MOXI SDK Unity Plugin User Guide

Table of contents

Release Notes

v.0.13 : release date: 03/26, 2025

- Improve the firmware update mechanism.
- Fix the issue where the avatar appears to float during rendering.

v.0.12 : release date: 01/16, 2025

- Support firmware updates for upper or lower body devices separately (via USB connection) for MOXI SDK.
- The issue of incorrect firmware information retrieval when connecting the clothing via USB.

v.0.11 : release date: 06/20, 2024

- Add new device IDs for setting up the finger motion.
- Change the avatar in the example scene.
- Enhance network connection stability

v.0.10 : release date: 06/11, 2024

- Provide the multiple floor contact feature.
- Fix the delay in restarting motion capture.

v.0.9 : release date: 05/20, 2024

- Fix an error when the system could not reconnect to the MOXI device via WIFI if the network environment was changed.
- Modify MOXIRecorderComponent to export *.mxm file.

v.0.8 : release date: 05/13, 2024

- Add C Function : MxCStartRecordDeviceData
  - Start to record the motion data from MOXI devices.
- Add C Function : MxCFinishRecordDeviceData
  - Finish recording the motion data from MOXI devices.
- Remove the C Function – MxCGetBoneJointInitialData

v.0.7 : release date: 05/06, 2024

- Add C Function : MxCGetBoneJointInitialData
  - Retrieve the initial pose data for each bone joint in the MOXI SDK.
- Modify C Function :MxCGetBoneJointMotionData
  - Add a flag, "bWorld," to retrieve the motion data in the world coordinate system.

- Fix the callback function for the connection response to prevent it from being called twice if the connection is established again after the disconnection operation.

v.0.6 : release date: 04/19, 2024

- Enhance network connection stability.
- Add functions to receive and process compressed motion data for optimizing network performance.
- Add C Function : MxCIsDeviceConnected
  - Check if an IMU device is connected.
- Add C Function : MxCSetCaptureDataFormat
  - Configure the data format for motion capture.
  - There are three types of formats
    - eMocapDataType.MOXI_MOCAP_TYPE_MOTION
    - eMocapDataType.MOXI_MOCAP_TYPE_RAWMOTION
    - eMocapDataType.MOXI_MOCAP_TYPE_COMPRESSED_MOTION
    - The default type is eMocapDataType.MOXI_MOCAP_TYPE_COMPRESSED_MOTION

v.0.5 : release date: 04/19, 2024

- Shorten the initialization time.
- Add a moving flag to specify whether the avatar can move or not.
- Add C Function : MxCResetCharacterPosition
  - Reset the avatar position to it's base
- Add C Function : MxCStopCapture
  - Stop motion capture

v.0.4 : release date: 05/13, 2024

- add "MOXIRecorder" component to record the motion capture result.
- add file exporter for the animation clip data and the file in the *FBX* file format.
- Modifing the functional input parameters to the error handling callback function for retrieving error information.
- Fix: The callback function for handling error responses was not triggered.

v.0.3 : release date: 03/29, 2024

- modified the letters "Moxi" to "MOXI" in the name of parameters and functions.
- modified the class name "MoxiSDK" to "MOXICFunction".

v.0.2 : release date: 03/22, 2024

- Improve connection stability.
- Improve the UI Sample of the Plugin.

v.0.1 : release date: 03/15, 2024

- Release the sample scene with the UI sample of the Plugin.

v.0.0 : release date: 03/05, 2024

- Update the *MOXI SDK* library (2024.03.05 released).
- Enable floor contact feature.
- Add custom inspector to MOXIController.cs

Introduction

The *MOXI SDK Unity Plugin* is the plugin for developers to use *j-mex* motion capture device in their applications. The *Unity* plugin is the extension of the *MOXI SDK* which is a cross-platform API for *j-mex* IMU-based motion capture device and provides high-performance access to realtime motion control and captures in games, vTubing, and XR applications.

The plugin provides an interactive way to use the *MOXI SDK* in *Unity*, with which developers can connect to the hardware devices and capture the motion data by using the MOXIManager and MOXIController, the two major components of the plugins. Within the software package, a sample scene and the associated UI components are included for reference. The users can modify the sample scene or create a new scene following the instructions to use the *MOXI* system in Unity.

Requirement

- *Unity* version: 2022.3 or higher

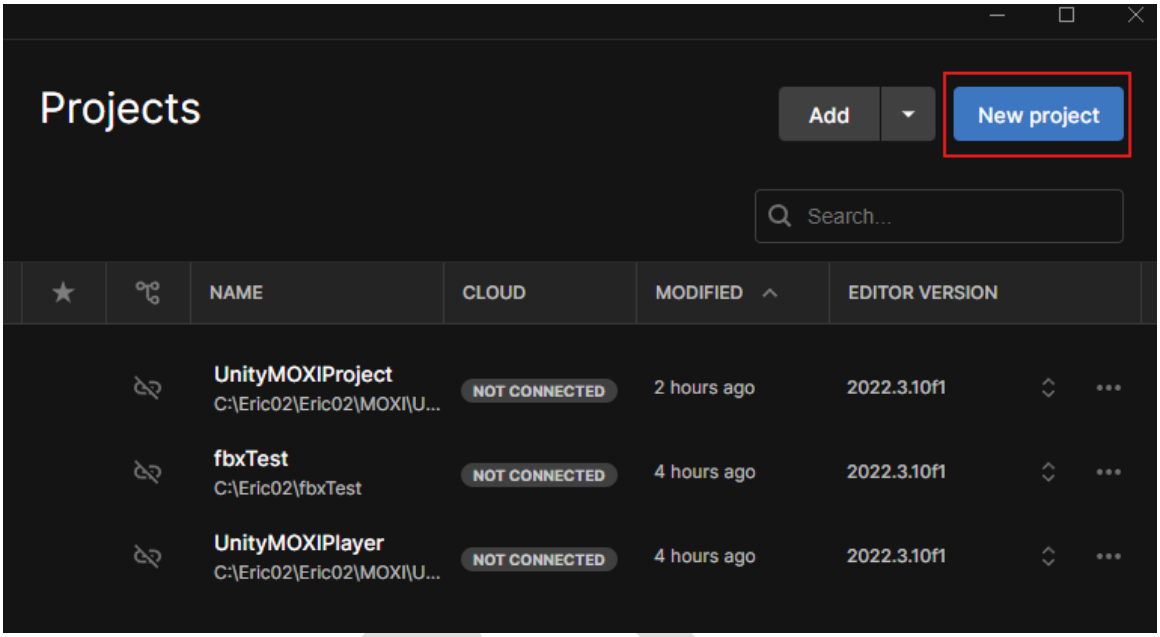
Support Platforms

- Windows
- MacOS
- iOS
- Android

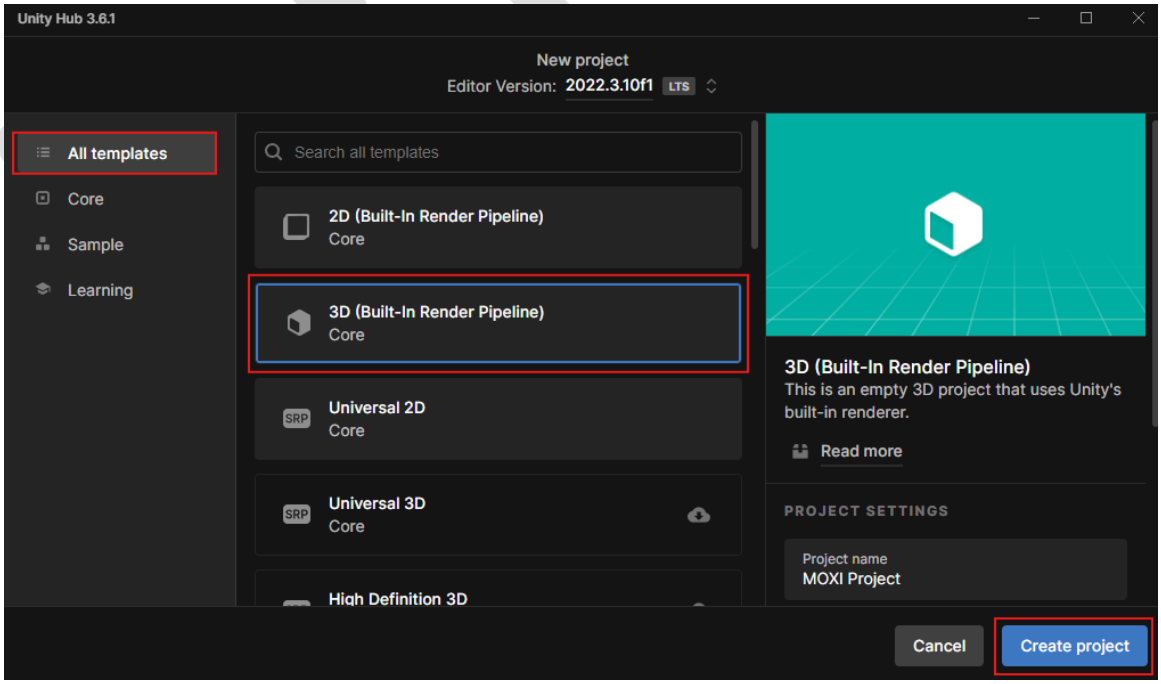
User Guide

1. Setup Unity MOXI Plugin

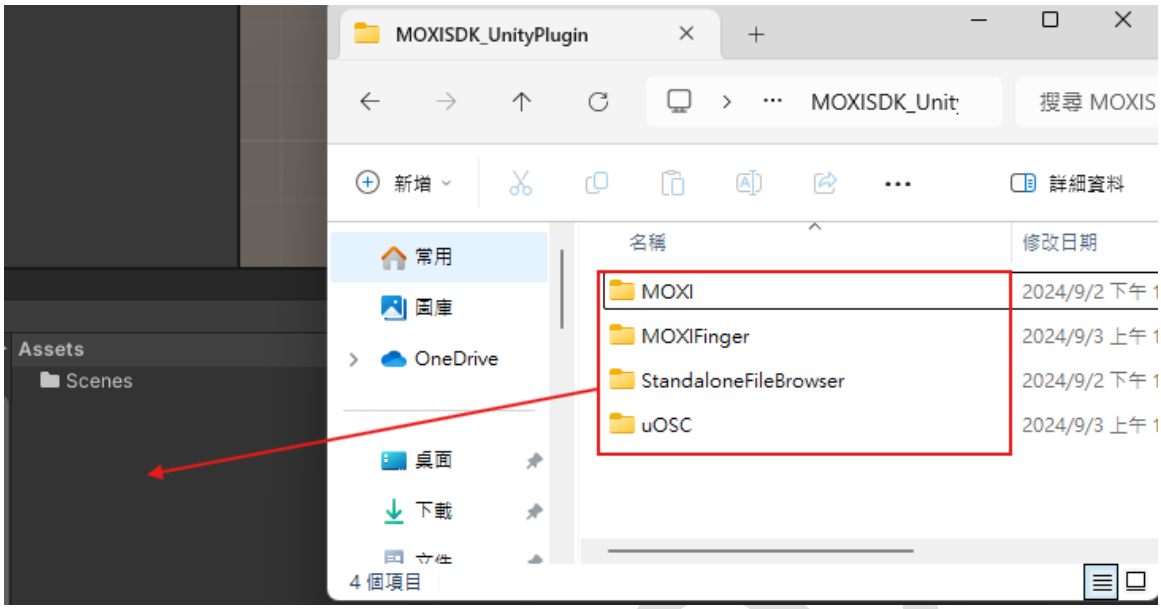
- When starting the *Unity*, use the sample project or create a new *Unity* project by following these steps.
  - Select 'New project' to create a new *Unity* project.



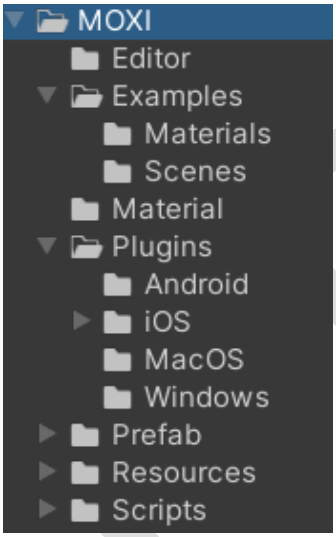
  - Select '3D (Built-in Render Pipeline)' and then click 'Create project' to create a new project.



  - Open the sample project folder. Drag and drop the selected folders located in the 'MOXISDK UnityPlugin' into the project panel under the 'Assets' area.



2. The Plugin content in *Unity* Project

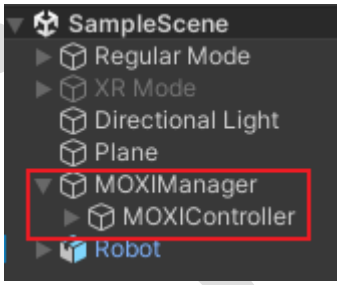


- The MOXI folder contains the necessary assets of the plugin.
  - Editor folder : the scripts to customize the *Unity Editor* for MOXI Components
  - Runtime folder : the runtime scripts and assets
    - Examples folder : the sample scene
    - Plugins folder : the native library for *Android, iOS, MacOS, Windows* platforms
    - Prefab folder : the sample UI object
    - Resources folder : the images for sample UI
    - Scripts folder : the scripts for *MOXI* device management

3. Main scripts (located at Assets->MOXI->Runtime->Scripts)

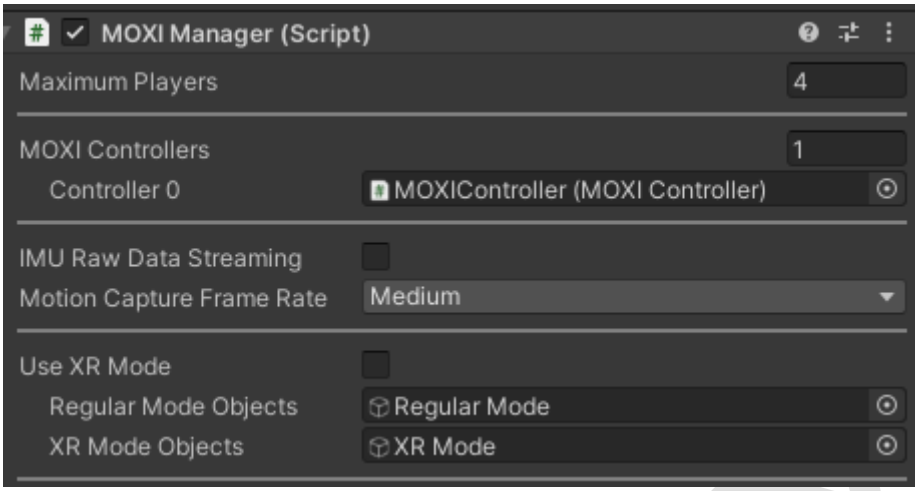
- UI folder
  - The C# scripts for the user interface of the toolkit.
- MOXICFuntion.cs
  - The C# programming interface to the *MOXI SDK* library.
- MOXIController.cs
  - The C# scripts to :
    - Assign a target avatar for a *MOXI* device
    - Setup bone mapping relationship between the *Unity* and *MOXI SDK* library
    - Communicate with the *MOXI* devices using a specified channel ID
- MOXIManager.cs
  - The C# scripts to :
    - Manage all game objects which have combined with MOXIController component
    - Manage the callbacks called by *MOXI SDK* library

4. The hierarchy in the sample scene

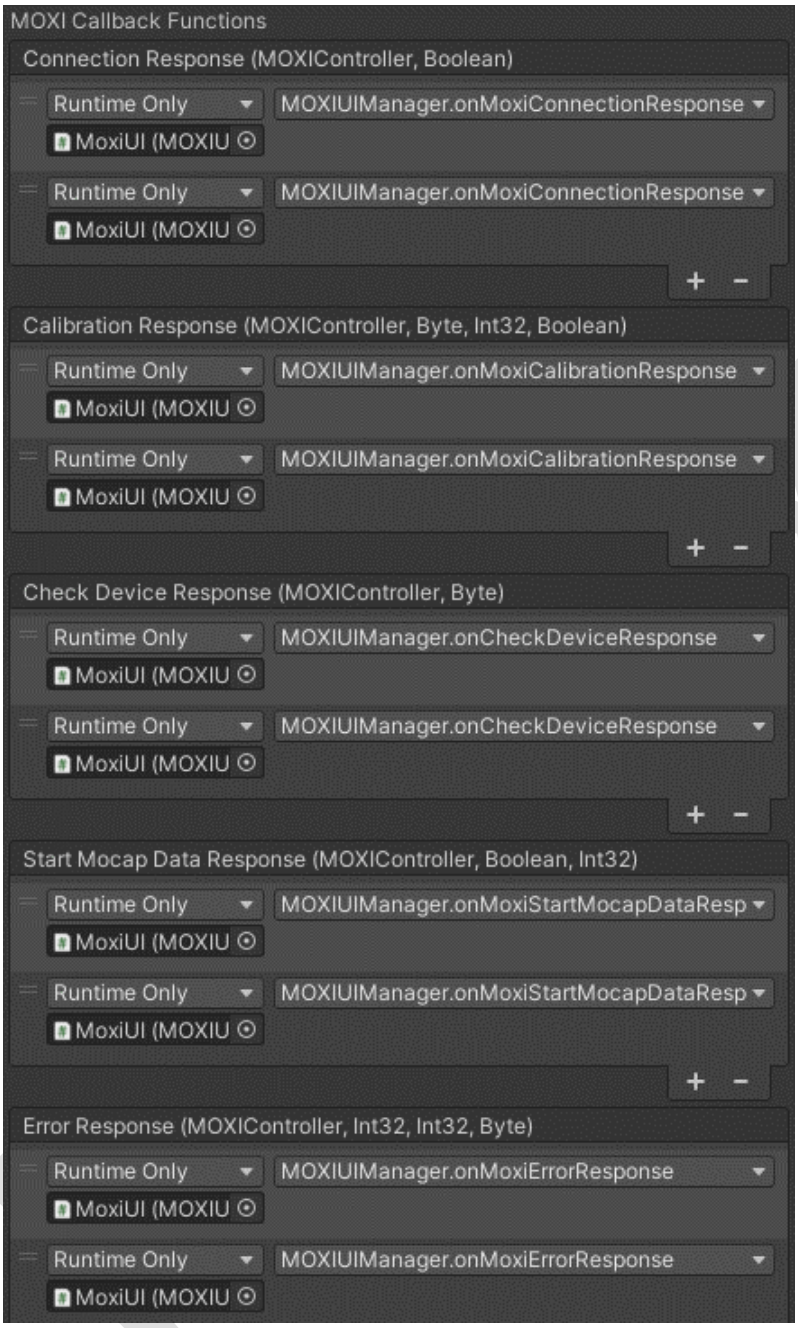


- The above image shows the hierarchy structure of the GameObjects in the sample scene.
  - In the first step, users should add a GameObject named “MOXIManager” and assign the MOXIManager Component of the plugin to the “MOXIManager” GameObject.
  - And then add an empty GameObject with the name, “MOXIController” within the MOXIManager GameObject and assign MOXIController Component of the plugin to it.
  - Assign the avatar used by the MOXIController Component.

- MOXIManager Component:



- Maximum Players
  - The maximum number of *MOXI* devices connected to an application.
  - The default value is 4.
- MOXI Controllers
  - The list of controllers with the scripts.
- IMU Raw Data Streaming
  - Set ON to specify the streaming of IMU raw Data.
  - Turn OFF to indicate that the streaming data is the quaternion data only.
- Motion Capture Frame Rate
  - Select the level of the motion capture frame rate used in the demo app. There are three levels of frequency supported recently: High (100 fps), Medium (60 fps), and Low (30 fps).
- Use XR Mode
  - Set ON to specify to use XR devices with MOXI device.
  - The default value is OFF.

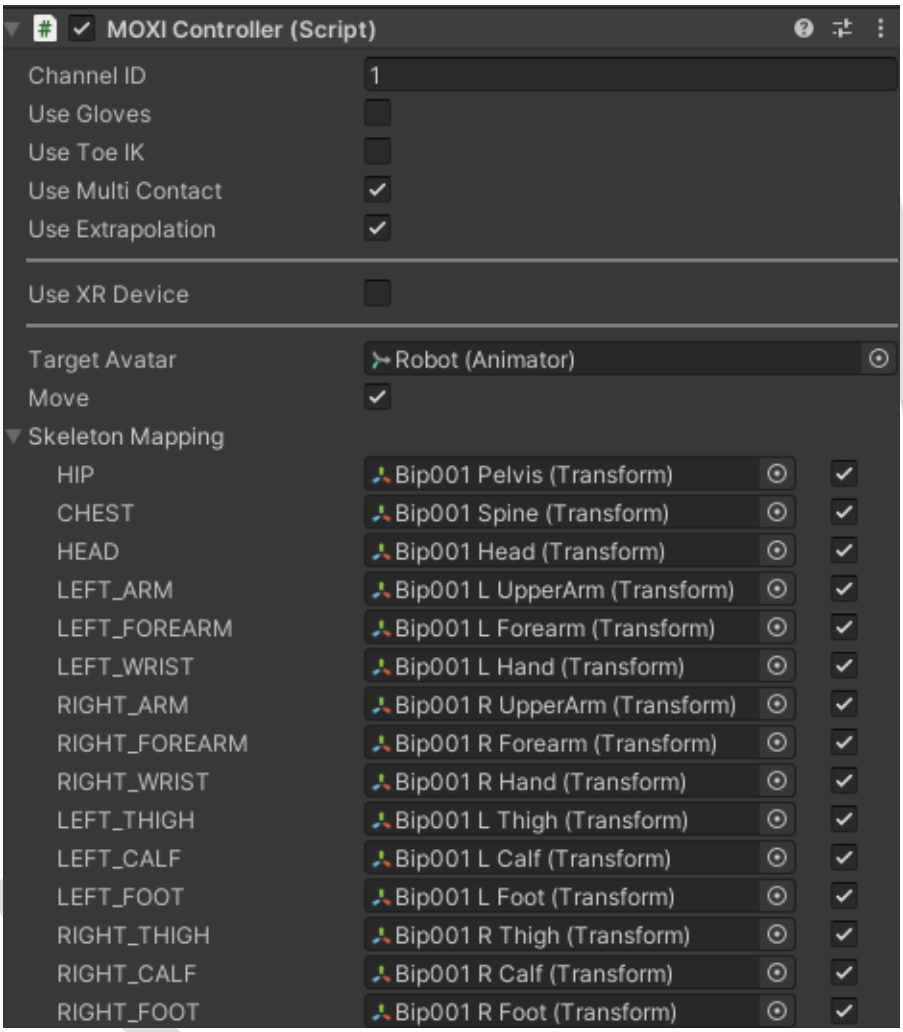


- MOXI Callback Functions
  - Connection Response
    - The callback function to handle the responses from device connection.
  - Calibration Response
    - The callback function to handle the responses in calibration.
  - Check Device Response
    - The callback function to handle the returned connection status of the device hub.
    - The system will return the status of upper-body hub and lower-body hub.
  - Start Mocap Data Response

- The callback function to handle the responses during the setup of natural pose.

- Error Response
  - The callback for error handling.

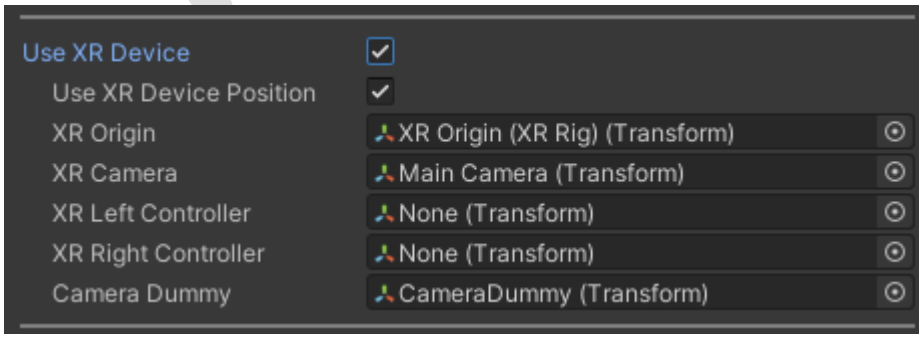
- MOXIController Component:



- Channel ID
  - The channel ID used in *MOXI Connect Mobile App* which is running on the *j-mex* hub to control all IMU-based devices.
  - The channel ID is a positive number which must be unique with the same networking environment. If the avatar is using the device directly connected to the system with USB, the ID is zero.

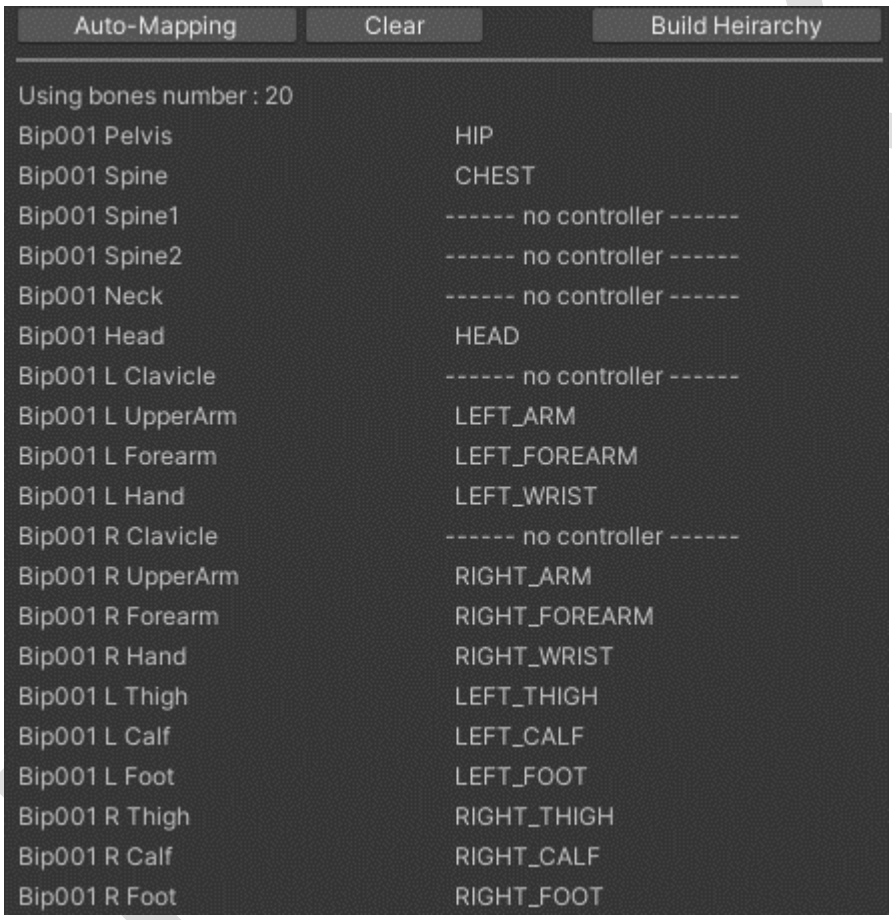
- Use Gloves
  - Set ON to use gloves with *j-mex* devices. Then a sub-menu as the following picture will be opened to specify the information and data of the gloves. For the *StretchSense Studio* gloves, the users need to set the VMC streaming port number.

- Use Toe IK
  - Set ON to use turn on the toe IK calculation
- Use Multi Contact
  - Set ON to have the multiple floor contacts
- Use Extrapolation
  - Set ON to extrapolate the motion data from previous frames when motion data of current frame is not available.
- Use XR Device
  - Set ON to use an XR device with *MOXI* devices. The following options will be hidden when set flag to OFF.
  - The options to use XR devices:
    - Use XR device Position
      - Set ON to apply XR device data to *MOXI SDK*
    - XR Origin
      - To specify XR origin for set headset position.
    - XR Camera
      - To specify XR camera for apply headset data.
    - XR Left Controller
      - To specify XR left hand controller for apply controller data.
    - XR Right Controller
      - To specify XR right hand controller for apply controller data.
    - Camera Dummy
      - To specify a empty GameObject in avatar's eye position.



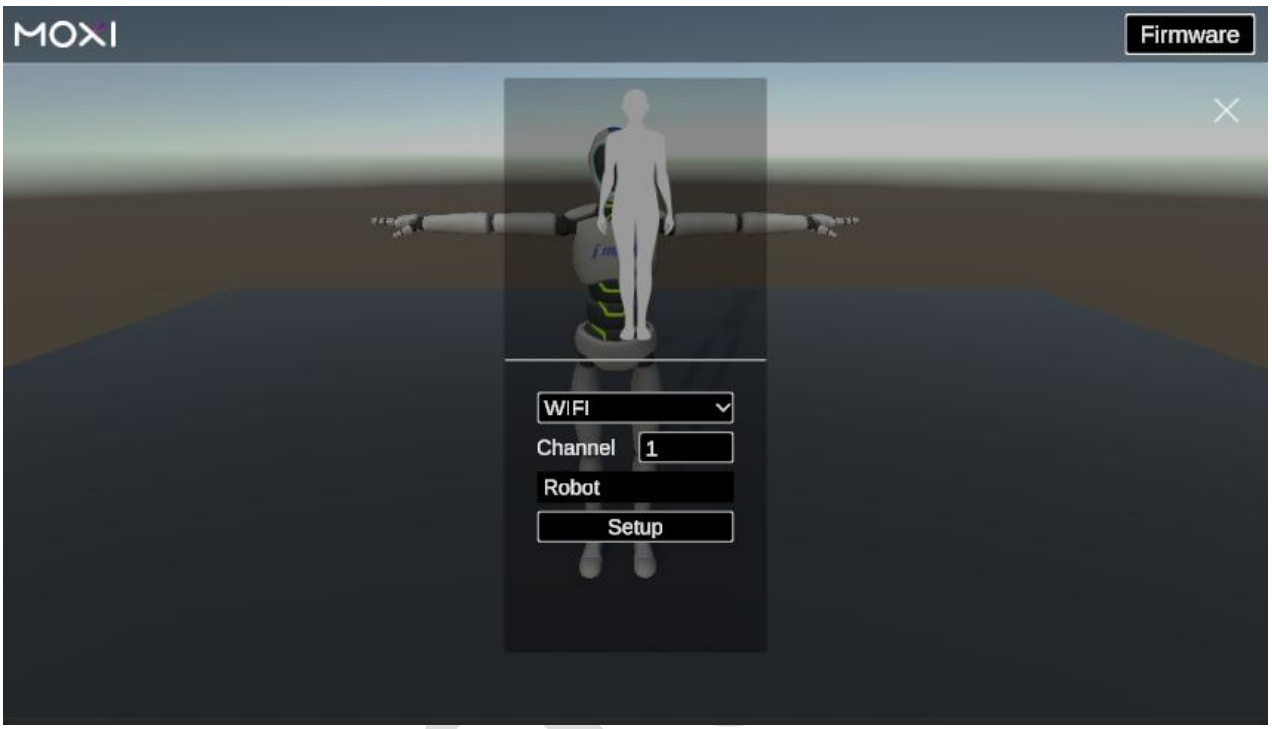
- Use Target Avatar
  - To specify an avatar using the channel ID.

- Move
  - To specify whether an avatar can move or not.
- Skeleton Mapping
  - The UI used for assigning the *MOXI* controllers to the bone joint of the target avatar.
    - The item names are the official name of the *MOXI* controllers, which are defined in `MOXI.h` in the *MOXI SDK*.
    - The value of each item is the assigned bone joint name of the target avatar.
    - The Checkbox is indicated that the controller is physically connected or not.



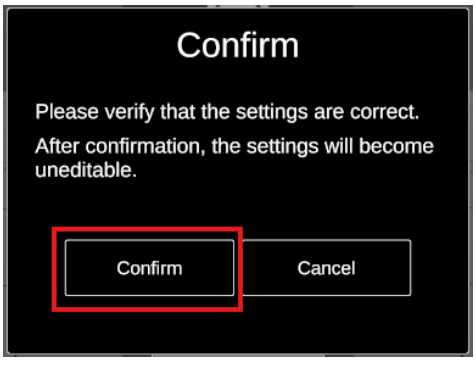
- Auto-Mapping
  - Automatically assign bone joint values in Skeleton Mapping according to the Humanoid Avatar System in Unity.
- Clear
  - Clear all assigned values for bone joints.
- Build Hierarchy
  - After assigning all the bone joints to the controllers, click this button to complete the jobs. The plugin will build a hierarchical structure of the bones controlled by the controllers and showing the result. The below image is the result of the hierarchy of the target avatar.

5. Tutorial of initializing the devices before doing the motion capture



- Step 1: Select the connection method on Connect UI

- Select the connection protocol : WIFI or USB
- Assign a channel ID
- Click "Start" button to be ready to start the initialization
- Before initializing, please double-check that the settings are correct. Once confirmed, click the "Confirm" button.



- Step 2 : Select the connection method on Connect UI

- Click "Connect" button to connect to the *MOXI* devices
- If the system is connected successfully, The plugin will open the “Calibration” button for next step.

- Step 3: Perform the calibration process

- Click "Calibration" button to do the hardware calibration.
  - Please refer the calibration step for more information.
- If the system completes the calibration job, the use can click the “Go Capture” button to the final step.

- Calibration Step:

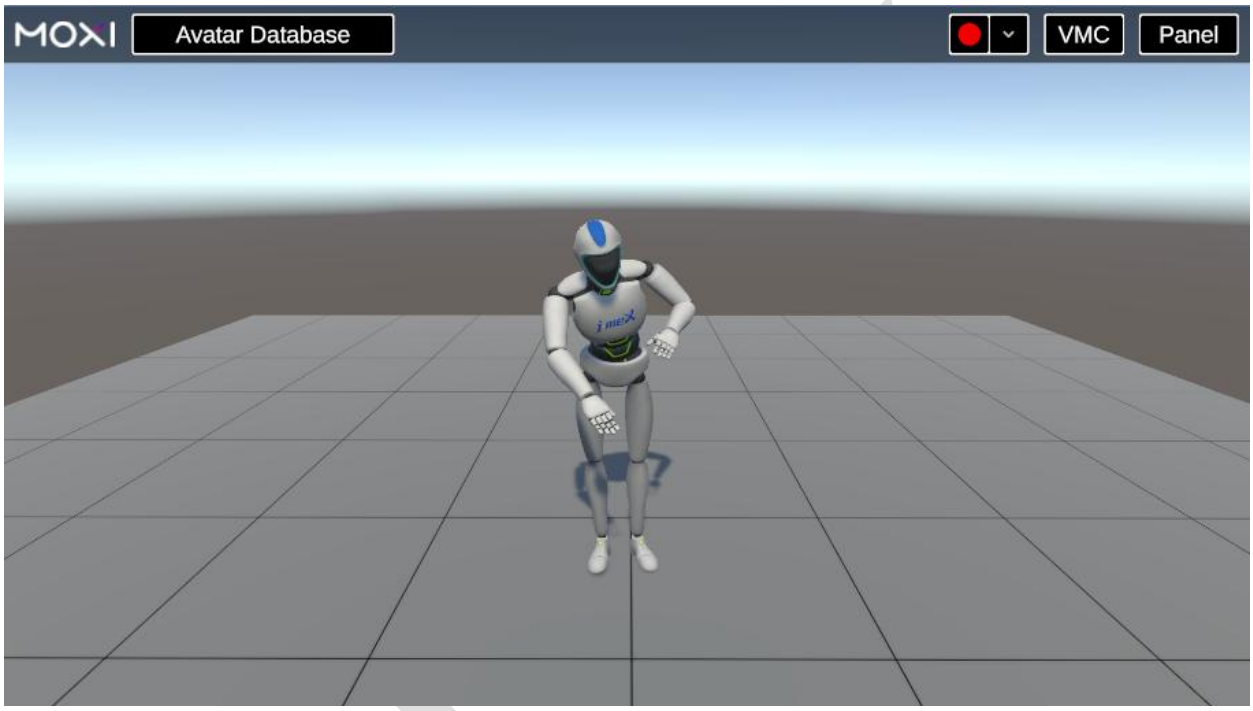
Calibrating devices on your body by moving and rotating the body parts until the target icon becomes green. then can go to the final step.

- Final Step: setup the natural pose of the target avatar

- Click "Setup Pose" button start the job of setting up the natural pose
- Before pressing the button, the user should ask the performer to perform the same pose of the target avatar in natural pose (which can be T-pose or A-pose). The system will record the pose as the rest pose of the motion capture. Be sure to have the same pose as the target avatar to get the better mocap result.
- The system will automatically move to motion capture mode after setting up the natural pose.

- Firmware
  - On the top-right corner of the page, there is a button, Firmware, for users to update the firmware of the devices. When pressing the button, a pop-up file browser for users to input the location of the firmware update file.
  - Follow the instruction of the browser to complete the update job.

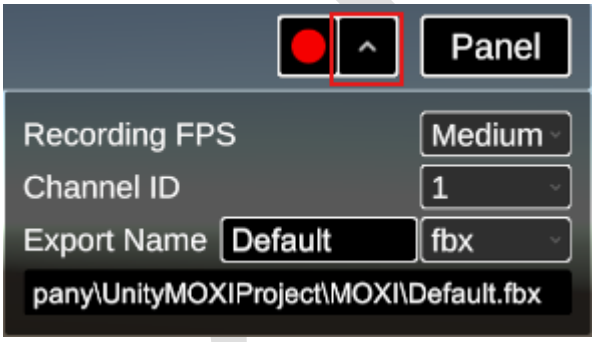
6. **Control View**



- Press WASD on the keyboard to move the view.
- Press Q to descend, press E to ascend
- Press Left Shift to accelerate movement.
- Right-click the mouse to rotate the view.

7. **Tutorial of recording the *i-mex* device or motion data**

- Record button with pull-down menu :
  - On the top of menu bar, there is a button with red dot, which is used for recording. Beside the button, there is a pull-down menu button to open a pop-up user interface to specify the recording functions.
  - The user interface is to set the recording frame rate, the channel ID, the file name and the type of motion data. The following image shows the layout of the user interface.



  - The recording job is running in another thread, of which the recording frame rate can be set independently to the rendering frame rate. There are three levels of frequency supported recently: High (100 fps), Medium (60 fps), and Low (30 fps).
  - The toolkit supports to record three types of motion data:
    - *Autodesk FBX* format for the full-body motion: indicates as the selection of ‘fbx’ on the user interface.
    - *Biovision BVH* format for the full-body motion: indicates as the selection of ‘bvh’ on the user interface.

- *j-mex MXM* format for the raw device data: indicates as the selection of ‘mxm’ on the user interface. To know the MXM format, please contact the technical support team of *j-mex*.

- Record button (the red dot button)
  - Hit Record button to record the motion data to the file.
  - Hit the Stop button to stop the recording. The Stop button is shown as a white square button.

- Panel button :
  - On the top of menu bar, there is a Panel button for users to turn ON or OFF the user interface to disconnect the devices.