于生物力学分析，临床步态分析，生理学研究，运动性能研究等。同时 OptiTrack 运动捕捉系统可以与 National Instruments 数据采集 (NI-DAQ) 设备集成。通过 NI-DAQ 设备，来自其它设备 (例如 transducers 或 EMG 传感器) 的模拟信号可以以用户定义的采样频率转换成数字信号，并且与 Motive 中的跟踪数据精确同步。

OptiTrack 运动捕捉系统与外部设备同步需要用到 eSync 设备，下面介绍 eSync 使用设置，OptiTrack 运动捕捉系统与力学平台、NI-DAQ 集成使用方法。

5.4.1 eSync

5.4.2 AMTI 力学平台

5.4.3 Bertec 力学平台

5.4.4 NI-DAQ