

Mocap API Sample

2022/09/20 Reallusion Inc.

iClone Python API Resource

- [Main Page](#)
 - [IC Python API Wiki](#)
 - [GitHub](#)
 - [Forum](#)
-
- [Upgrade iClone7 Python Plugin to iClone8](#)

Two Ways to Run iClone Python Plugin

Your First iClone Python Plugin

- Run every time iClone is started

iClone Install Directory \ Bin64 \ OpenPlugin folder\ [Your Plugin Name]

- Run manually

Menu > Script > Load Python

How to Create User Interface

Using Pyside2 For Creating User Interface

- Create UI through Pyside2
- Using .ui file
- Using qml file

How to Trace Python Plugin (Using visual studio code)

1. Launch cmd and switch to the path of iClone `cd [iClonePath]\Bin64\`
2. Install pip `iClonepy .\scripts\Python\get-pip.py`
3. Install ptvsd `iClonepy -m pip install ptvsd`
4. Modify the launch.json in VSCode (As shown on the right)
5. Add the following code to the main program of your Plugin (ex. main.py).

```
import ptvsd
ptvsd.enable_attach(address=("localhost", 5678), redirect_output=True)
ptvsd.wait_for_attach()
```
6. iClone will be blocked after loading the Plugin,
Execute "Python:Attach" in VSCode to trace your python code.

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Python: Current File",
      "type": "python",
      "request": "launch",
      "program": "${file}",
      "console": "integratedTerminal"
    },
    {
      "name": "Python: Attach",
      "type": "python",
      "request": "attach",
      "port": 5678,
      "host": "localhost",
      "justMyCode": false
    }
  ]
}
```

How to Trace Python Plugin (Using visual studio 2019)

修改中 — Visual Studio Professional 2019 — 16.8.5

工作負載 個別元件 語言套件 安裝位置

Web 與雲端 (4)

 **ASP.NET 與網頁程式開發**

使用 ASP.NET Core, ASP.NET, HTML/JavaScript 及容器 (包括 Docker 支援) 建立 Web 應用程式。

 **Python 開發**

對 Python 進行編輯、偵錯、互動式開發及原始碼控制。

 **Azure 開發**

用於使用 .NET Core 和 .NET Framework 開發雲端應用程式及建立資源的 Azure SDK、工具及專案，並包含應用程式...

 **Node.js 開發**

使用非同步的事件驅動 JavaScript 執行階段 Node.js 建置可調整的網路應用程式。

傳統型與行動裝置 (5)

 **.NET 桌面開發**

使用 C#, Visual Basic 及 F#, 利用 .NET Core 和 .NET Framework 建置 WPF、Windows Forms 與主控台應用程式...

 **使用 C++ 的桌面開發**

使用您選擇的工具 (包括 MSVC、Clang、CMake 或 MSBuild)，建置適用於 Windows 的新式 C++ 應用程式。

 **通用 Windows 平台開發**

使用 C#, VB 或選用 C++，來建立適用於通用 Windows 平台的應用程式。

 **使用 .NET 進行行動開發**

使用 Xamarin 建置適用於 iOS、Android 或 Windows 的跨平台應用程式。

位置
C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional

繼續進行即表示您同意所選 Visual Studio 版本的**授權**。我們也可讓您使用 Visual Studio 下載其他軟體。此軟體為分發授權，如同**[協力廠商聲明](#)**或其隨附的授權中所述。繼續進行即表示您也同意該授權。

安裝詳細資料

- 包含
 - ✓ Python 語言支援
- 選擇性
 - ✓ Python miniconda
 - ✓ Python Web 支援
 - ✓ Python 3 64 位元 (3.7.8)
 - ✓ Live Share
 - ✓ **Python 原生開發工具**
 - ☐ Azure 雲端服務核心工具
 - ☐ Python 2 64-bit (2.7.18)
 - ☐ Python 3 32 位元 (3.7.8)
 - ☐ Python 2 32-bit (2.7.18)

個別元件

- ✓ .NET 5.0 執行階段
- ✓ .NET Core 3.1 執行階段 (LTS)
- ✓ .NET Framework 4.5 目標套件
- ✓ .NET Core 2.1 執行階段 (LTS)

Options

在下載時

Options

- Environment
- Projects and Solutions
- Source Control
- Work Items
- Text Editor
- Debugging
 - General
 - Just-In-Time
 - Output Window
 - Symbols**
- Performance Tools
- CMake
- Cookiecutter
- Database Tools
- Graphics Diagnostics
- IntelliCode
- Live Share

Symbols

Symbol file (.pdb) locations:

- ☒ Microsoft Symbol Servers
- ☐ NuGet.org Symbol Server
- ☒ C:\Q\5.15.2\msvc2019_64\bin
- ☒ D:\Dev\Release\HeadShot2.0\20230614.2\CCIC8Headshot2_20230614.2\Bin64
- ☒ D:\Dev\Release\HeadShot2.0\20230614.2\CCIC8Headshot2_20230614.2\Plugin
- ☒ **C:\Users\JohnnyYang.BEYOUNG\AppData\Local\Programs\Python\Python38**

Warning: Using symbol files from an unknown or untrusted location can be harmful to your computer.

Cache symbols in this directory:

C:\Users\JOHNNY~1.BEY\AppData\Local\Temp\SymbolCache

Browse...

Load all symbols

Empty Symbol Cache

Automatic symbol loading preference:

Facial Mocap API

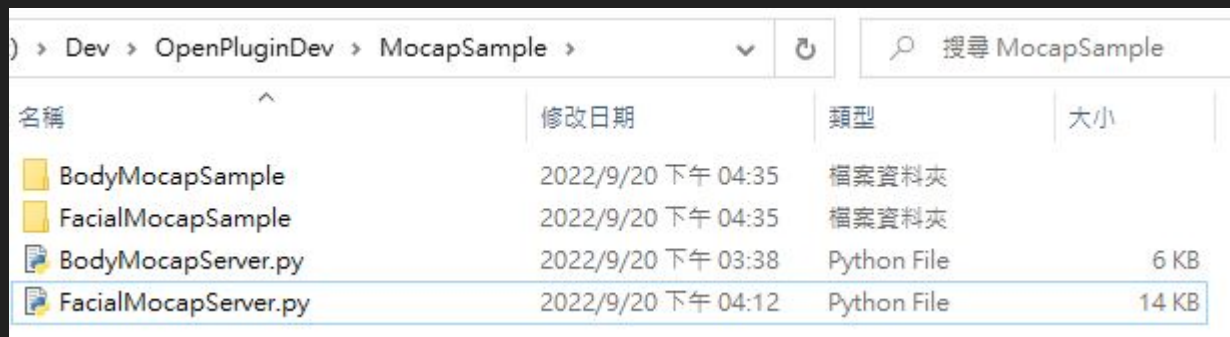
- [RI Mocap Manager](#)
- [RI Facial Device](#)
- [RI Facial Setting](#)

Body Mocap API

- [RMocapManager](#)
- [RBodyDevice](#)
- [RBodySetting](#)
- [RHandDevice](#)
- [RHandSetting](#)

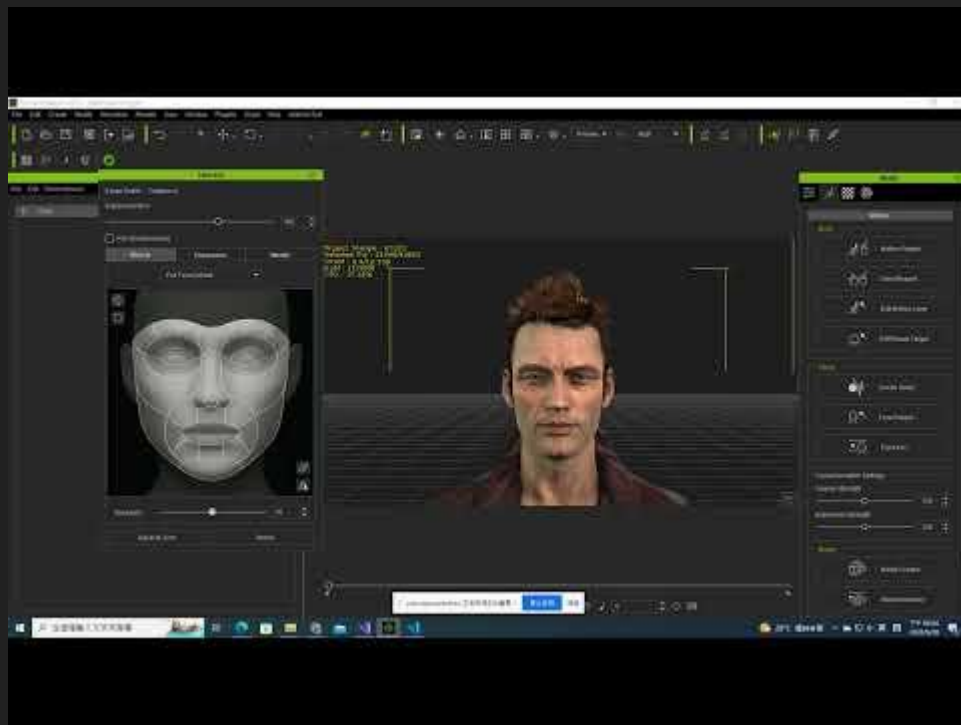
Mocap Sample

- FacialMocapServer.py Send facial data via TCP/IP
- [FacialMocapSample] Facial mocap iClone plugin
- BodyMocapServer.py Send body data via TCP/IP
- [BodyMocapSample] Body mocap iClone plugin



名稱	修改日期	類型	大小
BodyMocapSample	2022/9/20 下午 04:35	檔案資料夾	
FacialMocapSample	2022/9/20 下午 04:35	檔案資料夾	
BodyMocapServer.py	2022/9/20 下午 03:38	Python File	6 KB
FacialMocapServer.py	2022/9/20 下午 04:12	Python File	14 KB

Facial Mocap Sample



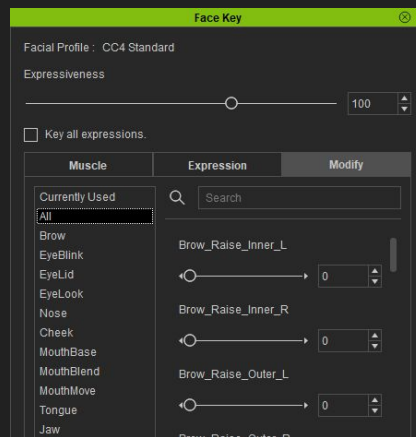
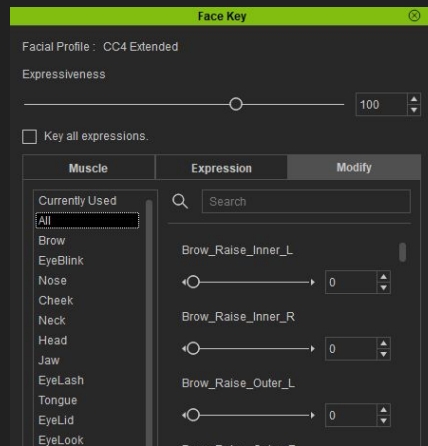
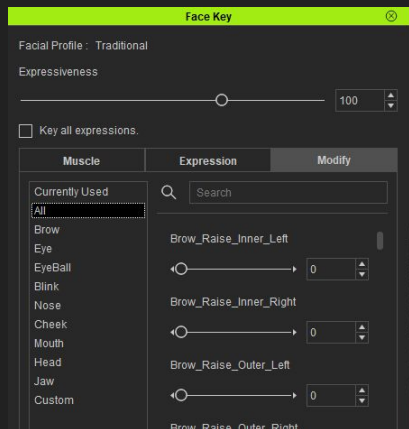
Customize Your Facial Mocap

- There are three Expression Sets for iClone characters:
 - **Traditional** (IC7 Standard): 143 blend shapes
 - **CC4 Standard**: 75 blend shapes
 - **CC4 Extended**: 160+ blend shapes (User can customize the expression)
- Using `face_component.GetExpressionSetUid()` to retrieve the type of expression set.
- `face_component.GetExpressionNames("", True)` can get the names of the expression,
This order is also the order of the second parameter `exp_list` while calling `facial_device.ProcessData()`.

IC7 Standard Exp Set

```
000: 'Brow_Raise_Inner_Left'
001: 'Brow_Raise_Inner_Right'
002: 'Brow_Raise_Outer_Left'
003: 'Brow_Raise_Outer_Right'
004: 'Brow_Drop_Left'
005: 'Brow_Drop_Right'
006: 'Brow_Raise_Left'
007: 'Brow_Raise_Right'
008: 'Eyes_Blink'
009: 'Eye_Blink_L'
010: 'Eye_Blink_R'
011: 'Eye_Wide_L'
012: 'Eye_Wide_R'
013: 'Eye_Squint_L'
014: 'Eye_Squint_R'
015: 'Nose_Scrunch'
016: 'Nose_Flanks_Raise'
017: 'Nose_Flank_Raise_L'
018: 'Nose_Flank_Raise_R'
019: 'Nose_Nostrils_Flare'
020: 'Cheek_Raise_L'
021: 'Cheek_Raise_R'
```

Tree types of expression sets



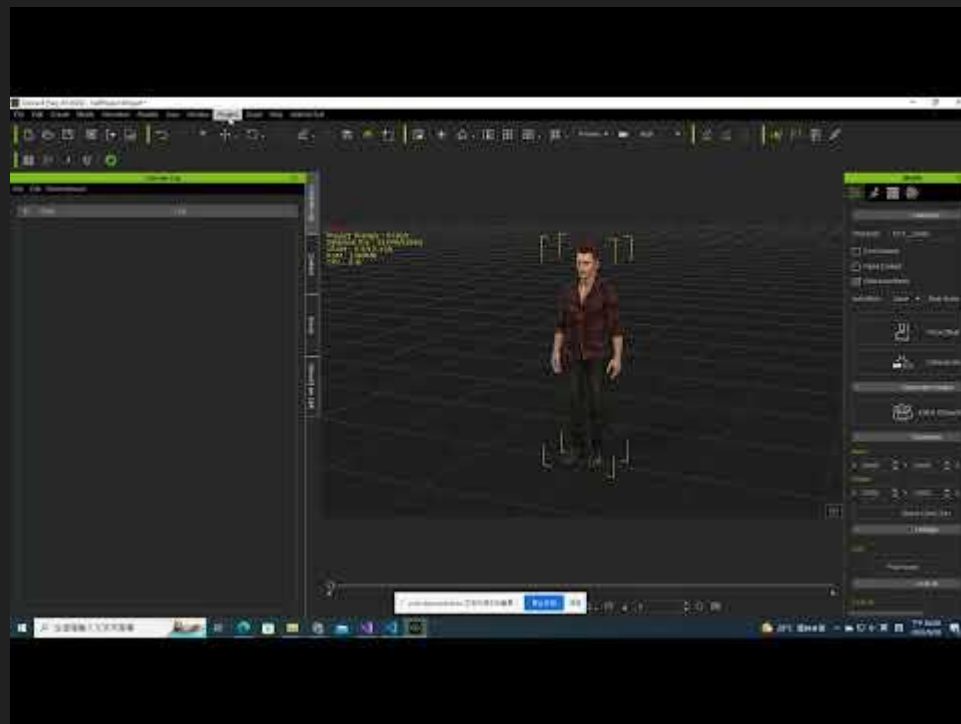
Customize Your Facial Mocap (FacialMocap/main.py)

Modify `NetworkEventCallback.OnDataReceived(self)`

- Receive your blend shape data.
- Mapping your blend shape to iClone's expression set.
- `facial_device.ProcessData(mocap_avatar, exp_list)`

`exp_list` is a float list, Represents the intensity value of the expression (-0.0~1.0+),
The order of the list can be retrieved by `face_component.GetExpressionNames("", True)`.

Body Mocap Sample



Customize Your Body Mocap (BodyMocap/mi

- Define your skeleton `bone_list`
 - bone name
 - parent bone name
 - Corresponding Hik bone name
- Call `body_device.Initialize(bone_list)`

```
bone_list = [hips, rightupleg, rightleg, rightfoot, leftupleg, leftleg, leftfoot, spine, spine1, spine2, spine3, neck, head, rightshoulder, rightarm, rightforearm, righthand, righthandthumb1, righthandthumb2, righthandthumb3, rightinhandindex, righthandindex1, righthandindex2, righthandindex3, rightinhandmiddle, righthandmiddle1, righthandmiddle2, righthandmiddle3, rightinhandring, righthandring1, righthandring2, righthandring3, rightinhandpinky, righthandpinky1, righthandpinky2, righthandpinky3, leftshoulder, leftarm, leftforearm, lefthand, lefthandthumb1, lefthandthumb2, lefthandthumb3, leftinhandindex, lefthandindex1, lefthandindex2, lefthandindex3, leftinhandmiddle, lefthandmiddle1, lefthandmiddle2, lefthandmiddle3, leftinhandring, lefthandring1, lefthandring2, lefthandring3, leftinhandpinky, lefthandpinky1, lefthandpinky2, lefthandpinky3]
```

```
#{bone_name, parent_bone_name, Hik_bone_name}
hips = ["hips", "", "Hips"]
rightupleg = ["rightupleg", "hips", "RightUpLeg"]
rightleg = ["rightleg", "rightupleg", "RightLeg"]
rightfoot = ["rightfoot", "rightleg", "RightFoot"]
leftupleg = ["leftupleg", "hips", "LeftUpLeg"]
leftleg = ["leftleg", "leftupleg", "LeftLeg"]
leftfoot = ["leftfoot", "leftleg", "LeftFoot"]
spine = ["spine", "hips", "Spine"]
spine1 = ["spine1", "spine", "Spine3"]
spine2 = ["spine2", "spine1", "Spine6"]
spine3 = ["spine3", "spine2", "Spine9"]
neck = ["neck", "spine3", "Neck"]
head = ["head", "neck", "Head"]
rightshoulder = ["rightshoulder", "spine3", "RightShc
rightarm = ["rightarm", "rightshoulder", "RightArm"]
rightforearm = ["rightforearm", "rightarm", "RightFor
righthand = ["righthand", "rightforearm", "RightHand"]
righthandthumb1 = ["righthandthumb1", "righthand", "R
righthandthumb2 = ["righthandthumb2", "righthandthumb
righthandthumb3 = ["righthandthumb3", "righthandthumb
ex = ["rightinhandindex", "righthand",
1 = ["righthandindex1", "rightinhandinc
2 = ["righthandindex2", "righthandindex
```


Customize Your Body Mocap (HIK bone name)

"Hips",	"Spine3",	"HipsTranslation",	"LeftHandExtraFinger4",	"RightHandExtraFinger4",	"LeftFootExtraFinger4",	"RightFootExtraFinger4",
"LeftUpLeg",	"Spine4",	"LeftHandThumb1",	"RightHandThumb1",	"LeftFootThumb1",	"RightFootThumb1",	"LeftInHandThumb",
"LeftLeg",	"Spine5",	"LeftHandThumb2",	"RightHandThumb2",	"LeftFootThumb2",	"RightFootThumb2",	"LeftInHandIndex",
"LeftFoot",	"Spine6",	"LeftHandThumb3",	"RightHandThumb3",	"LeftFootThumb3",	"RightFootThumb3",	"LeftInHandMiddle",
"RightUpLeg",	"Spine7",	"LeftHandThumb4",	"RightHandThumb4",	"LeftFootThumb4",	"RightFootThumb4",	"LeftInHandRing",
"RightLeg",	"Spine8",	"LeftHandIndex1",	"RightHandIndex1",	"LeftFootIndex1",	"RightFootIndex1",	"LeftInHandPinky",
"RightFoot",	"Spine9",	"LeftHandIndex2",	"RightHandIndex2",	"LeftFootIndex2",	"RightFootIndex2",	"LeftInHandExtraFinger",
"Spine",	"Neck1",	"LeftHandIndex3",	"RightHandIndex3",	"LeftFootIndex3",	"RightFootIndex3",	"RightInHandThumb",
"LeftArm",	"Neck2",	"LeftHandIndex4",	"RightHandIndex4",	"LeftFootIndex4",	"RightFootIndex4",	"RightInHandIndex",
"LeftForeArm",	"Neck3",	"LeftHandMiddle1",	"RightHandMiddle1",	"LeftFootMiddle1",	"RightFootMiddle1",	"RightInHandMiddle",
"LeftHand",	"Neck4",	"LeftHandMiddle2",	"RightHandMiddle2",	"LeftFootMiddle2",	"RightFootMiddle2",	"RightInHandRing",
"RightArm",	"Neck5",	"LeftHandMiddle3",	"RightHandMiddle3",	"LeftFootMiddle3",	"RightFootMiddle3",	"RightInHandPinky",
"RightForeArm",	"Neck6",	"LeftHandMiddle4",	"RightHandMiddle4",	"LeftFootMiddle4",	"RightFootMiddle4",	"RightInHandExtraFinger",
"RightHand",	"Neck7",	"LeftHandRing1",	"RightHandRing1",	"LeftFootRing1",	"RightFootRing1",	"LeftInFootThumb",
"Head",	"Neck8",	"LeftHandRing2",	"RightHandRing2",	"LeftFootRing2",	"RightFootRing2",	"LeftInFootIndex",
"LeftToeBase",	"Neck9",	"LeftHandRing3",	"RightHandRing3",	"LeftFootRing3",	"RightFootRing3",	"LeftInFootMiddle",
"RightToeBase",	"LeftUpLegRoll",	"LeftHandRing4",	"RightHandRing4",	"LeftFootRing4",	"RightFootRing4",	"LeftInFootRing",
"LeftShoulder",	"LeftLegRoll",	"LeftHandPinky1",	"RightHandPinky1",	"LeftFootPinky1",	"RightFootPinky1",	"LeftInFootPinky",
"RightShoulder",	"RightUpLegRoll",	"LeftHandPinky2",	"RightHandPinky2",	"LeftFootPinky2",	"RightFootPinky2",	"LeftInFootExtraFinger",
"Neck",	"RightLegRoll",	"LeftHandPinky3",	"RightHandPinky3",	"LeftFootPinky3",	"RightFootPinky3",	"RightInFootThumb",
"LeftFingerBase",	"LeftArmRoll",	"LeftHandPinky4",	"RightHandPinky4",	"LeftFootPinky4",	"RightFootPinky4",	"RightInFootIndex",
"RightFingerBase",	"LeftForeArmRoll",	"LeftHandExtraFinger1",	"RightHandExtraFinger1",	"LeftFootExtraFinger1",	"RightFootExtraFinger1",	"RightInFootMiddle",
"Spine1",	"RightArmRoll",	"LeftHandExtraFinger2",	"RightHandExtraFinger2",	"LeftFootExtraFinger2",	"RightFootExtraFinger2",	"RightInFootRing",
"Spine2",	"RightForeArmRoll",	"LeftHandExtraFinger3",	"RightHandExtraFinger3",	"LeftFootExtraFinger3",	"RightFootExtraFinger3",	"RightInFootPinky",

Customize Your Body Mocap (BodyMocap/main.py)

- Define your mocap coordinate system:
 - Coordinate axis: `device_setting.SetMocapCoordinate(Up-axis, Front-axis, RightHand/LeftHand)`
 - Coordinate offset: `device_setting.SetCoordinateOffset(0, [0, 0, 0])`
- Define the data format of mocap (Pass to SetTPoseData() and ProcessData())
 - **Position format** (`position_setting`)
Local/World: `position_setting.SetCoordinateSpace(RLPy.ECoordinateSpace_Local)`
Position unit: `position_setting.SetUnit(RLPy.EPositionUnit_Centimeters)`
 - **Rotation format** (`rotation_setting`)
Local/World: `rotation_setting.SetCoordinateSpace(RLPy.ECoordinateSpace_Local)`
Rotation representation: `rotation_setting.SetType(RLPy.ERotationType_Euler)`
`rotation_setting.SetEulerOrder(RLPy.EEulerOrder_ZXY)`
Rotation unit: `rotation_setting.SetUnit(RLPy.ERotationUnit_Degrees)`

Customize Your Body Mocap (BodyMocap/main.py)

- Define format of mocap data (pass to SetTPoseData() and ProcessData())

ex: `position_setting.SetCoordinateSpace(RLPy.ECoordinateSpace_Local)`
`position_setting.SetUnit(RLPy.EPositionUnit_Centimeters)`
`rotation_setting.SetCoordinateSpace(RLPy.ECoordinateSpace_Local)`
`rotation_setting.SetType(RLPy.ERotationType_Euler)`
`rotation_setting.SetEulerOrder(RLPy.EEulerOrder_ZXY)`
`rotation_setting.SetUnit(RLPy.ERotationUnit_Degrees)`

```
t_pose_data = [
```

```
    0.0, 105.85, 0.0,  
    -11.5, -1.85, 0.0,  
    0.0, -48.0, 0.0,  
    0.0, -48.0, 0.0,  
    11.5, -1.85, 0.0,  
    0.0, -48.0, 0.0,
```

```
...]
```

Local Position X,Y,Z (cm)

```
    0, 0, 0,  
    0, 0, 0,  
    0, 0, 0,  
    0, 0, 0,  
    0, 0, 0,  
    0, 0, 0,
```

Local Rotation in Euler Z,X,Y (degree)

← hips
← rightupleg
← rightleg
← rightfoot
← leftupleg
← leftleg

```
bone_list = [hips, spine, spine1, spine2,
leftshoulder, leftarm, leftforearm,lefthand,
rightshoulder, rightarm, rightforearm, righthand,
neck, head,
leftupleg, leftleg, leftfoot, lefttoebase,
rightupleg, rightleg, rightfoot, righttoebase]
```

```
t_pose_data = [
```

Local Position X,Y,Z (cm)	Local Rotation in Quaternion X,Y,Z,W	
0.0, 105.85, 0.0,	0, 0, 0, 1,	← hips
-11.5, -1.85, 0.0,	0, 0, 0, 1,	← spine
0.0, -48.0, 0.0,	0, 0, 0, 1,	← spine1
0.0, -48.0, 0.0,	0, 0, 0, 1,	← spine2
11.5, -1.85, 0.0,	0, 0, 0, 1,	← leftshould
0.0, -48.0, 0.0,	0, 0, 0, 1,	← leftarm
0.0, 105.85, 0.0,	0, 0, 0, 1,	
-11.5, -1.85, 0.0,	0, 0, 0, 1,	
0.0, -48.0, 0.0,	0, 0, 0, 1,	← ..
0.0, -48.0, 0.0,	0, 0, 0, 1,	
11.5, -1.85, 0.0,	0, 0, 0, 1,	
0.0, -48.0, 0.0,	0, 0, 0, 1]	← righttoebase