Table of Contents

产品简介 1.1
- Motionverse 1.1.1

API 2.1
- 接入前准备 2.1.1
- 功能对接设计 2.1.2
- 获取Token 2.1.3
- 声音文件驱动数字人（播报类型） 2.1.4
- 文字驱动数字人（播报类型） 2.1.5
- NLP文字驱动数字人（问答类型） 2.1.6

插件 2.2
- 说明 2.2.1
- 接入前准备 2.2.2
- UE插件对接 2.2.3
  - 概述 2.2.3.1
  - 对接说明 2.2.3.2
- U3D插件使用（快速驱动数字人） 2.2.4
  - 概述 2.2.4.1
  - 对接说明 2.2.4.2
- 小程序插件对接 2.2.5
  - 概述 2.2.5.1
  - 对接说明 2.2.5.2

WebView 2.3
- 说明 2.3.1
- 接入前准备 2.3.2
- Web接入 2.3.3
- iOS接入 2.3.4
- 安卓接入 2.3.5

DM身体绑定规范 3.1

Motionverse SDK产品介绍

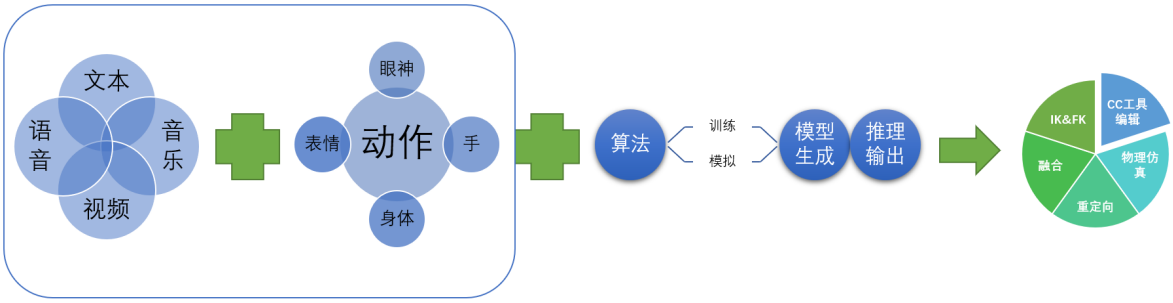
> V3.0 2023-6-8

一、Motionverse 简介

Motionverse是一款多模态实时驱动虚拟数字人的业务中台，主要面向有二次开发能力的行业开发者，用AI动态生成的方案，提供SDK和管理后台，解决产品和终端的虚拟数字人驱动问题。 元宇宙世界中，每个人都有自己的数字分身，也会有大量的AI数字人做为新的交互方式出现，虚拟数字人的数量将会数倍于人类的数量，传统的动作制作和动作捕捉技术方案，已经无法满足数以百亿计的动作表情需求，通过各种不同的输入情况，智能的实时的来生成所需要的动作表情和口型，这个就是Motionverse定义的多模板实时驱动解决方案。 Motionverse使用人工智能技术，利用高质量、大规模数据集，以音频源、文本源输入、传感器输入、语义输入、脚本输入等等，通过AI生成数字人的身体动作数据、面部表情数据，以及口型数据。 Motionverse做为中台产品，以API/SDK的方式，开放性的向行业提供解决方案，无论是元宇宙社区，还是新添数字人的App，或者线下大屏，VR/AR设备，只要有数字人的地方，就可以让驱动起来变得更简单，更实时，更低成本。

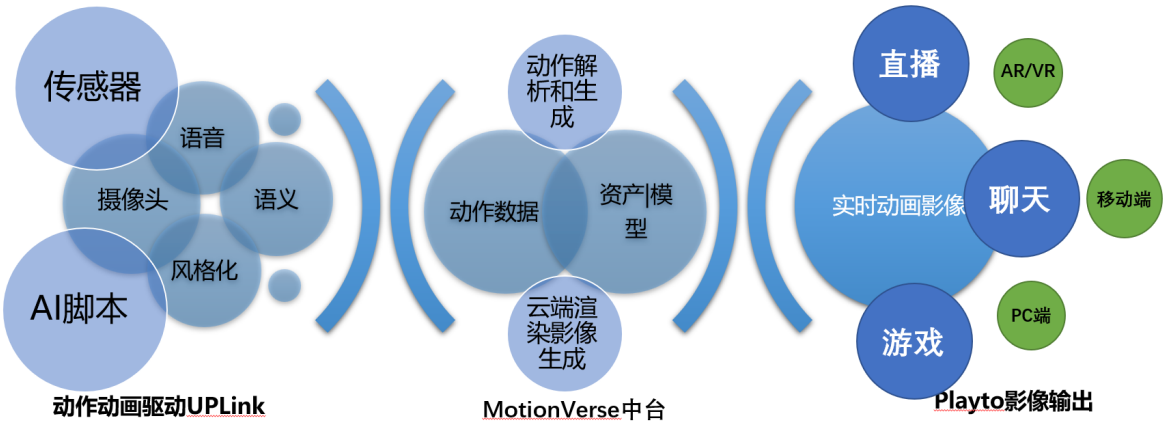
二、Motionverse产品能力原理

1、核心AI生成能力- 动作表情生成（Audio to Motion） AI生成动作表情和口型，是基于机器学习的神经网络模型。这种方法不需要针对数据提取特定类型的音素，得益于神经网络的强大的函数拟合能力，可以直接将音频数据映射到表情、动作和口型中。 对于使用神经网络的方案来说，最重要的是选择合理的数据表示以及网络结构，我们使用 mesh 的方法来进行动作和表情的表示，网络结构则使用卷积神经网络。 卷积神经网络已经在计算机视觉领域证明了它提取特征的强大能力，而语音特征转换为频谱后可以用卷积神经网络来处理，用于处理语音频谱特征，较好的解决了时序信息依赖和生成模式单一的问题。 对于表情和口型来说，使用 51 个 blendershape数值来表示脸部的动作，神经网络的输出是这 51 个 blendershape 的权重，通过组合这些 blendershape 的值，可以得到脸部表情和口型合理表示。对于身体动作来说，模型输出的是关节点的旋转，这里采用四元数来表示，在使用时，将模型输出的值按特定的顺序赋给角色骨骼即可。 多模态风格化动作数据解析和生成依赖于高质量大规模数据集，随着搭建精巧高效深度学习网络结构和不停录入训练数据，基于 transformer 架构的生成网络非常便捷高效的完成了从音频到动作、表情和口型的任务。

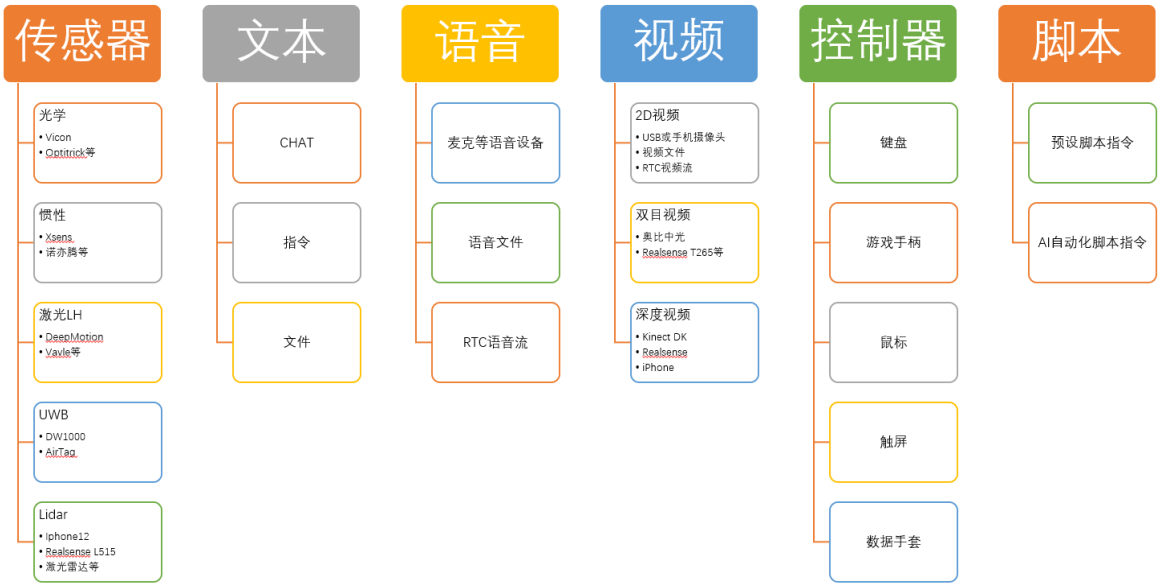


2、云渲染能力 Motionverse除了大量AI能力外，因为其围绕的3D数字人领域展开服务，必不可少的提供了3D渲染相关的能力。云渲染把渲染任务提交到云端，云端的服务器会来完成渲染工作，整个渲染不占用本地的任何算力，对本地系统资源没有任何要求，还提高了工作效率，让实时渲染终究可以跨越引擎、版本以及渲染模式，随用随取，多设备、多显卡、多服务器自动分配，自动均衡。

三、Motionverse产品架构



上图表述了Motionverse的产品中台结构，多模态的输入后，进入到Motionverse平台，通过AI Creator的过程，得到生成的数据，提供给各个产品和终端，用于驱动虚拟数字人。 其多模态输入方式，包含可能的各种输入，包括语音、文本、传感器、控制器、脚本等等多种方式

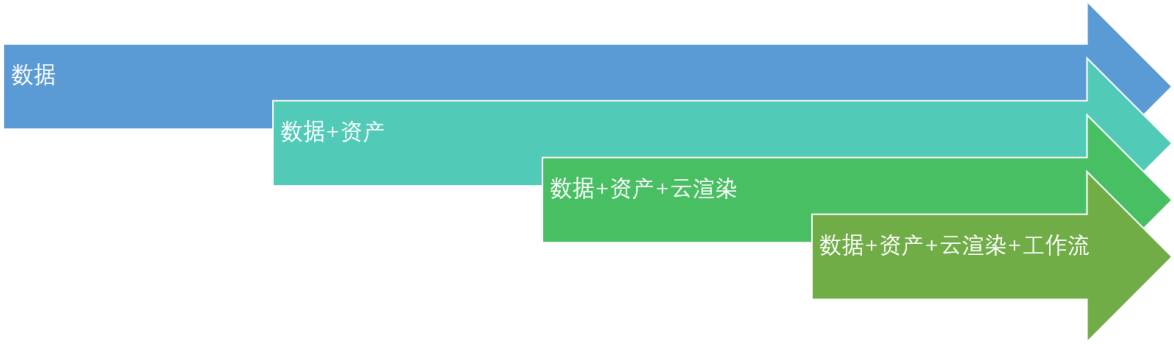


下述列举一些常见的输入到生成的使用场景　输入一段语音文件，中台生成并返回动作数据文件　通过麦克风实时输入语音，中台实时返回动作数据流　输入一段文本，中台通过AI转化合成为语音，并返回动作数据文件　通过6个传感器，输入传递头、手、脚的信息，中台生成全身的动作　通过3个传感器，输入头和手的信息，中台猜测并生成全身动作　通过输入视频文件，中台提取视频，生成动作　通过输入控制器指令，中台生成对应的动作和表情

基于Motionverse的产品架构，通过多模态的输入信息，进行动作表情生成，结合各种数字模型，可以通过云端渲染完成最终输出。Motionverse在接收到输入后，可以完成从数据生成，到渲染相关的数字人整个工作流

四、Motionverse四个层次的能力输出

Motionverse在接收到输入后，可以完成从数据生成，到渲染相关的数字人整个工作流。开发者可以轻松从中获取



1、动作表情生成数据输出 适用于已有数字人模型的产品，直接管理场景，应用数据，使用Motionverse最核心的能力，获取动作和表情数据。 有播报和问答两个场景，播报是根据输入的内容，进行播报动作、口型和表情，并播放对应语音。问答是根据配置好的问答库，根据输入得到答案，并生成动作、口型和表情，并播放对应答案的语音。

2、数字人模型 + 动作表情生成数据输出 适用于没有自己数字人模型的产品，可以使用Motionverse捏脸工具或者模型库的方式，从Motoinverse的资产中，获取对应的数字人模型，然后再获取对应的驱动数据。 整个模型的驱动数据来自于Motionverse中台，渲染是在本地利用本地自身算力完成。

3、数字人模型 + 动作表情生成数据 + 云渲染输出 适用于本地算力不足，或者显示质量非常高的如超写实角色的使用场景。 在本层次输出中，一方面模型和驱动数据都来源于Motionverse中台，另外一方面，也使用了Motionverse的云渲染能力，利用云端的渲染资源，完成了驱动和渲染所有工作，本地端

直接拉取视频流即可。

4、定制工作流 根据用户需求，完全自定义定制独立的工作流

接入流程

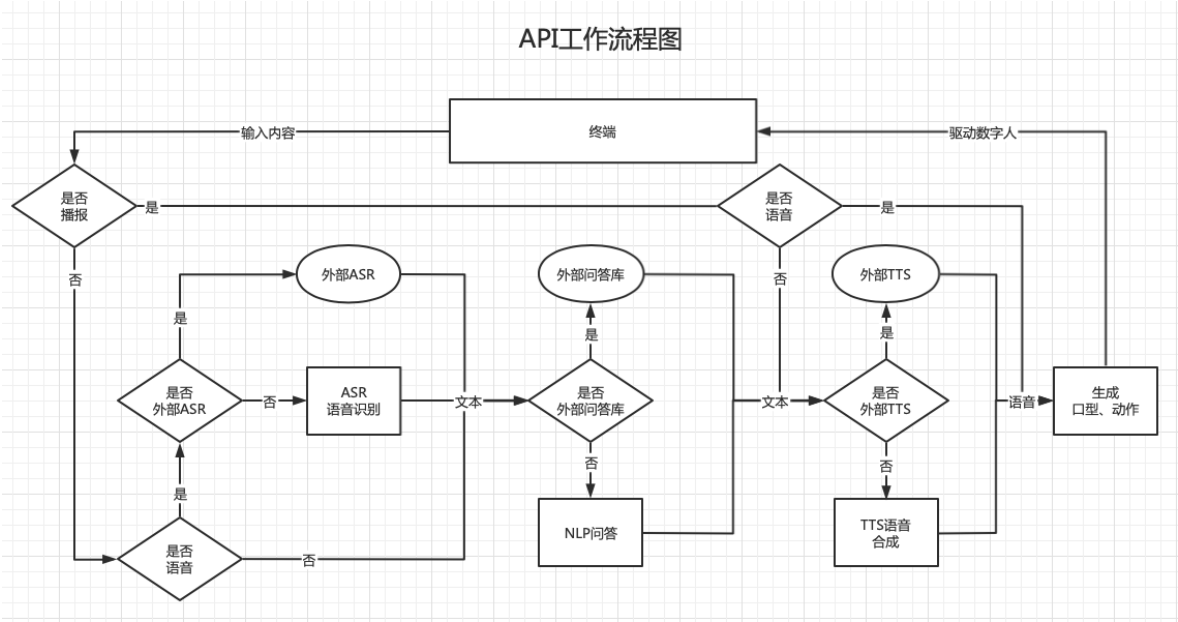
1、首先登录Motionverse管理平台创建应用

2、获取当前应用的appid，secret

3、通过应用appid，secret换取token，验证权限

4、携带获取的token请求API接口，获取数据

API工作流程图


API工作流程图
终端
输入内容
驱动数字人
是否
播报
是
否
是否
语音
是
否
外部ASR
外部问答库
外部TTS
是
是
是
是否
外部ASR
否
ASR
语音识别
文本
是否
外部问答库
文本
是否
外部TTS
语音
生成
口型、动作
否
否
是
NLP问答
TTS语音
合成
是否
语音
否

获取 Token

> token既是用户的权限代码，接下来所有的请求接口headers都需要携带token码

请求地址： `https://motionverseapi.deepscience.cn/users/getAppToken`

请求方式： `GET`

请求参数：

名称	类型	必填	描述	示例值
appid	string	true	应用id	-
secret	string	true	应用秘钥	-
timestamp	number	true	当前时间戳	1662431431712
sign	string	true	通过安全哈希算法生成签名	3a69b981f6fc0597aa3df1c387ae22a43a8fd249

生成签名示例代码：

```
let appid = '填入appid'
let secret = '填入app secret'
let timestamp = new Date().getTime() // 当前时间戳
let str = appid + timestamp + secret // 组合字符串
let sign = sha1(str) // 通过安全哈希算法生成签名
```

返回参数：

名称	类型	描述	示例值
code	number	响应码	0
data ▼	object	响应数据	-
access_token	string	token 值	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2MzEwNjgxOGJhYjE2NDRjM2VkNjBjZDYiLCJhcHBpZCI6IjE2NjIxMDY0MTg0NjE1MjI3NiIsImlhdCI6MTY2MjQ0NjYwMiwiZXhwIjoxNjYyNTMzMDAyfQ.1LujsajGwrw0MdZKCOCrWp83bMRRBtt5qW-jVJcR-6I
msg	string	响应消息	ok

返回示例：


```
{
  "code":0,
  "msg":"ok",
  "data":
    {
     "access_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2MzEwNjgxOGJhYjE2NDRjM2VkNjBjZDYiL
    }
}
```


Token使用

1、设置接口请求headers.access_token值为当前获取的token。

2、token有效期为24小时

声音文件驱动数字人（播报类型）

> 通过声音文件，生成口型、身体动作信息，身体动作可以根据实际情形选择是否生成。

请求地址：`https://motionverseapi.deepscience.cn/v3.1/api/voiceBroadcastMotion`

请求方式：`POST`

请求类型：`application/json`

请求参数：

名称	类型	必填	描述	示例值
audio_url	string	是	语音文件链接(.wav, .mp3)或者内容为语音base64 的文件链接(.json)	http://ds-model-tts.oss-cn-beijing.aliyuncs.com/temp/166320502925538069.wav http://ds-model-tts.oss-cn-beijing.aliyuncs.com/temp/166320502925538069.json
audio_format	string	否	表示语音的类型；若audio_format 为”wav”，则表示语音数据为 wav 格式。若为“mp3”则语音数据是MP3 格式；若为”base64”，则表示语音数据为 base64 格式；	默认为”wav”
is_cache	int	否	表示表情及动作是否进行缓存，此项为全局缓存，会同时针对表情及动作进行作用。1 为缓存，0 为不缓存；	默认为 0
face_config ▼	object	否	表情请求参数设置 **（注：如果使用默认视素驱动效果，则可传这部分参数。默认的视素驱动效果更佳。）**	{}
face_type	number	否	表情类型选择，1 为语音驱动表情，此时face_gender 和face_tag 起作用。2 为视素驱动，此时face_gender 和face_tag 不起作用。 3、MetaHuman 驱动。 4、表情生成驱动	默认为 2

名称	类型	必填	描述	示例值
face_gender	string	否	可以设定表情性别，性别不同表情效果不同。性别 male 男性 女性 female	默认为 male， 当 face_type 为 1 时，该项才起作用。
face_tag	number	否	可以调整表情的情绪。0-综合 1-高兴 2-悲伤 3-生气 4-惊讶 5-中立 6-激动 7-沮丧 8-恐惧 9-厌恶	默认值为 0， 当 face_type 为 1 时，该项才起作用。
face_stretch_scale	number	否	face_type 为 2、3 时，表示希望嘴部发‘a’的音的夸张的程度，值越大在发’a’时嘴越夸张；范围[1.0, 3.0]	1.5
face_mouth_divisor	number	否	face_type 为 2、3 时，表示整体的嘴部的幅度，值越大嘴部整体幅度越小，范围[0.5, 3]	1.6
face_smile_template	number	否	微笑模板，范围[0, 80] 只有 face_type 为 4 时生效	40
body_config ▼	object	否	动作请求参数设置	{}
body_motion	number	否	身体动作，0=不生成身体动作 1=自然闲聊 2=通用演讲 3=通用讲解 4=丰富演讲 5=金姐演讲 6=可爱女生 7=站立直播 8=标准客服 9=坐姿直播 10=主持人风格	默认为空
body_filter_window	number	否	表示动作自然度 数值越大自然效果越明显，但细节损失也会越多。范围 3-31，必须为奇数；此项必须与 order 搭配使用。非必须请默认此项。	默认值 15

名称	类型	必填	描述	示例值
body_filter_order	number	否	动作自然阶数，范围 2-4，数值越小阶数越高；此项必须与 window 搭配使用。非必须请默认此项。	默认值 3
body_fixed	bool	否	表示获取动作数据是半身还是全身。如果是 True 就直接获取上半身数据，False 直接获取全身数据；	默认 false
body_head_x_rot	number	否	用于调整头部的仰俯角度。范围为-180 度 ~ 180 度；正数表示低头，负数表示抬头	默认值 0
body_compress	bool	否	表示返回的 oss_url 的内容是否是二进制；若为 True，生成的是二进制格式，二进制格式传输速度更快；若为 False，生成的是可读的 json 格式；	默认为 false

返回参数：

名称	类型	描述	示例值
code	number	响应码	200
msg	string	提示信息	正常返回”ok”，错误返回具体错误信息。
data ▼	object	返回数据	{}
allfaceData ▼	object	表情动作数据	
oss_url	string	表情动作数据的url	href="http://xxx.xxx.json" http://xxx.xxx.json
time_log	string	流程中的时间戳日志	
audio_time	number	播报时长	"audio_time": 2.8875
motionFusionedData ▼	object	身体动作数据	
oss_url	string	身体动作数据的url	href="http://xxx.xxx.json" http://xxx.xxx.json
time_log	string	流程中的时间戳日志	
audio_time	number	播报时长	"audio_time": 2.8875

请求示例：

```
{
  "audio_url": "http://ds-model-tts.oss-cn-beijing.aliyuncs.com/temp/166320562819877046.wav",
  "body_config": { "body_motion": 1 }
}
```

返回示例：

```
{
  "code": 0,
  "msg": "ok",
  "data": {
    "allfaceData": {
      "oss_url": "https://ds-vhost-action-dev.oss-cn-beijing.aliyuncs.com/mouth/77db9e92-e385-4d19-8cce-b4c
      "time_log": {
        "all_time": 0.3748183250427246,
        "download_time": 0.0007715225219726562,
        "featextr_time": 0.19159865379333496,
        "inference_time": 0.020695924758911133,
        "upload_time": 0.16169285774230957
      },
      "audio_time": 6.9079375
    },
    "motionFusionedData": {
      "oss_url": "https://ds-vhost-action-dev.oss-cn-beijing.aliyuncs.com/body/77db9e92-e385-4d19-8cce-b4cf
      "time_log": {
        "all_time": 1.8889949321746826,
        "postprocess_motion_time": 0.000102996826171875,
        "query_audio_time": 0.11474728584289551,
        "synth_motion_time": 1.5677759647369385,
        "upload_time": 0.19675874710083008
      },
      "audio_time": 6.9079375
    }
  }
}
```


视素请求示例：

```
{
  "upload_type": 1,
  "viseme_list": [
    { "beginTime": 0, "id": "21", "ipa": "p,b,m" },
    { "beginTime": 75, "id": "4", "ipa": "\u025b,\u028a" }
  ]
}
```

视素返回示例：

```
{
  "audio_time": 26.233333333333334,
  "code": 0,
  "face_data": null,
  "oss_url": "https://phoneme-mouth-bs.oss-cn-beijing.aliyuncs.com/mouth/0c70e00f-9114-41ac-bea5-ad52eac737
  "time_log": {
    "all_time": 0.37830376625061035,
    "synth_face_time": 0.12147140502929688,
    "upload_time": 0.25348329544067383
  },
  "version": "v2.0.0"
}
```

文字驱动数字人（播报类型）

> 输入一段文字信息，通过 TTS 转化为声音信息，并生成口型、身体动作信息，身体动作可以根据实际情形选择是否生成。

请求地址：`https://motionverseapi.deepscience.cn/v3.0/api/textBroadcastMotion`

请求方式：`POST`

请求类型：`application/json`

请求参数：

名称	类型	必填	描述	示例值
draft_content	string	是	文本	您好
audio_format	string	否	表示语音的类型；若 audio_format 为”wav”，则表示语音数据为 wav 格式。若为“mp3”则语音数据是 MP3 格式；若为”base64”，则表示语音数据为 base64 格式；	默认为”wav”
tts_config ▼	object	否	tts 请求参数设置	{}
tts_voice_name	string	否	声音角色，可供选择的有 声音列表	默认值 aixia
tts_speed	number	否	声音的速度	默认值 50，范围 0-100；face_type 为 2 的时候，建议此项数值设定为 40；
tts_volume	number	否	声音的音量	默认值 50，范围 0-100
tts_fm	number	否	声音的音调	默认值 50，范围 0-100
face_config ▼	object	否	表情请求参数设置	{}
face_type	number	否	表情类型选择，1 为语音驱动表情，此时 face_gender 和 face_tag 起作用。2 为视素驱动，此时 face_gender 和 face_tag 不起作用。3、MetaHuman 驱动。4、表情生成驱动	默认为 2
face_gender	string	否	可以设定表情性别，性别不同表情效果不同。性别 male 男性 女性 female	默认为 male
face_tag	number	否	面部情绪 情绪 0-综合 1-高兴 2-悲伤 3-生气 4-惊讶 5-中立 6-激动 7-沮丧 8-恐惧 9-厌恶	默认值为 0

名称	类型	必填	描述	示例值
face_stretch_scale	number	否	face_type 为 2、3 时，表示希望嘴部发‘a’的音的夸张的程度，值越大在发’a’时嘴越夸张；范围[1.0, 3.0]	1.5
face_mouth_divisor	number	否	face_type 为 2、3 时，表示整体的嘴部的幅度，值越大嘴部整体幅度越小，范围[0.5, 3]	1.6
face_smile_template	number	否	微笑模板，范围[0, 80] 只有 face_type 为 4 时生效	40
body_config ▼	object	否	动作请求参数设置	{}
body_motion	number	否	身体动作，0=不生成身体动作 1=自然闲聊 2=通用演讲 3=通用讲解 4=丰富演讲 5=金姐演讲 6=可爱女生 7=站立直播 8=标准客服 9=坐姿直播 10=主持人风格	默认为空
body_filter_window	number	否	表示动作自然度 数值越大自然效果越明显，但细节损失也会越多。范围 3-31，必须为奇数；此项必须与 order 搭配使用。非必须请默认此项。	默认值 15
body_filter_order	number	否	动作自然阶数，范围 2-4，数值越小阶数越高；此项必须与 window 搭配使用。非必须请默认此项。	默认值 3
body_fixed_hips	bool	否	表示获取动作数据是半身还是全身。如果是 True 就直接获取上半身数据，False 直接获取全身数据；	默认 false
body_head_x_rot	number	否	用于调整头部的仰俯角度。范围为-180 度～ 180 度；正数表示低头，负数表示抬头	默认值 0
body_compress	bool	否	表示返回的 oss_url 的内容是否是二进制；若为 True，生成的是二进制格式，二进制格式传输速度更快；若为 False，生成的是可读的 json 格式；	默认为 false

返回参数:

名称	类型	描述	示例值
code	number	响应码	0
msg	string	提示信息	正常返回”ok”，错误返回具体错误信息。
data▼	object		
ttsSynthesizeData▼	object	tts 语音合成数据	
audio_url	string	生成的音频地址	href="http://xxx.xxx.wav" http://xxx.xxx.wav
allfaceData▼	object	表情动作数据	
oss_url	string	bs 的 url	href="http://xxx.xxx.json" http://xxx.xxx.json
time_log	string	流程中的时间戳日志	
audio_time	number	播报时长	"audio_time": 2.8875
motionFusionedData▼	object	身体动作数据	
oss_url	string	bs 的 url	href="http://xxx.xxx.json" http://xxx.xxx.json
time_log	string	流程中的时间戳日志	
audio_time	number	播报时长	"audio_time": 2.8875

声音角色列表

音库类型	发音人名称	发音人风格	发音人代码	发音人描述
精品音库	橙橙	默认	chengcheng-neutral-plus	知性大方
		开心	chengcheng-happy-plus	
		生气	chengcheng-angry-plus	
		厌恶	chengcheng-disgust-plus	
		害怕	chengcheng-fear-plus	
		难过	chengcheng-sad-plus	
		吃惊	chengcheng-amazed-plus	
	莎莎	默认	shasha-neutral-plus	亲和自然
		开心	shasha-happy-plus	
		生气	shasha-angry-plus	
		厌恶	shasha-disgust-plus	
		害怕	shasha-fear-plus	
		难过	shasha-sad-plus	
		惊喜	shasha-surprise-plus	
	晓迪	默认	xiaodi-plus	温柔舒缓
		温暖	xiaodi-warm-plus	
	Kiyo	元气	kiyo-plus	可爱女生
		职业	suren-plus	
	萱萱	默认	xuanxuan-plus	甜美女生
		端庄	xuanxuan-dignified-plus	
	晓琴	默认	xiaoqin-plus	亲切温和
	Jenny	默认	jenny-plus	纯正美音
	小峰	默认	xiaofeng-plus	男播音员
		庄重	xiaofeng-solemn-plus	
	小雯	默认	xiaowen-plus	女播音员
		庄重	xiaowen-solemn-plus	

	天天	默认	tiantian-plus	天真男孩
	糖糖	默认	tangtang-plus	活泼女孩
	珍妮	默认	zhenni-plus	异域风情
	玲玲	默认	lingling-plus	台湾女生
	晓珂	默认	xiaoke-plus	漂亮御姐
	宁宁	默认	ningning-plus	粤语女声
	冰冰	默认	bingbing-duihua-plus	明亮利落
		客服	bingbing-kefu-plus	
		消极	bingbing-neg-plus	
		积极	bingbing-pos-plus	
	贝儿	默认	beier-plus	元气少女
	惠惠	默认	huihui-normal-plus	成熟大方
		中立	huihui-neutral-plus	
		生气	huihui-angry-plus	
		害怕	huihui-fear-plus	
		开心	huihui-happy-plus	
		难过	huihui-sad-plus	
		惊喜	huihui-surprise-plus	
		讽刺	huihui-sarcastic-plus	
		可疑	huihui-suspicious-plus	
	晨阳	默认	chenyang-normal-plus	真实自然
		中立	chenyang-neutral-plus	
		生气	chenyang-angry-plus	
		害怕	chenyang-fear-plus	
		开心	chenyang-happy-plus	
		难过	chenyang-sad-plus	
		惊喜	chenyang-surprise-plus	

		讽刺	chenyang-sarcastic-plus	
		可疑	chenyang-suspicious-plus	
	小颖	默认	xiaoying-plus	亲和自然
	雅琳	默认	yalin-plus	成熟温和
	小亮	默认	xiaoliang-plus	清朗明快
	明宇	默认	mingyu-plus	稳重浑厚
	盈盈	默认	yingying-plus	四川女声
	Tiffany	默认	tiffany-plus	纯正美音
	Johnny	默认	johnny-plus	纯正美音
	飞行少年	默认	feixingshaonian-plus	Rapper
	嘉仪	默认	jiayi-plus	粤语女声
普通音库	Kiyo	默认	kiyo-base	可爱女生
	小雯	默认	xiaowen-base	女播音员
	小峰	默认	xiaofeng-base	男播音员
	萱萱	默认	xuanxuan-base	甜美女生
	天天	默认	tiantian-base	天真男孩
	糖糖	默认	tangtang-base	活泼女孩
	玲玲	默认	lingling-base	台湾女生

	-	默认	stella aifei annie ruilin aishuo yaqun sicheng xiaoyun xiaogang aida sitong aicheng aijing xiaobei ailun aitong aiyu ninger jielidou aibao aijia xiaoxian aiya aiwei ruoxi aimei qiaowei aiqi siyue guijie aiyue xiaomei aixia	-

请求示例：

```
{
  "draft_content": "今天天气怎么样",
  "body_config": { "body_motion": 1 }
}
```

返回示例：

```
{
  "code": 0,
  "msg": "ok",
  "data": {
    "ttsSynthesizeData": {
      "audio_url": "https://ds-model-tts.tos-cn-beijing.volces.com/temp/168619399638616425.wav"
    },
    "allfaceData": {
      "oss_url": "https://ds-vhost-action-dev.oss-cn-beijing.aliyuncs.com/mouth/168619399638616425.json",
      "time_log": {
        "all_time": 0.1282792091369629,
        "download_time": 0.0002853870391845703,
        "featextr_time": 0.07973647117614746,
        "inference_time": 0.010663270950317383,
        "upload_time": 0.037532806396484375
      },
      "audio_time": 1.5375
    },
    "motionFusionedData": {
      "oss_url": "https://ds-vhost-action-dev.oss-cn-beijing.aliyuncs.com/body/168619399638616425.json",
      "time_log": {
        "all_time": 0.36855459213256836,
        "postprocess_motion_time": 0.00008344650268554688,
        "query_audio_time": 0.12358212471008301,
        "synth_motion_time": 0.17075419425964355,
        "upload_time": 0.06978988647460938
      },
      "audio_time": 1.5375
    }
  }
}
```

NLP 文字驱动数字人（问答类型）

> 输入问题文本获得答案文本、语音文件，并获得表情、身体动作等信息，身体动作可以根据实际情形选择是否生成。

请求地址： `https://motionverseapi.deepscience.cn/v3.0/api/AnswerCollectMotion`

请求方式： `POST`

请求类型： `application/json`

请求参数：

名称	类型	必填	描述	示例值
text	string	是	问答请求文本	您好
audio_format	string	否	表示语音的类型；若 audio_format 为”wav”，则表示语音数据为 wav 格式。若为“mp3”则语音数据是 MP3 格式；若为”base64”，则表示语音数据为 base64 格式；	默认为”wav”
tts_config ▼	object	否	tts 请求参数设置	{}
tts_voice_name	string	否	声音角色，可供选择的有 声音列表	默认值 aixia
tts_speed	number	否	声音的速度	默认值 50，范围 0-100；face_type 为 2 的时候，建议此项数值设定为 40；
tts_volume	number	否	声音的音量	默认值 50，范围 0-100
tts_fm	number	否	声音的音调	默认值 50，范围 0-100
face_config ▼	object	否	表情请求参数设置	{}
face_type	number	否	表情类型选择，1 为语音驱动表情，此时 face_gender 和 face_tag 起作用。2 为视素驱动，此时 face_gender 和 face_tag 不起作用。3、MetaHuman 驱动。4、表情生成驱动	默认为 2
face_gender	string	否	可以设定表情性别，性别不同表情效果不同。性别 male 男性 女性 female	有默认值、默认 male
face_tag	number	否	面部情绪 情绪 0-综合 1-高兴 2-悲伤 3-生气 4-惊讶 5-中立 6-激动 7-沮丧 8-恐惧 9-厌恶	默认值为 0

名称	类型	必填	描述	示例值
face_stretch_scale	number	否	face_type 为 2、3 时，表示希望嘴部发‘a’的音的夸张的程度，值越大在发’a’时嘴越夸张；范围[1.0, 3.0]	1.5
face_mouth_divisor	number	否	face_type 为 2、3 时，表示整体的嘴部的幅度，值越大嘴部整体幅度越小，范围[0.5, 3]	1.6
face_smile_template	number	否	微笑模板，范围[0, 80] 只有 face_type 为 4 时生效	40
body_config ▼	object	否	动作请求参数设置	{}
body_motion	number	否	身体动作，0=不生成身体动作 1=自然闲聊 2=通用演讲 3=通用讲解 4=丰富演讲 5=金姐演讲 6=可爱女生 7=站立直播 8=标准客服 9=坐姿直播 10=主持人风格	默认为空
body_filter_window	number	否	表示动作自然度 数值越大自然效果越明显，但细节损失也会越多。范围 3-31，必须为奇数；此项必须与 order 搭配使用。非必须请默认此项。	默认值 15
body_filter_order	number	否	动作自然阶数，范围 2-4，数值越小阶数越高；此项必须与 window 搭配使用。非必须请默认此项。	默认值 3
body_fixed_hips	bool	否	表示获取动作数据是半身还是全身。如果是 True 就直接获取上半身数据，False 直接获取全身数据；	默认 false
body_head_x_rot	number	否	用于调整头部的仰俯角度。范围为-180 度～ 180 度；正数表示低头，负数表示抬头	默认值 0
body_compress	bool	否	表示返回的 oss_url 的内容是否是二进制；若为 True，生成的是二进制格式，二进制格式传输速度更快；若为 False，生成的是可读的 json 格式；	默认为 false

返回示例：

名称	类型	描述	示例值
code	number	响应码	0
msg	string	提示信息	正常返回"ok"，错误返回具体错误信息。
data▼	object		
answerDmeasyData▼	object	问答返回数据	
answer	string	答案文本	酒店有早餐
query	string	问题文本	有早餐吗
result_from	string	语义匹配：1.eva_chatbot：闲聊机器人；2.tencent_sim：语义匹配	tencent_sim
simquery	string	与问答库中的哪个字段语义相似	
ttsSynthesizeData▼	object	tts 语音合成数据	
audio_url	string	生成的音频地址	href="http://xxx.xxx.wav" http://xxx.xxx.wav
allfaceData▼	object	表情动作数据	
oss_url	string	bs 的 url	href="http://xxx.xxx.json" http://xxx.xxx.json
time_log	string	流程中的时间戳日志	
audio_time	number	播报时长	"audio_time": 2.8875
motionFusionedData▼	object	身体动作数据	
oss_url	string	bs 的 url	href="http://xxx.xxx.json" http://xxx.xxx.json
time_log	string	流程中的时间戳日志	
audio_time	number	播报时长	"audio_time": 2.8875

请求示例：

```
{
  "text": "今天天气怎么样",
  "body_config": { "body_motion": 1 }
}
```

返回示例：

```
{
  "code": 0,
  "msg": "ok",
  "data": {
    "answerDmeasyData": {
      "answer": "抱歉，我还没想好怎么回答您的问题",
      "code": 0,
      "query": "你好，今天天气怎么样",
      "result_from": "tencent_sim",
      "simquery": ""
    },
    "ttsSynthesizeData": {
      "audio_url": "https://ds-model-tts.tos-cn-beijing.volces.com/temp/168619376484532062.wav"
    },
    "allfaceData": {
      "oss_url": "https://ds-vhost-action-dev.oss-cn-beijing.aliyuncs.com/mouth/168619376484532062.json",
      "time_log": {
        "all_time": 0.19165873527526855,
        "download_time": 0.0008087158203125,
        "featextr_time": 0.11763906478881836,
        "inference_time": 0.01115274429321289,
        "upload_time": 0.06200051307678223
      },
      "audio_time": 2.8875
    },
    "motionFusionedData": {
      "oss_url": "https://ds-vhost-action-dev.oss-cn-beijing.aliyuncs.com/body/168619376484532062.json",
      "time_log": {
        "all_time": 0.7395775318145752,
        "postprocess_motion_time": 0.0001316070556640625,
        "query_audio_time": 0.05816936492919922,
        "synth_motion_time": 0.5564372539520264,
        "upload_time": 0.11744952201843262
      },
      "audio_time": 2.8875
    }
  }
}
```

说明

U3D、UE、小程序，通过使用我们平台提供的插件（插件封装了常见调用数字人驱动的方法，可以减少开发者工作量，开发者也可以选择不使用平台提供的插件，直接进行开发），插件包含U3D/UE插件、小程序插件。特点：

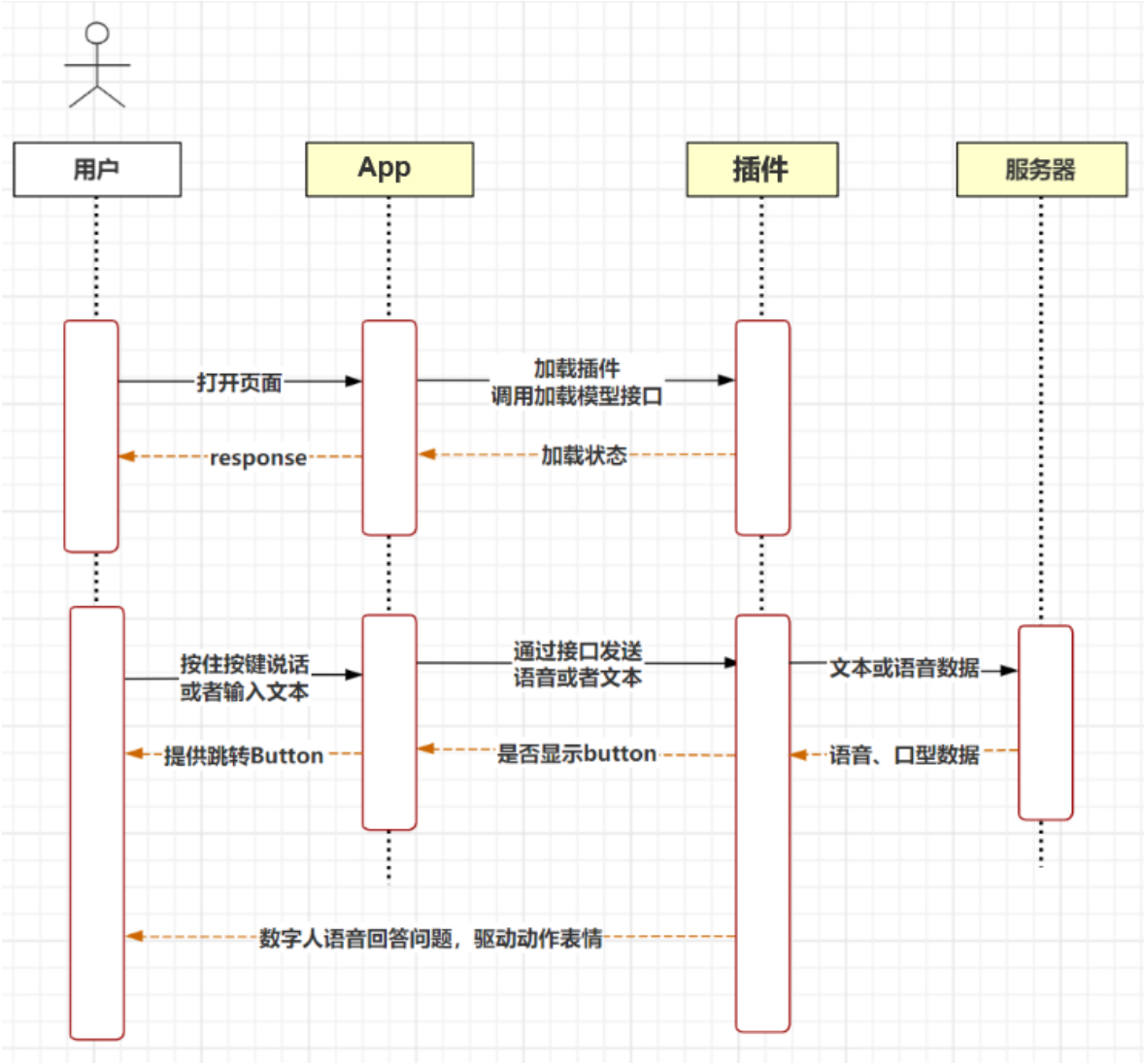
1、接入方便和灵活 开发者后台注册、创建应用，使用平台提供的插件，可以快速使用数字人驱动

2、沉浸式体验 原生开发，沉浸式体验更加美好

3、接口灵活、满足多样化场景 丰富的 3D 资产 ，多语种、多音色的情感化 TTS 、动作表情实时生成

4、多样化部署 支持saas、公有云、私有化部署，满足企业的各种需求

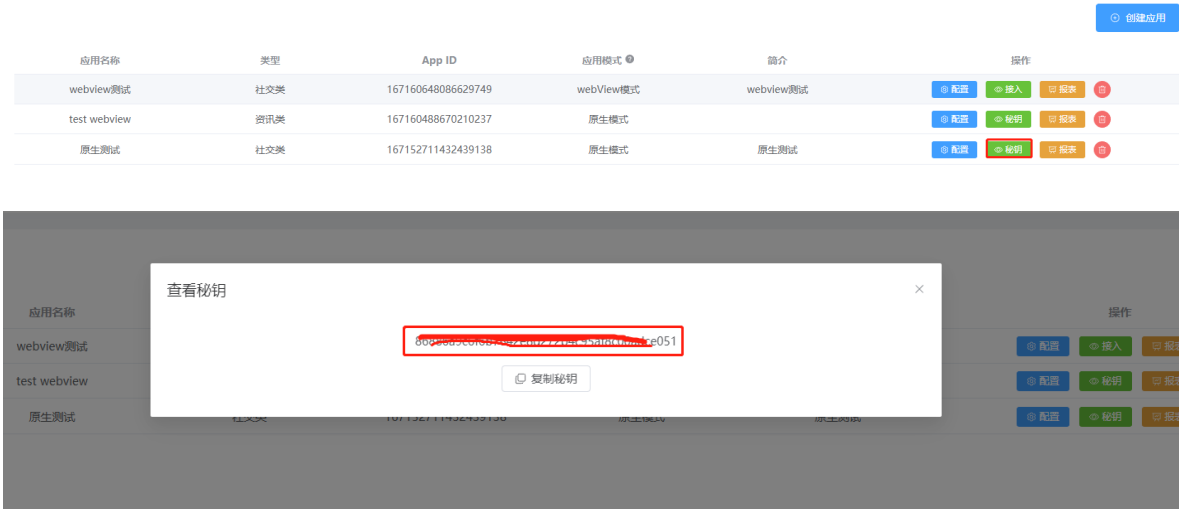
功能时序图



接入前准备

创建应用，选择原生模式

点击“秘钥”，获取到应用秘钥字符串，后续可以接入数字人驱动



后台配置

问答库配置：如果使用问答功能，需要提前在后台配置好问答的内容，支持新增、模板导入来创建问答内容。

概述

版本号：v1.1.5

用户点击对应按钮测试效果，数字人通过播报或者回答文字内容输出语音和驱动口型动作。

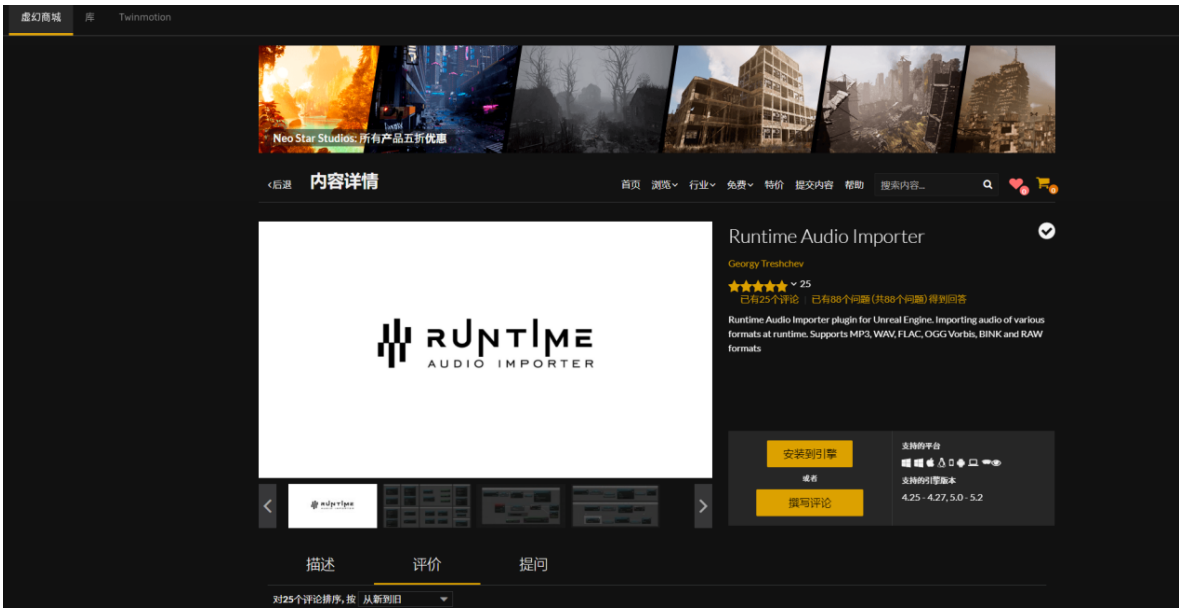
UE5.0：下载插件

UE5.1：下载插件

UE5.2：下载插件

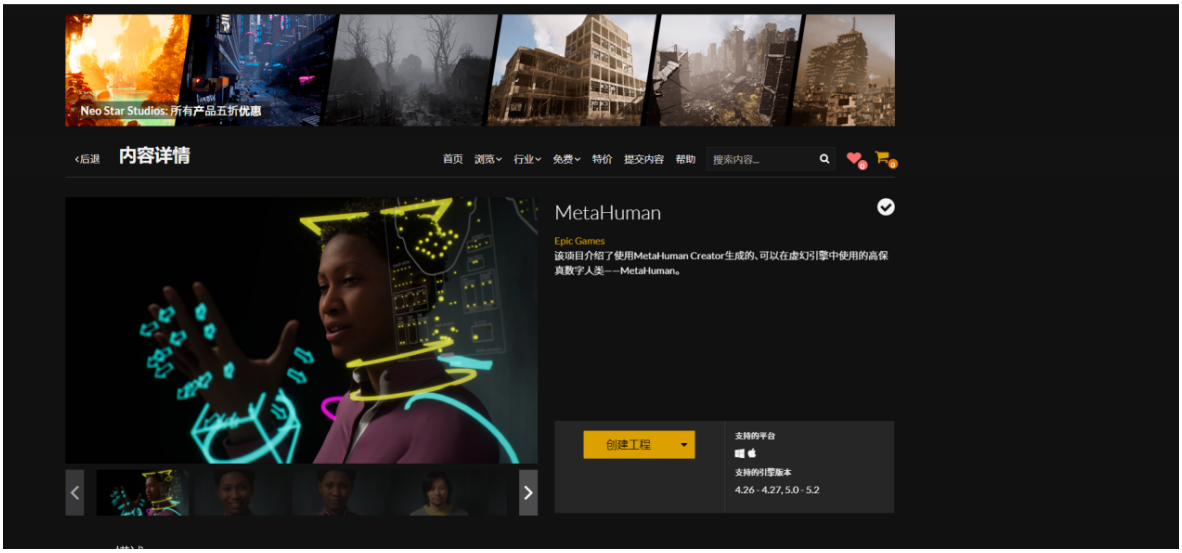
UE/MetaHuman骨骼

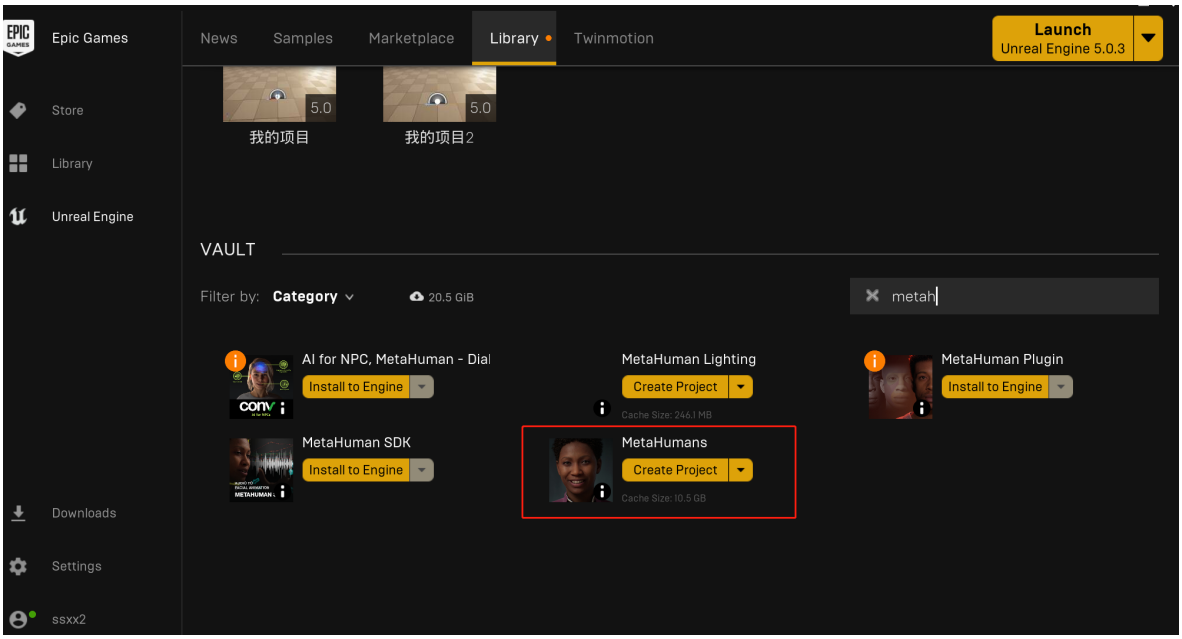
1、安装Runtime Audio Importer插件：



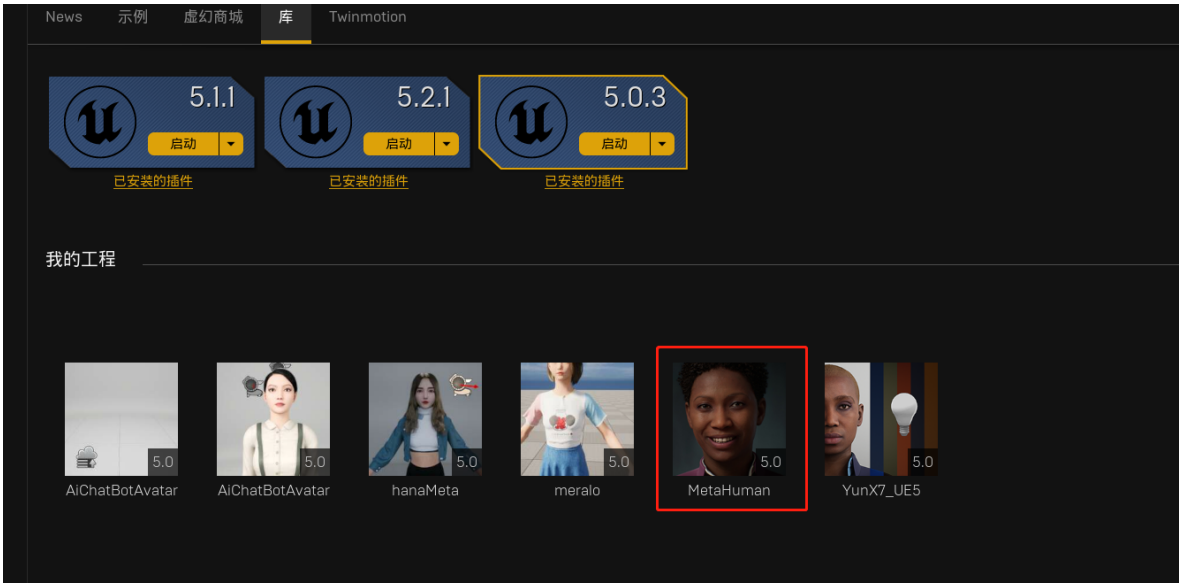
请注意这个插件一定要更新最新版本

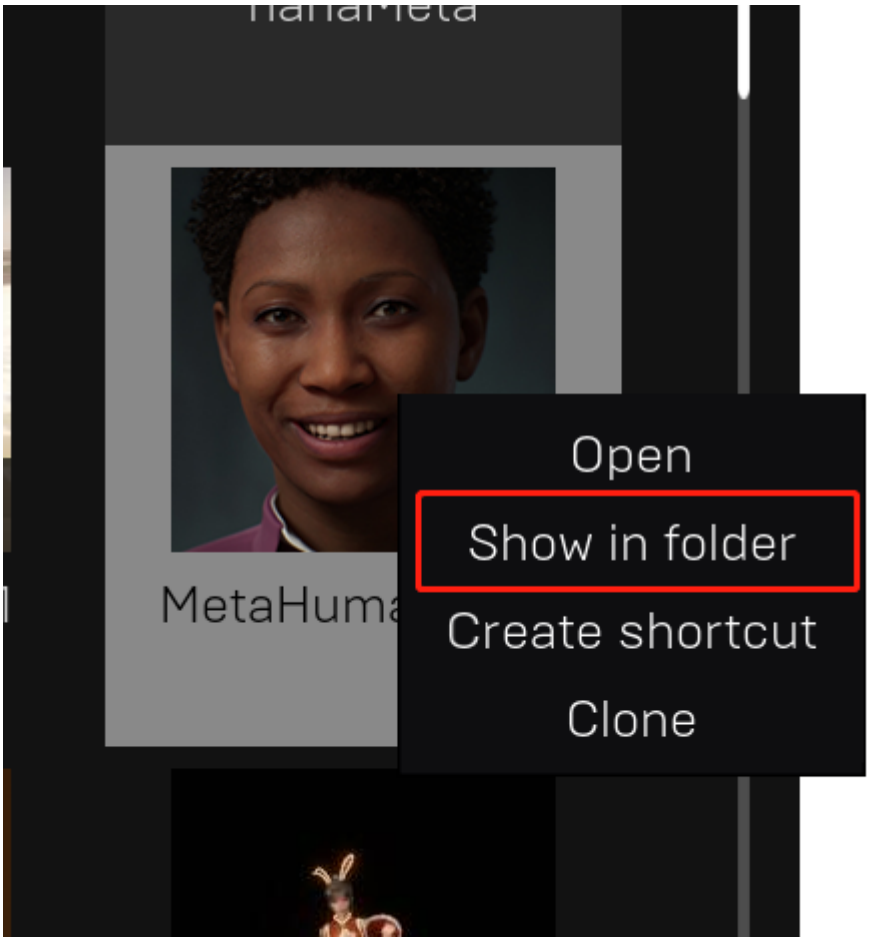
2、创建MetaHuman工程：

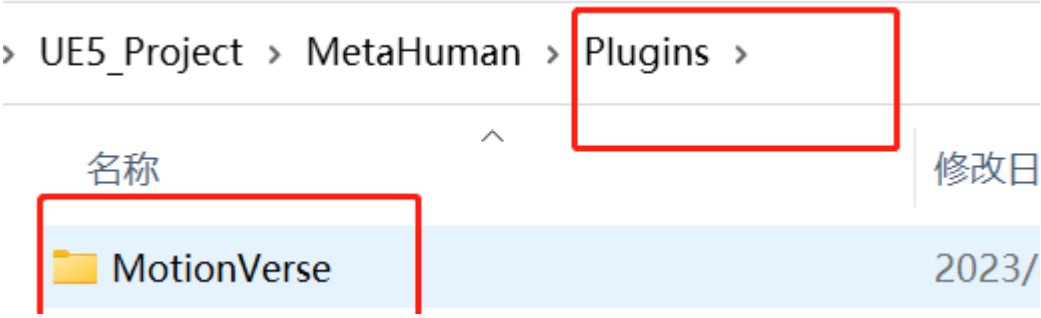




3、右键选择MetaHuman工程，点击在文件夹中显示，新建Plugins文件夹，把MotionVerse插件复制到Plugins文件夹下：

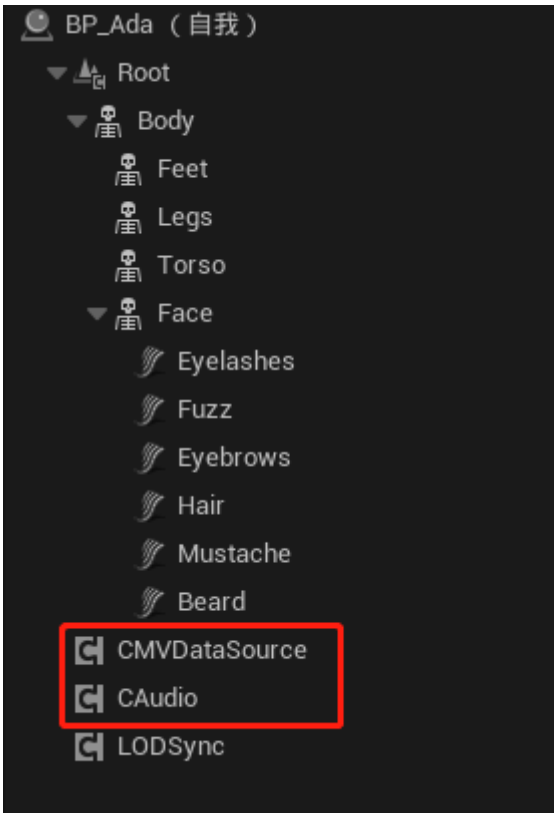

News
示例
虚幻商城
库
Twinmotion
5.1.1
启动
已安装的插件
5.2.1
启动
已安装的插件
5.0.3
启动
已安装的插件
我的工程
5.0
AiChatBotAvatar
5.0
AiChatBotAvatar
5.0
hanaMeta
5.0
meralo
5.0
MetaHuman
5.0
YunX7_UE5


Open
Show in folder
Create shortcut
Clone

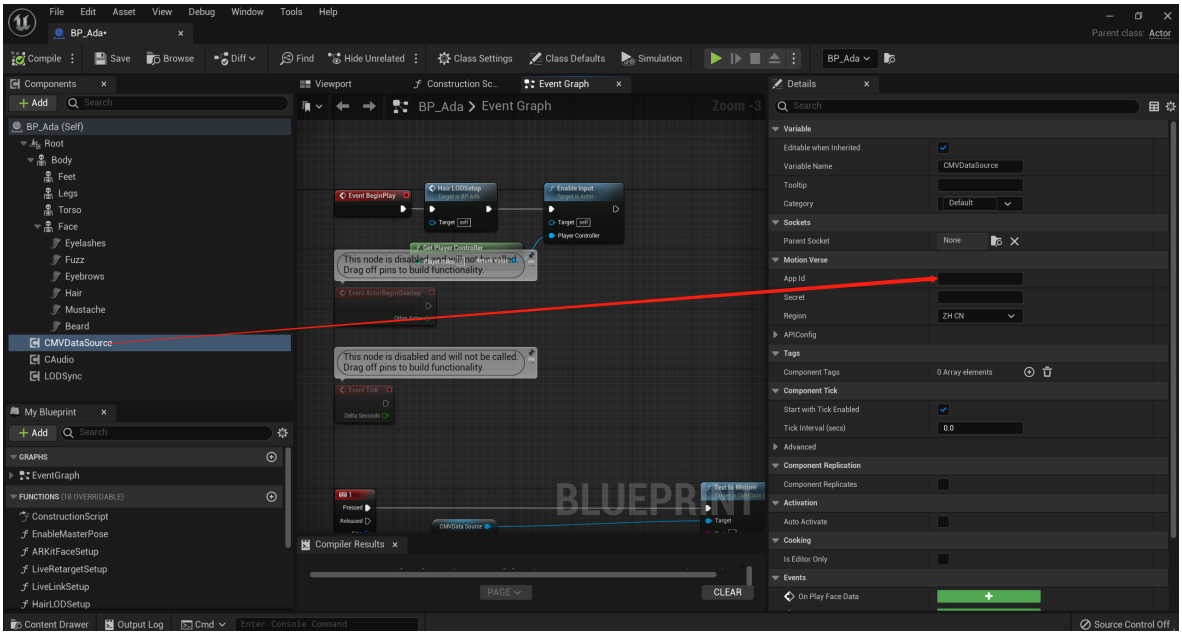

UE5_Project
MetaHuman
Plugins
名称
修改日
MotionVerse
2023/

插件可以从官网下载

4、打开MetaHuman角色蓝图，如下图所示添加CMVDataSource组件和CAudio组件



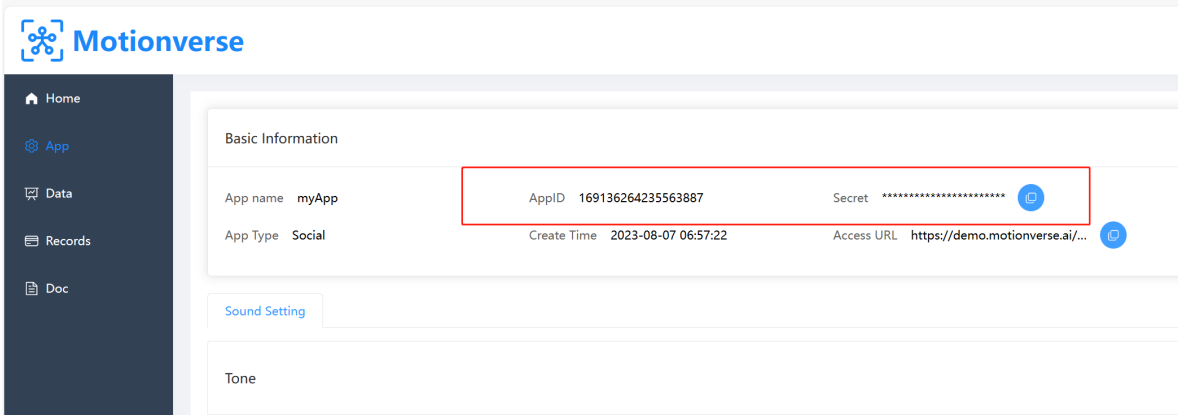
5、在CMVDataSource组件上写入motionverse对应的AppID和Secret。



我们的站点分为东亚和北美两个服务器

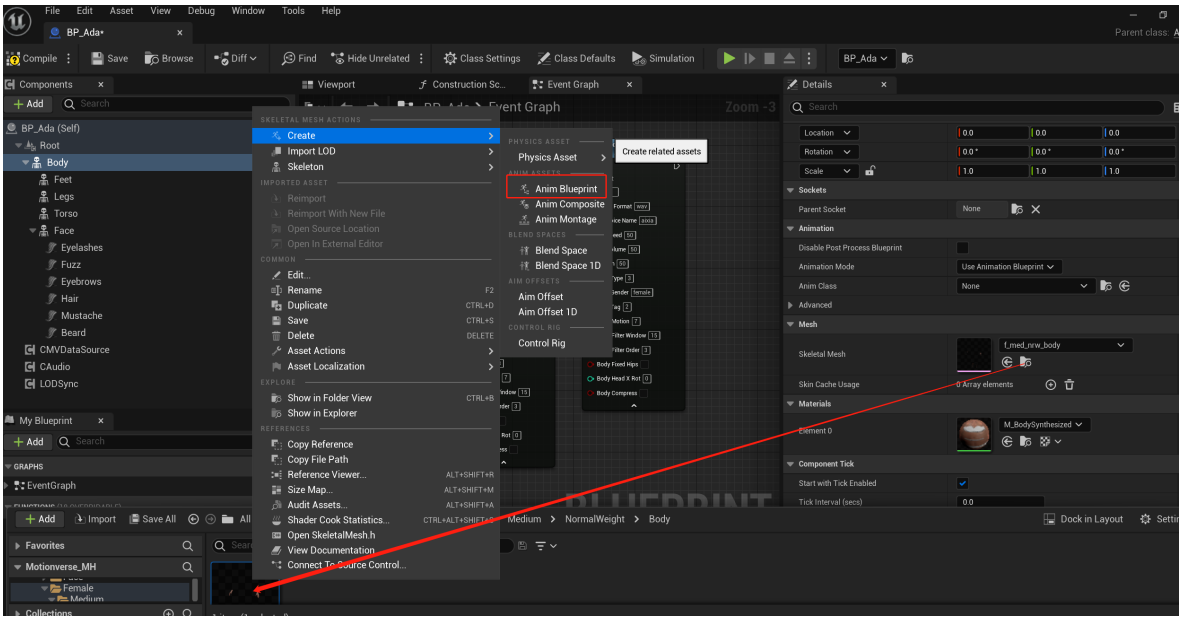
东亚需要在 motionverse.deepscience.cn 网站进行账号注册

北美需要在 motionverse.ai 网站进行账号注册

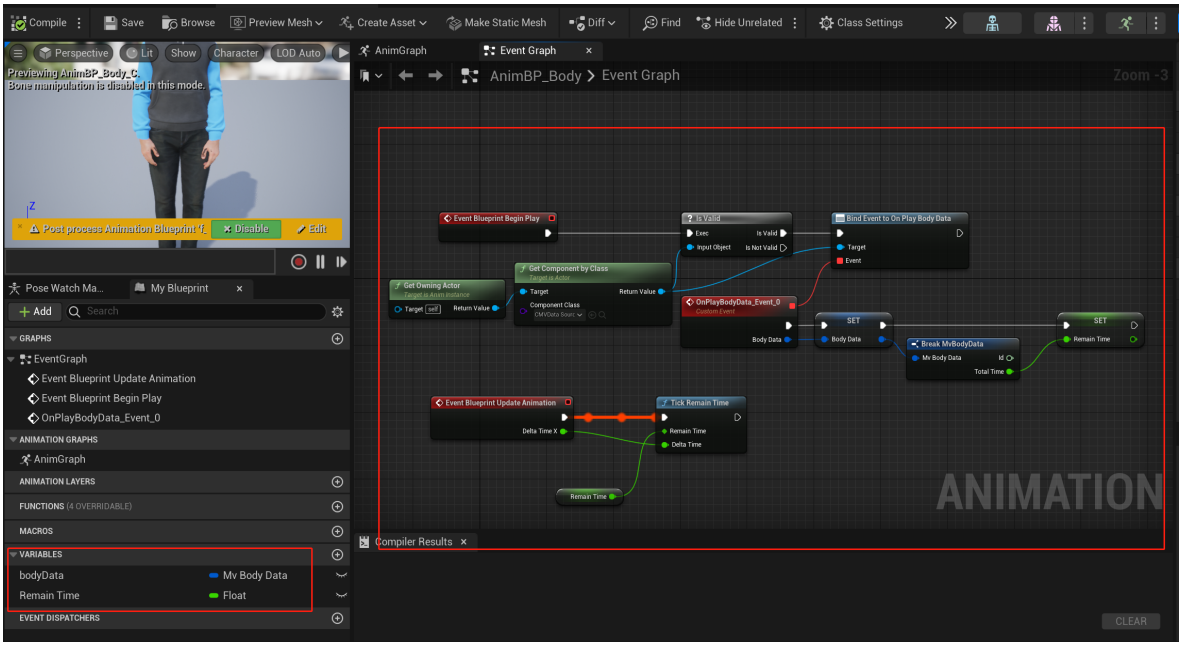


6、身体动画蓝图：

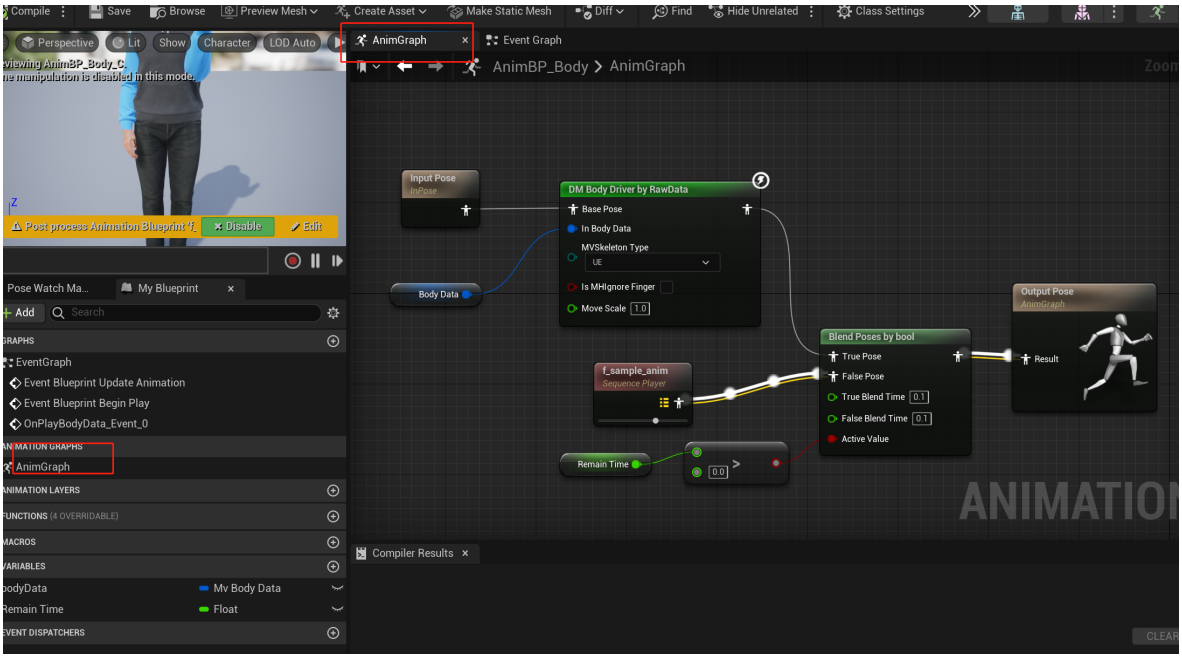
如果身体上没有动画蓝图，需要为身体新增动画蓝图



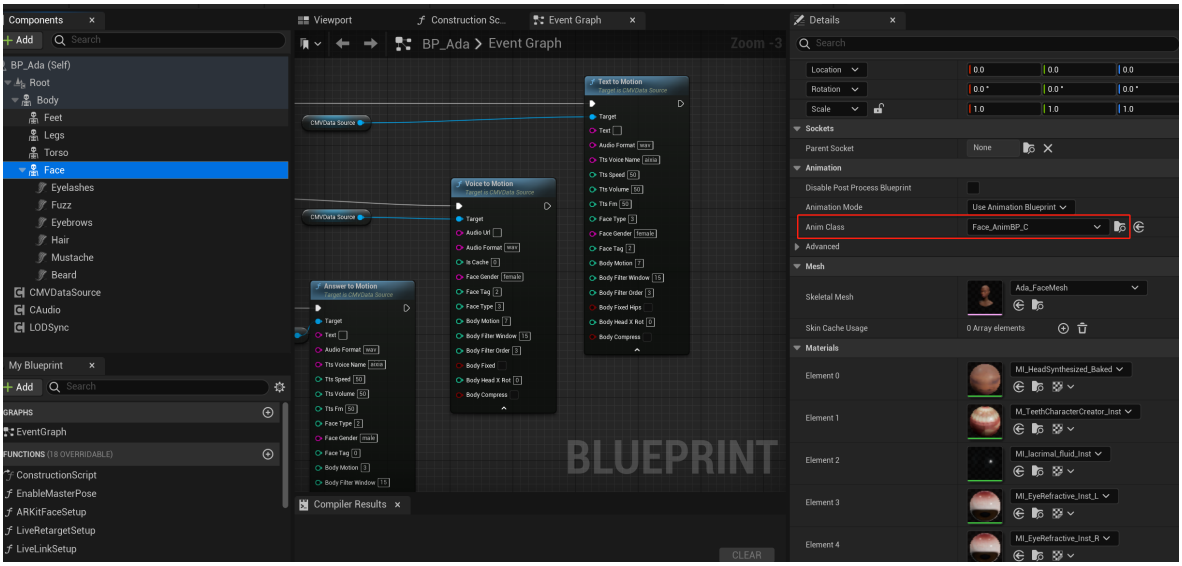
在动画蓝图上新增bodyData变量，类型为MvBodyData，新增Remain Time变量，类型为浮点型，添加下图所示蓝图节点：



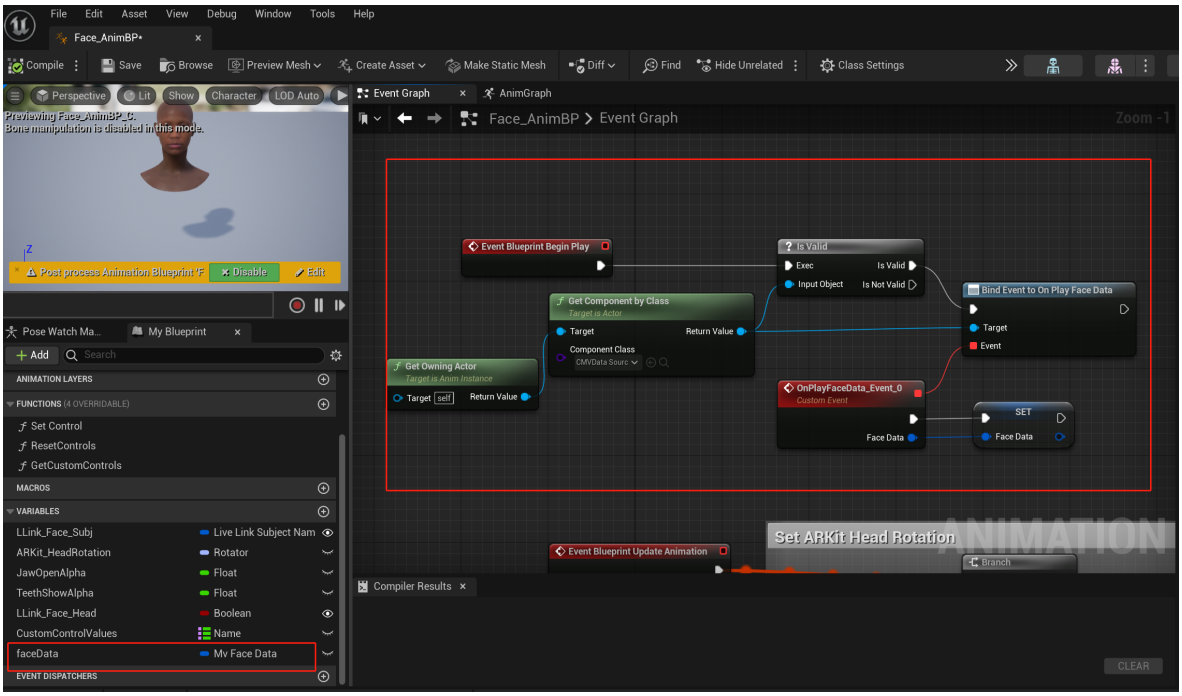
身体动画图：



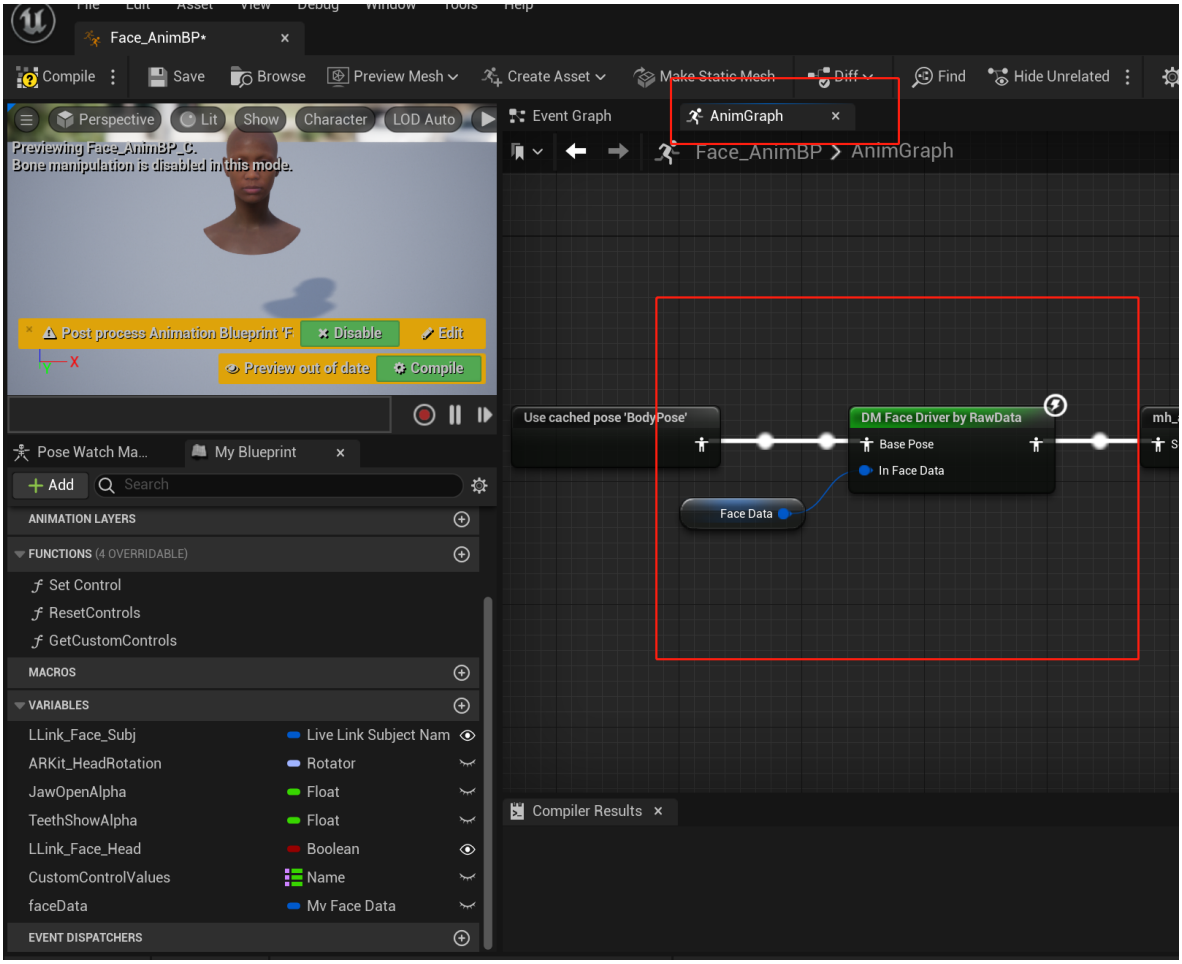
7、面部动画蓝图：



找到面部的动画蓝图，新增faceData变量，类型为MvFaceData，如下图所示添加蓝图节点：



面部动画图：



8、参数设定：

蓝图内有参数设定，包括文本内容，声音音色，动作风格等，具体参数，请参考API文档

Voice to Motion （语音驱动）

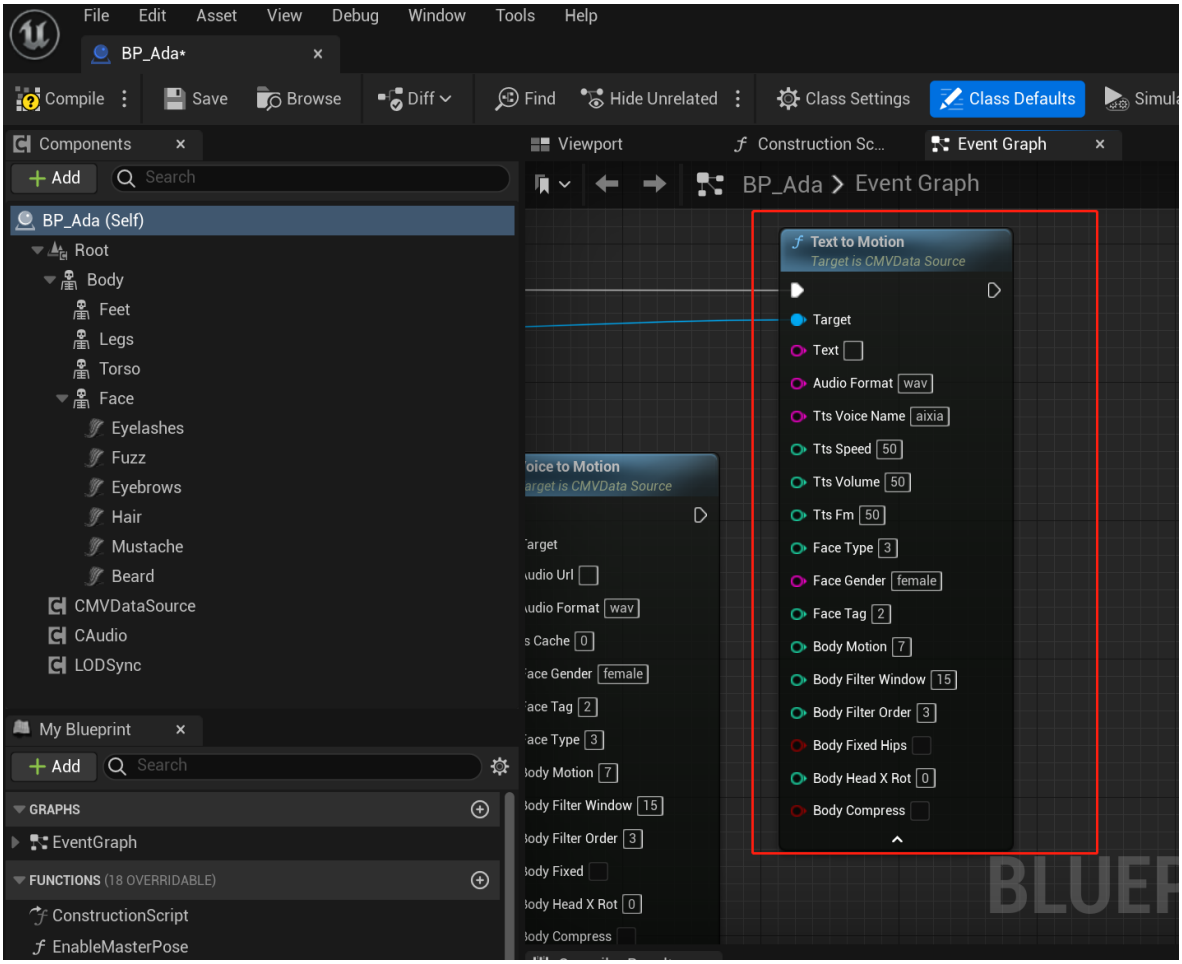
https://doc.deepscience.cn/AUDIO2MOTION.html

Text to Motion （文本驱动）

https://doc.deepscience.cn/BOARDCAST.html

Answer to Motion （问答驱动）

https://doc.deepscience.cn/NLPACTION.html



概述

版本号：v3.1.1

用户点击对应按钮测试效果，数字人通过播报或者回答文字内容输出语音和驱动口型动作。

插件下载地址：下载插件

> 支持unity 2020.3+版本

对接说明

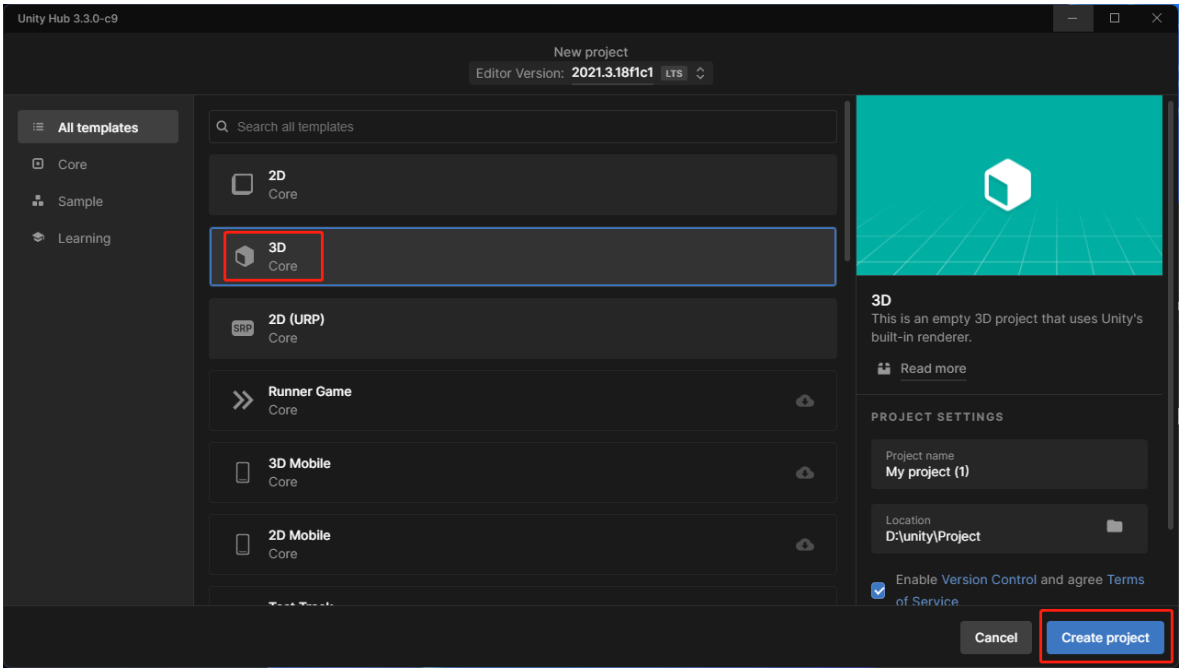
1，模型准备和绑定标准

（1）需要准备绑定完毕的模型FBX文件。

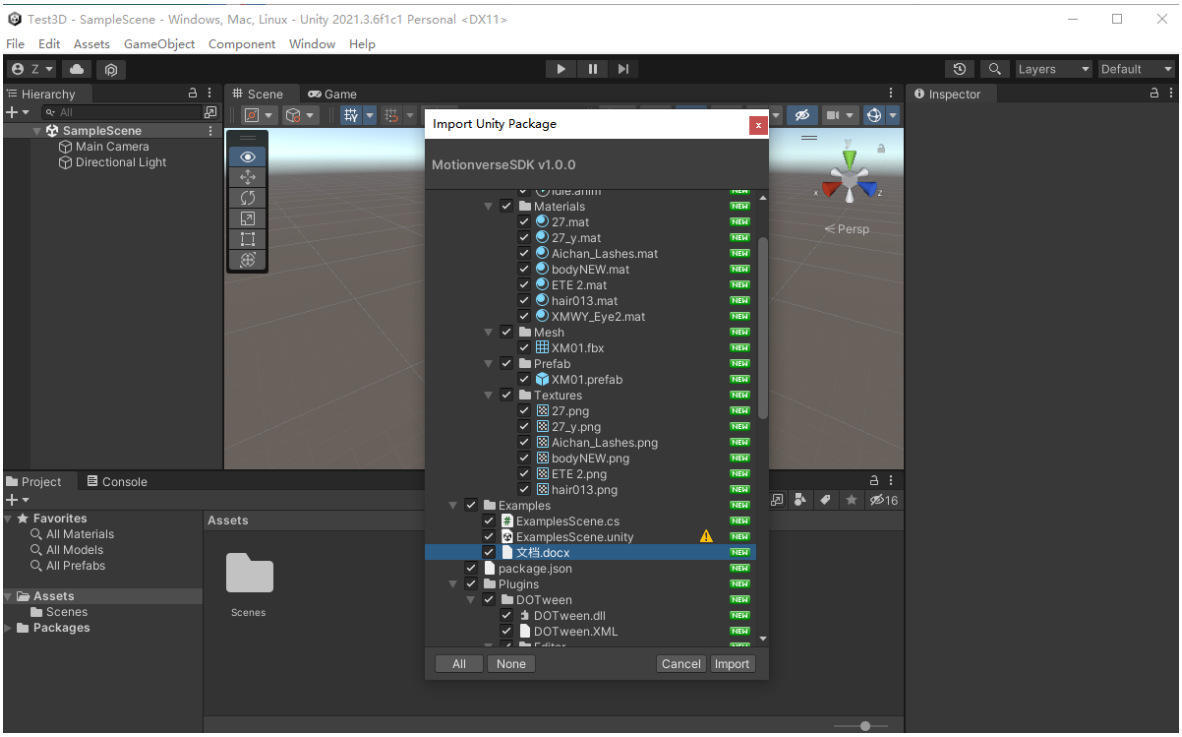
（2）对于模型的身体骨骼结构，我们插件提供了基础的数据重定向功能，可以重定向绝大部分的人形骨骼。

（3）对于脸部绑定，我们使用的是苹果ARKIT 52个blendshape的标准，需要需求驱动表情和口型的，就需要符合这个绑定规范。

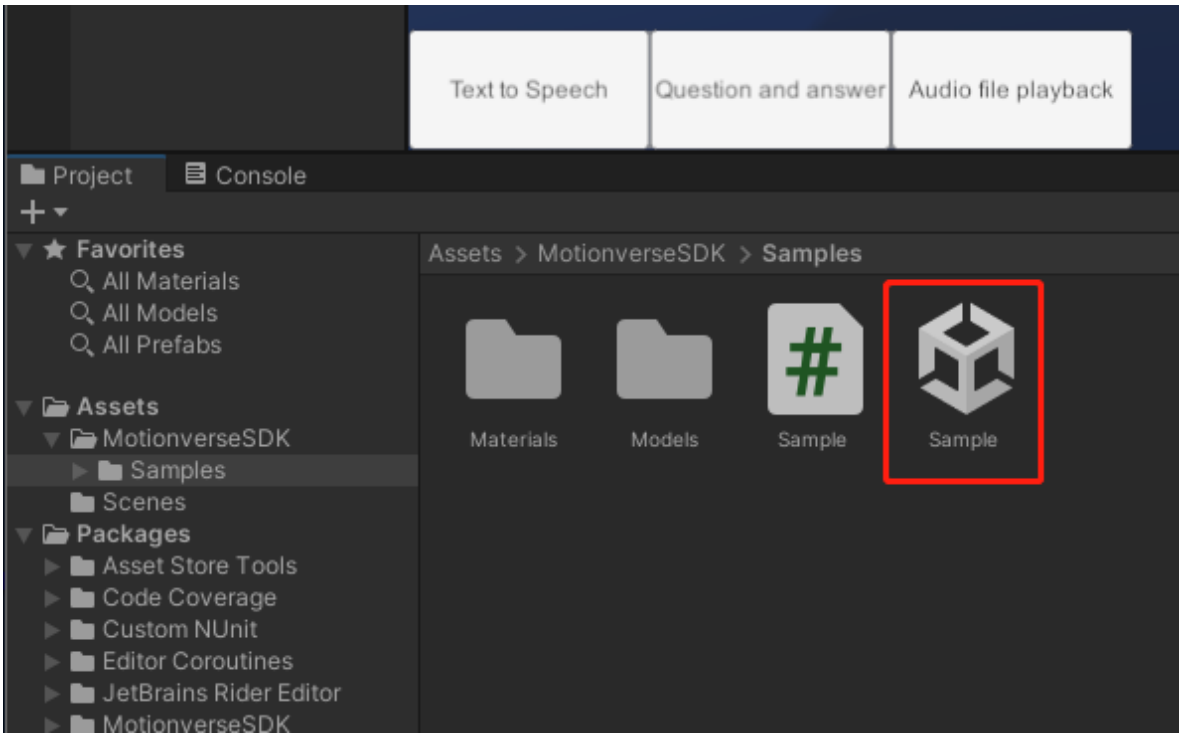
2，新建Unity项目，选择“3D核心模块”模板，创建项目，稍等片刻项目就创建成功。



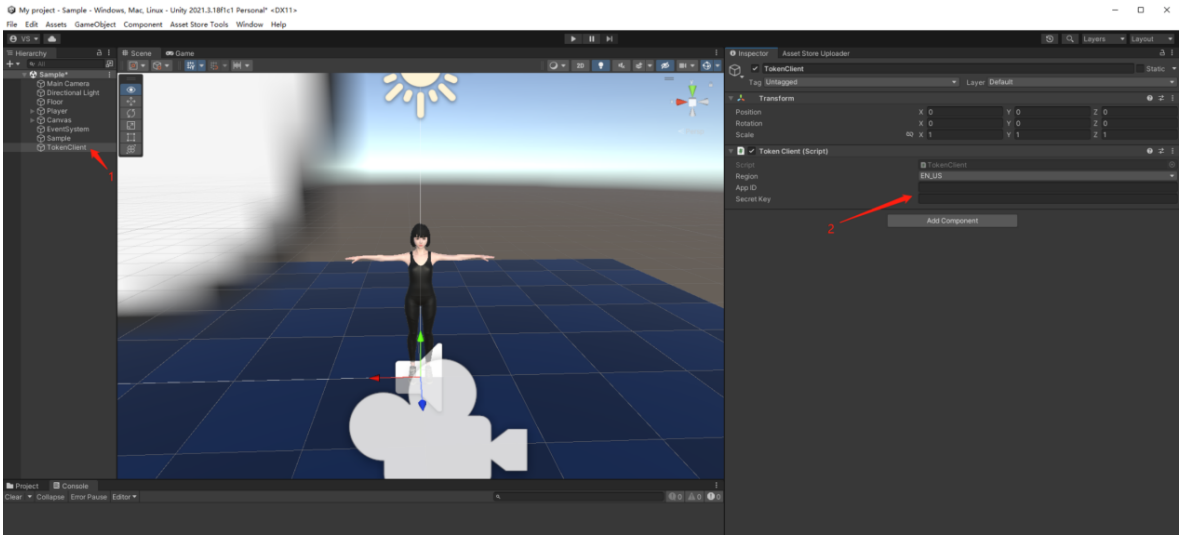
3，新建项目成功后，导入 MotionverseSDK，点击“import”即可



4、点击Assets/MotionverseSDK/Samples场景，可以查看使用Sample



5、选中“TokenClient”，然后在Inspector里面进行信息设置。Region根据你的地区来选择，这里要注意你的账号注册地址也要是一致的地区。然后输入创建好的App ID和Secret Key。



App ID 和 Secret Key可以在motionverse.deepscience.cn 注册后从创建应用获取。

6、Player组件进行参数配置，您可以调整声音动作类型等内容。可调参数如下：

VoiceName 为TTS音色名称：详情

Voicespeed 为合成声音速度，默认值50，范围0-100

VoiceVolume 为合成声音音量默认值50，范围0-100

Voice FM 为合成声音音调，默认值50，范围0-100

BodyMotion 为动作类型：详情

FaceType 表情驱动类型 1:语音驱动；2：音素驱动。音素驱动效果更好，口型咬合更准确。

Body Head X Rot 为角色在播放生成动作的时候，控制抬头或者低头，负数为抬头，比如 -10；

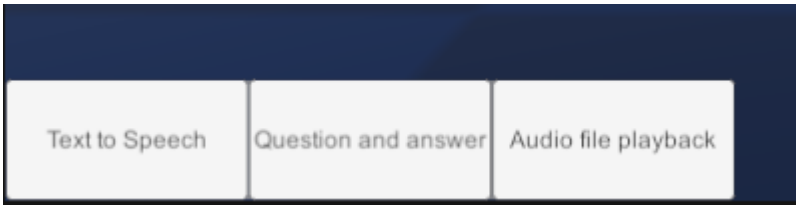
Left Hand Offest

Right Head Offset

Head Rotation Offset

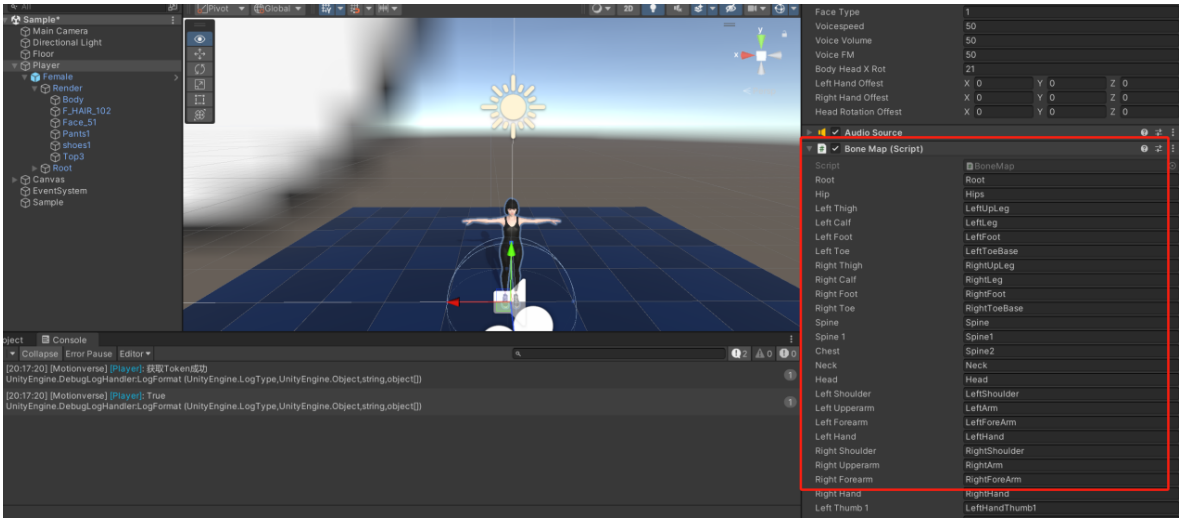
这三个参数，为重定向的时候，调整重定向的头手位置信息使用

7、点击对应按钮，就可以调试数字人驱动了。注意：如果此前Region选择的是EN_US,则无法使用Question and answer。



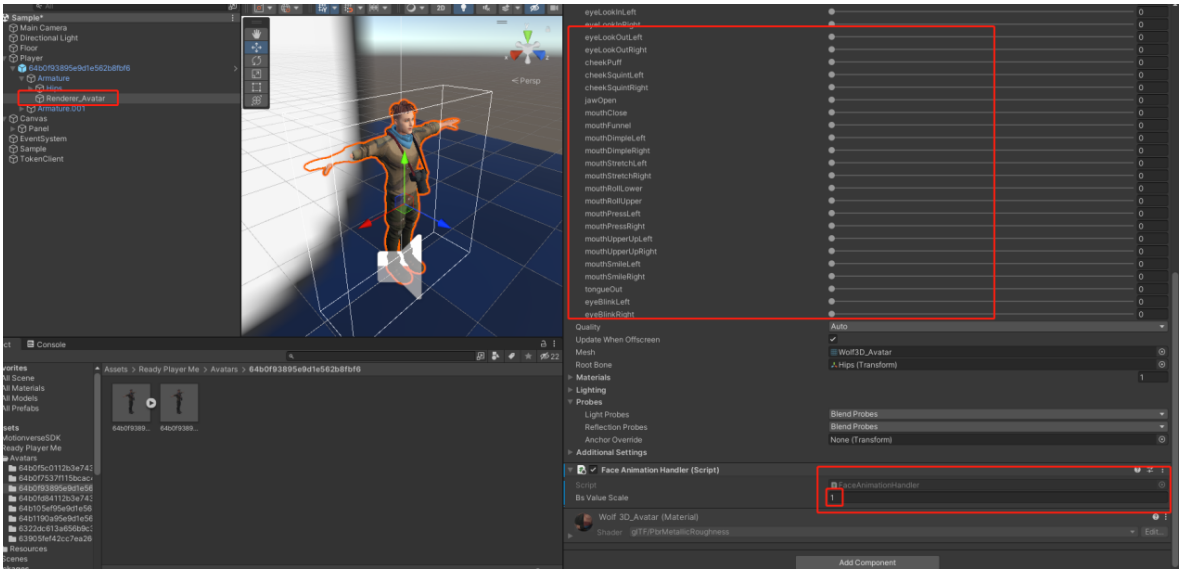
如果您需要使用自己的数字人，需要进行以下重定向步骤，流程如下：

1、配置重定向参数，对于有重定向需求的模型，需要配置模型的骨骼映射关系。



此处配置好后，会自动支持重定向功能，数据可以直接对数字人进行驱动

2、在有表情BS的部分，加入Face Animation Handler



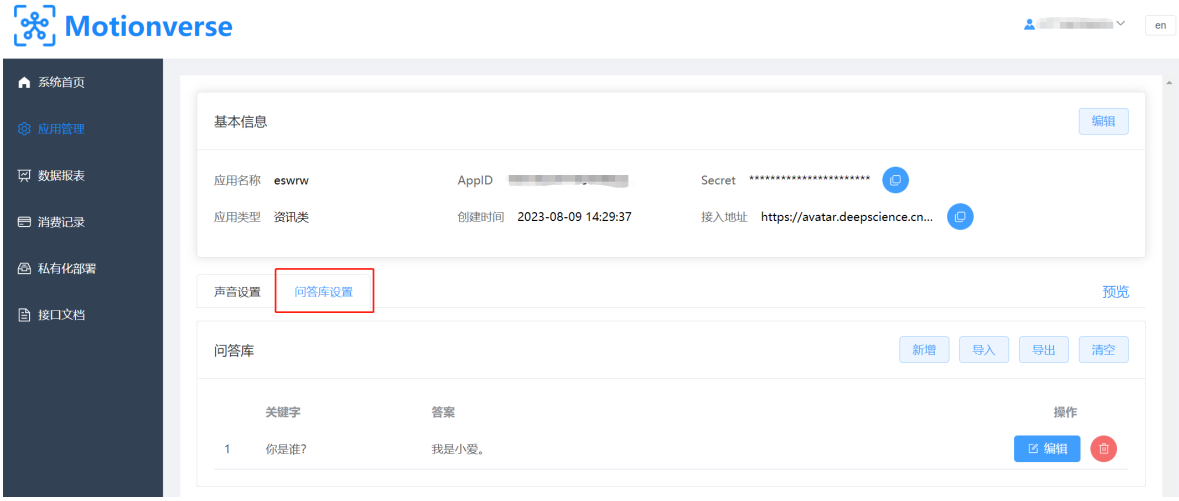
当您熟悉数字人插件后，就可以根据自己的需求去开发数字人应用了！

可以参考 Sample.cs

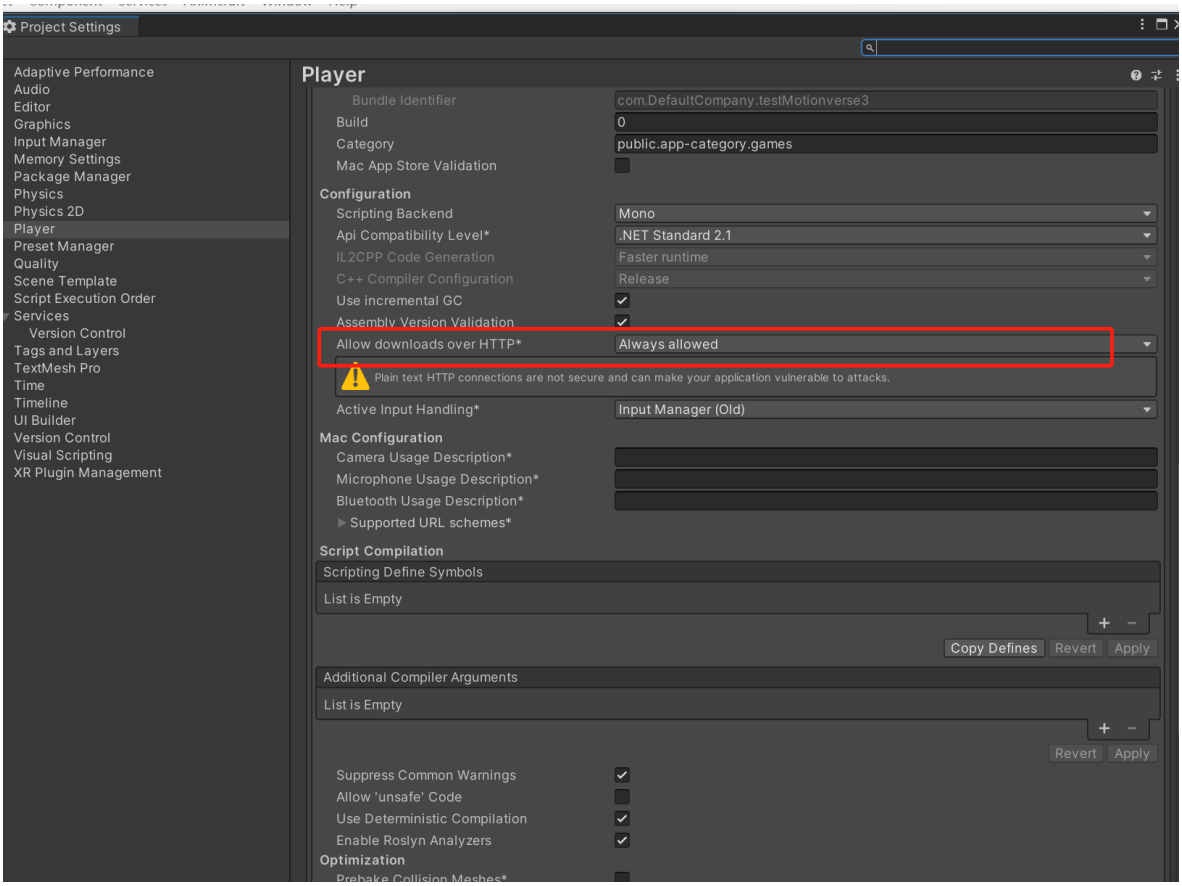
```
public void OnTextDrive(string text)
{
    DriveTask driveTask = new DriveTask() { player = player, text = text};

    TextDrive.GetDrive(driveTask,true);
}

0 个引用
public void OnAudioUrlDrive(string url)
{

    DriveTask driveTask = new DriveTask() { player = player, text = url };


    AudioUrlDrive.GetDrive(driveTask, true);
}


0 个引用
public void OnNLPDrive(string text)
{
    DriveTask driveTask = new DriveTask() { player = player, text = text };
    NLPDrive.GetDrive(driveTask, true);
}

0 个引用
```

```
TextDrive ：  输入文本，会返回合成的语音、动作和表情，驱动数字人

AudioUrlDrive:  输入音频的URL，返回动作和表情，驱动数字人

OnNLPDrive: 输入问题文本，会返回答案的语音、动作和表情，驱动数字人

多次调用会按调用顺序播放

GetDrive(driveTask, false) // 顺序调用播放

GetDrive(driveTask, true) // 提前播放

开始播放回调    player.OnPlayStart

播放完成回调    player.OnPlayComplete

播放错误回调    player.OnPlayError

播放暂停         player.StopPlay();
```

备注：问答部分，在motionverse.deepscience.cn 后台对应ID的产品中配置



注意：对于Unity2022 版本，需要在Player设置中允许Http下载



概述

版本号：v1.1.0

用户通过语音或文本提交问题，机器人通过回答的文字内容输出语音和驱动口型动作。

对接说明

1、插件申请接入：

在腾讯公众平台中，“微信小程序官方后台-设置-第三方服务-插件管理” 里点击 “添加插件”，搜索 “Motionverse交互数字人” 申请，审核通过后，小程序开发者可在小程序内使用该插件。

2、引入插件包：

Motionverse交互数字人appId： wx89668f1adf0dd7a8

```
// app.json
{
  "plugins": {
    "motion-plugin": {
      "version": "1.1.5",
      "provider": "wx89668f1adf0dd7a8"
    }
  }
}
```

3、使用插件的组件模式

组件调用示例：

在需要使用问答机器人组件的页面 json 文件中注册该组件

```
// page.json
{
  "usingComponents": {
    "robot-view": "plugin://motion-plugin/robot-component"
  }
}
```

在小程序页面内使用

```
<!-- page.wxml -->
<robot-view id="robot"/>
```

```
// page.js
Page({
  onReady() {
    this.robotView = this.selectComponent("#robot")
  },
})
```

组件暴露的方法

```
/**
 * 鉴权
 * @param {fun} successCallback 成功回调
 */
auth(successCallback)

/**
  * 终端初始化
  * @param {string} terminalNo 设备编号，可选值。该值优先级：参数传递 > 配置对象 > 服务端获取
  * @param {*} callback 结果回调
  */
init(terminalNo, callback)


/**
  * 终端切换
  * @param {string} terminalNo 设备编号
  * @param {*} callback 结果回调
  */
switch(terminalNo, callback)

/**
  * 根据语音获取对应的答案
  * @param {OnStopCallbackResult} audioRes 音频文件对象
  * @param {*} options 配置数据 { "isNeedPlay": false }
  * @param {*} resultCallback 服务器结果回调
  * @param {*} playStartCallback 语音播报开始回调
  * @param {*} playCompleteCallback 语音播报结束回调
  */
getAnswerAccordAudio(audioRes, options, resultCallback, playStartCallback, playCompleteCallback)

/**
  * 根据文本获取对应的答案
  * @param {string} text 文本
  * @param {*} options 配置数据 { "isNeedPlay": false }
  * @param {*} resultCallback 服务器结果回调
  * @param {*} playStartCallback 语音播报开始回调
  * @param {*} playCompleteCallback 语音播报结束回调
  */
getAnswerAccordText(text, options, resultCallback, playStartCallback, playCompleteCallback)

/**
  * 根据文本获取对应的语音播报
  * @param {string} text 文本
```

```
  * @param {*} options 配置数据 { "isNeedPlay": false }
  * @param {*} resultCallback 服务器结果回调
  * @param {*} playStartCallback 语音播报开始回调
  * @param {*} playCompleteCallback 语音播报结束回调
  */
getBoardcastAudio(text, options, resultCallback, playStartCallback, playCompleteCallback)

/**
  * 开始动作(肢体动作、脸部表情)
  * @param {string} actionName 动作名称
  * @param {Number} actionId 动作唯一标识，(使用场景：一个动作用于多种回答) 选填
  */
startAction(actionName)

/**
 * 开始播放语音、动作、表情
 * @param {motionObj} 需要播放的对象
 * @param {*} playStartCallback
 * @param {*} playCompleteCallback
 *
 * motionObj
 * {
 * "actionName": "" // 预制身体动作名称 【与bodyUrl二选一】
 * "bodyUrl": "", // 生成身体动作文件的url  【与actionName二选一】
 * "audioUrl": "", // 生成声音文件的url
 * "mouthUrl": "" // 生成表情动作文件的url
 * }
 */
startPlayAnswer(motionObj, playStartCallback, playCompleteCallback)

/**
  * 停止动作
  */
stopAction()

/**
  * 改变语音的音量
  * @param {Number} volume 声音的音量 0~1
  */
changeAudioVolume(volume)
```

4、使用插件的扩展方法

在 js 文件中导入

```
const plugin = requirePlugin('motion-plugin')
```

插件暴露的方法

```
/**
  * 插件对象数据的配置
  * @param {*} obj 数据对象
  * {
  *   localMotionArr: [
  *     {
  *       actionName: "Jiang", // 由模型提供方提供
  *       audioUrl: "https://zksz/123456.wav", // 对应行为的语音
  *       mouthUrl: "https://zksz/123456.json" // 对应行为的口型
  *     }
  *   ],
  *   randomMotionKeyArr: ["Jiang1", "Jiang2"], // 由模型提供方提供
  *   faceMorphKey: "Face_51_0,Face_51_1", // 由模型提供方提供，0:小程序，1:web
  *   modelUrl: "https://wwww.xxx.com/kefu.glb", // 模型url地址，可以固定配置，也可以从服务端获取
  *   terminalNo: "00211", // 终端设备编号
  *   asrName: "baidu" // 语音识别名称
  *   audioFileFormat: "pcm" // 语音文件格式 pcm m4a
  *   serverUserId: '15001191540', // 用户名称
  *   appid: '166554574663336380',  // app ID
  *   secret: '52ffd4ac353f75a646d7cdb82165f709cab5f61d', // app 秘钥
  *   sceneOption:{ // 场景配置
  *     useSceneLight: true, // 使用场景光
  *     // modelPosition:{x:0, y:0, z:-1.5}, // 设置模型位置
  *     // clearVertexColor: 'f_tal_unw_body_lod1_mesh2' // 清除节点顶点颜色
  *   }
  * }
  */
plugin.configData(obj)

/**
  * 预加载模型，提前进行模型的下载
  * @param {string} modelUrl 模型链接地址
  */
preloadModel(modelUrl)

/**
  * 模型下载任务，可根据此对象监听下载进度
  */
downloadModelTask
```

问题反馈

如果在使用Motionverse交互数字人插件过程中遇到问题，可以通过发送邮箱（邮箱：luozhiwen@deepscience.cn）反馈。

说明

app（安卓、ios、windows、web）应用，通过WebView控件将数字人嵌入到app里面，从而使用数字人驱动。特点：

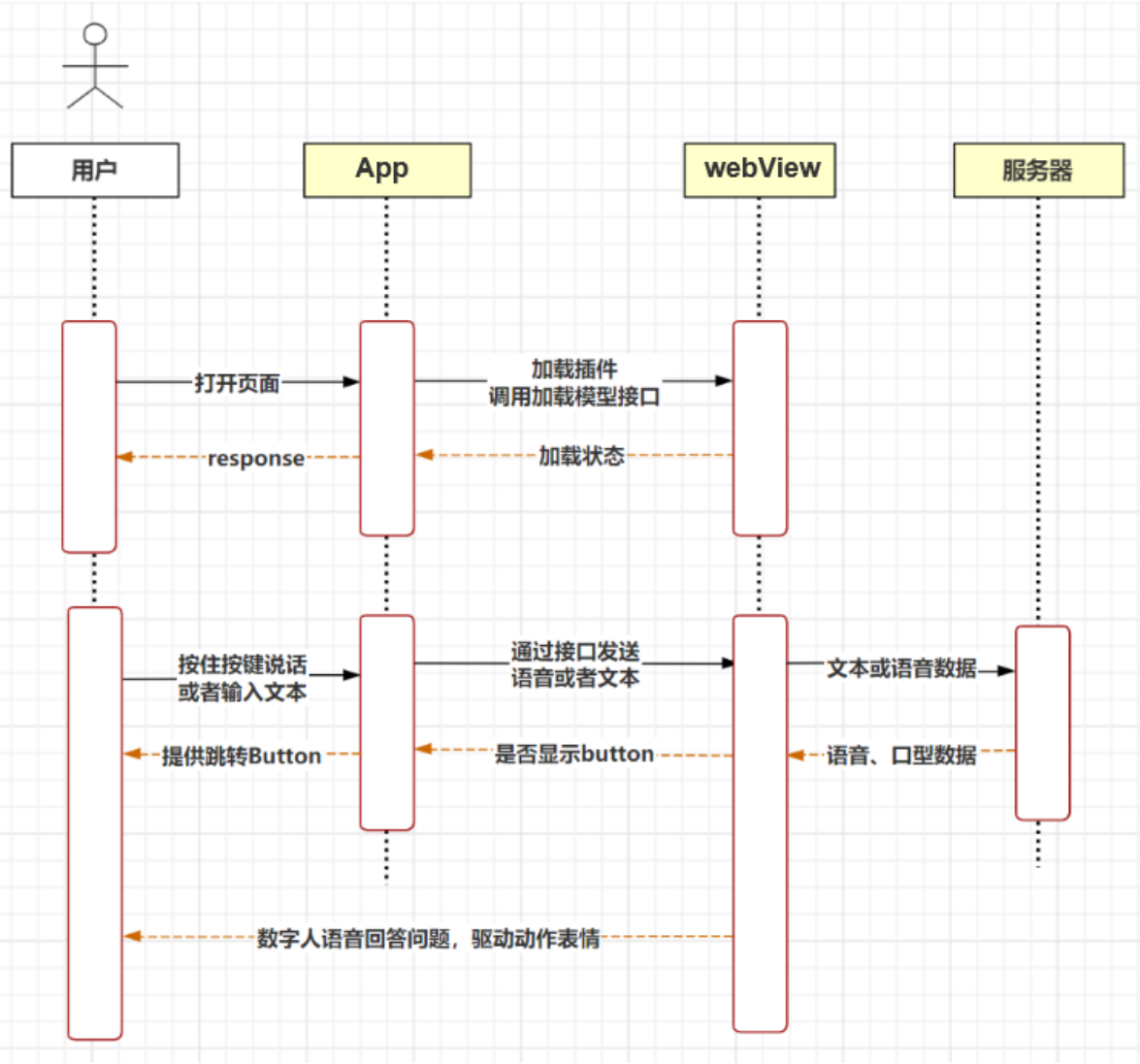
1、配置简单、一键接入 后台注册、创建应用，傻瓜式配置后即可在app中接入

2、跨平台兼容 支持在安卓、ios、h5等多平台兼容

3、限制少 没有包体限制，并且支持自己上传数字人或者使用平台数字人

4、多样化部署 支持saas、公有云、私有化部署，满足企业的各种需求

功能时序图



接入前准备

创建应用，选择webViiew模式

点击“链接”，获取webview的URL，后续可以接入到h5、安卓和ios

后台配置

问答库配置：如果使用问答功能，需要提前在后台配置好问答的内容，支持新增、模板导入来创建问答内容。

预览

点击"预览"，可以体验数字人应用，相关体验流程见“demo说明”

Motionverse 示例
点击播报语音文件

WEB接入 数字人

WEB以iframe方式加载数字人

```
<iframe src="链接地址从后台获取" frameborder="0" allow="autoplay" id="iframeDom"></iframe>
```

WEB调用iframe页面的方法

```
// 获取iframe dom
var iframeDom = document.getElementById('iframeDom')

// 更换数字人 type: 方法名 data:方法参数【数字人的名称】
iframeDom.contentWindow.postMessage({ type: 'ChangeCharacter', data: roleName }, '*')

// 播报声音文件地址 type: 方法名 data:方法参数【声音文件URL地址】
iframeDom.contentWindow.postMessage({ type: 'AudioBroadcast', data: audioUrl }, '*')

// 播报文本 type: 方法名 data:方法参数【播报的文本】 bodyMotion:身体动作 isFirst:播放是否放到第一位
// 注: 该方法可以连续多次调用, 输入不通的文本, 数字人会按调用顺序逐次播报
iframeDom.contentWindow.postMessage({ type: 'TextBroadcast', data: boardcastText, bodyMotion:3, isFirst:fal

// 问答 文字问答 type: 方法名 data:方法参数【问题的文本】 bodyMotion:身体动作
iframeDom.contentWindow.postMessage({ type: 'TextAnswerMotion', data: qaText, bodyMotion:3 }, '*')

// 问答 语音问答 type: 方法名 data:方法参数【录音流的BASE64编码】
iframeDom.contentWindow.postMessage({ type: 'AudioAnswerMotion', data: base64 }, '*')

// 中断数字人当前动作 type: 方法名 data: 参数无
iframeDom.contentWindow.postMessage({ type: 'BroadcastStop', data: null }, '*')

// 播放音量 type: 方法名 data: 参数 float【0~1】
iframeDom.contentWindow.postMessage({ type: 'AudioVolume', data: 1 }, '*')

// tts音色 type: 方法名 data: 参数 string【声音名称】
iframeDom.contentWindow.postMessage({ type: 'VoiceName', data: 'aixia' }, '*')

// 生成音频速度 type: 方法名 data: 参数 int【0~100】
iframeDom.contentWindow.postMessage({ type: 'VoiceSpeed', data: 50 }, '*')

// 生成音频音调 type: 方法名 data: 参数 int【0~100】
iframeDom.contentWindow.postMessage({ type: 'VoiceFM', data: 50 }, '*')

// 锁定数字人下半身动作
iframeDom.contentWindow.postMessage({ type: 'BodyFixedHips', data: true }, '*')
```

WEB监听iframe页面的方法

```
// 监听iframe方法
 window.addEventListener('message', function(){

  // iframe页面载入进度
  if (e.data.type == 'loading') {
    console.log(Math.ceil(e.data.data * 100))
  }

  // 交互完成loading
  if (e.data.type == 'loadAb') {
    interactiveLoading.value = e.data.data
  }

  // 播放开始
  if (e.data.type == 'playStart'){
    // 问题文本
    console.log(e.data.text)

    // 答案文本
    console.log(e.data.answer)
  }

  // 播放结束
  if (e.data.type == 'playEnd'){
    // code...
  }

  // 检测webview受到点击，【通常用来解决浏览器禁止自动播放音频的安全策略】
  if (e.data.type == 'touch'){
    // code...
  }

 }, false)
```

Web 访问链接： https://demo.deepscience.cn/html-poc/

Demo代码下载： https://github.com/Motionverse/html-poc

WebView 接入文档

WebView的初始化

```
let configuration = WKWebViewConfiguration()
configuration.allowsInlineMediaPlayback = true
configuration.mediaTypesRequiringUserActionForPlayback = .video
configuration.userContentController.add(self, name: "jsBridge")
let webView = WKWebView(frame: CGRect(x: 0, y: CGRectGetMaxY(navView.frame), width: UIScreen.main.bounds.wi
view.addSubview(webView)
```

WebView加载指定网页

```
let url = URL(string: "https://demo.deepscience.cn/poc/index.html")
let request = URLRequest(url: url!, cachePolicy: .returnCacheDataElseLoad, timeoutInterval: 60)
webView.load(request)
```

WebView中执行js函数

```
// 更换数字人 type: 方法名 data:方法参数【数字人的名称】
webView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"ChangeCharacter\",\"data\":\"\(robot.abName)\"}')")

// 播报声音文件地址 type: 方法名 data:方法参数【声音文件URL地址】
webView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"AudioBroadcast\",\"data\":\"https://ds-model-tts.os

// 播报文本 type: 方法名 data:方法参数【播报的文本】 bodyMotion:身体动作
webView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"TextBroadcast\",\"data\":\"\(text)\",\"bodyMotion\"

// 问答 文字问答 type: 方法名 data:方法参数【问题的文本】 bodyMotion:身体动作
webView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"TextAnswerMotion\",\"data\":\"\(text)\",\"bodyMotio

// 问答 语音问答 type: 方法名 data:方法参数【录音流的BASE64编码】
webView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"AudioAnswerMotion\",\"data\":\"\(detail)\"}')")

// 中断数字人当前动作
webView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"BroadcastStop\",\"data\":\"null\"}')")
```

WebView网页加载完成的回调

```
webView.navigationDelegate = self; // WKNavigationDelegate
func webView(_ webView: WKWebView, didFinish navigation: WKNavigation!) {
    print("网页加载完成")
}
```

APP收到WebView发过来的js消息

```
func userContentController(_ userContentController: WKUserContentController, didReceive message: WKScriptMe
    if message.name == "jsBridge" {
        print(message.body)
    }
}
```

Demo代码下载： https://github.com/Motionverse/MotionVerseWeb

WebView 接入文档

[toc]

---

1. 主要功能

调用js方法

```
// 更换数字人 type: 方法名 data:方法参数【数字人的名称】
mWebView.evaluateJavascript("SendMsgToWebGL('{\"type\":\"ChangeCharacter\",\"data\":\"${robot.abName}\"}')"

// 播报声音文件地址 type: 方法名 data:方法参数【声音文件URL地址】
mWebView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"AudioBroadcast\",\"data\":\"$audioUrl\"}')") {}

// 播报文本 type: 方法名 data:方法参数【播报的文本】bodyMotion:身体动作
mWebView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"TextBroadcast\",\"data\":\"$text\",\"bodyMotion\":

// 问答 文字问答 type: 方法名 data:方法参数【问题的文本】bodyMotion:身体动作
mWebView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"TextAnswerMotion\",\"data\":\"$text\",\"bodyMotion

// 问答 语音问答 type: 方法名 data:方法参数【录音流的BASE64编码】
mWebView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"AudioAnswerMotion\",\"data\":\"$audioEncode\"}')")

// 中断数字人当前动作
mWebView.evaluateJavaScript("SendMsgToWebGL('{\"type\":\"BroadcastStop\",\"data\":\"null\"}')") {}
```

展示设置页面

```
/**
 * 展示设置页面
 */
private fun showSetting() {
    BottomDialog(this@MainActivity)
        .setResId(R.layout.dialog_bottom) { view, dialog ->
              ...
        }
        .setOutSideCancelable(true)
        .show()
}
```

数字人更换

```
val listFragment = CharacterListFragment {
    Log.e(TAG, "character: $it")
    // 拿到数字人的地址后，调用js方法
    callJs(ChangeCharacter, it)
}
listFragment.show(supportFragmentManager, "characterFragment")
```

交互类型更换

```
/**
 * 展示交互类型更换
 */
private fun showChangType() {
    BottomDialog(this@MainActivity)
        .setResId(R.layout.dialog_type_change) { view, dialog ->
             ...
        }
        .setOutSideCancelable(true)
        .show()
}
```

关于示例

```
/**
 * 展示 关于示例
 */
private fun showExample() {
    BottomDialog(this@MainActivity)
        .setResId(R.layout.dialog_example)
        .setOutSideCancelable(true)
        .show()
}
```

2. 网络框架的初始化

主要作用是缓存网页中包含的图片、文件、字体等资源，在离线时也可正常使用。

```
setBaseUrl("http://www.xxx.com/")
setHttpClient(OkHelper.httpClient(applicationContext))
```

3. WebView的使用

WebView的预加载

主要作用是提前WebView的初始化，在使用时可以更快的获取WebView的实例。

```
WebViewManager.prepare(applicationContext)
```

WebView设置加载监听

```
mWebViewHelper.setOnPageChangedListener(object : WebViewHelper.OnPageChangedListener {
    override fun onPageStarted(view: WebView?, url: String?, favicon: Bitmap?) {
        Log.i(TAG, "onPageStarted $url")
    }

    override fun onPageFinished(view: WebView?, url: String?) {
        Log.i(TAG, "onPageFinished $url")
    }

    override fun onProgressChanged(view: WebView?, newProgress: Int) {
        Log.i(TAG, "onProgressChanged $newProgress")
    }
})
```

WebView加载指定网页

```
mWebViewHelper.loadUrl("http://www.xxx.com/")
```

WebView中执行js函数

```
mWebView.evaluateJavascript("js function")
```

WebView的回收

```
WebViewManager.recycle(mWebView)
```

Demo代码下载：
https://github.com/Motionverse/MotionVerseWeb_Android

DM身体绑定规范

1、骨骼命名与层级

按标准，所有命名及层级都不能错。

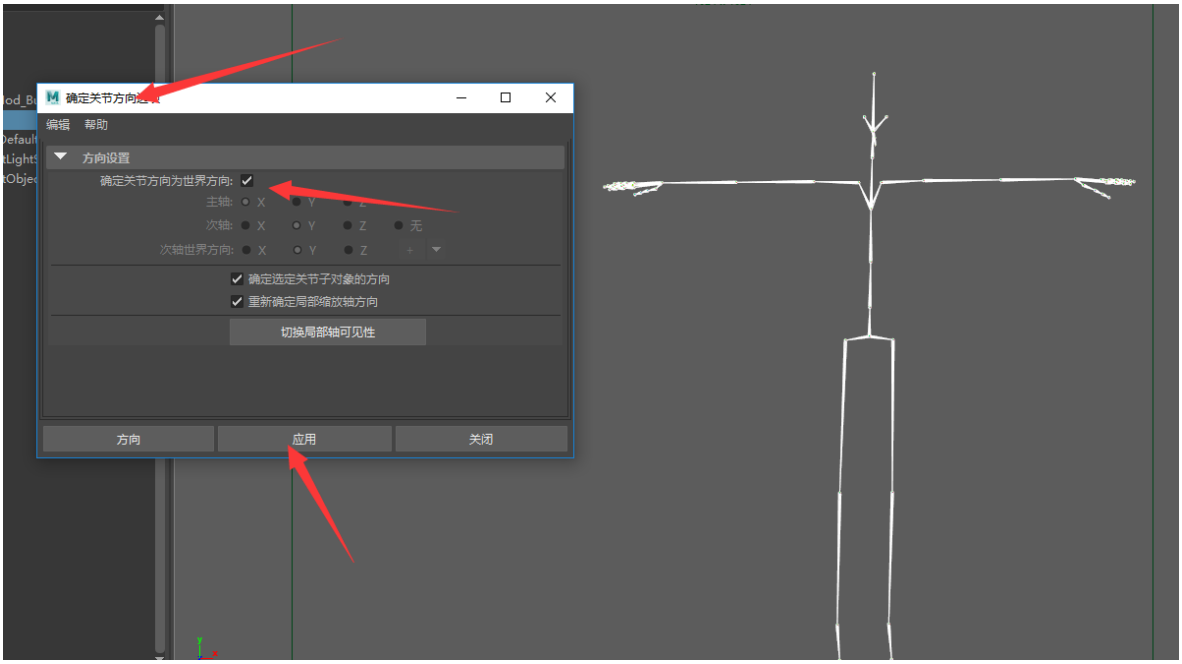
标准骨骼**St_Bone_G**为标准，可直接用作匹配模型。以下为下载链接。

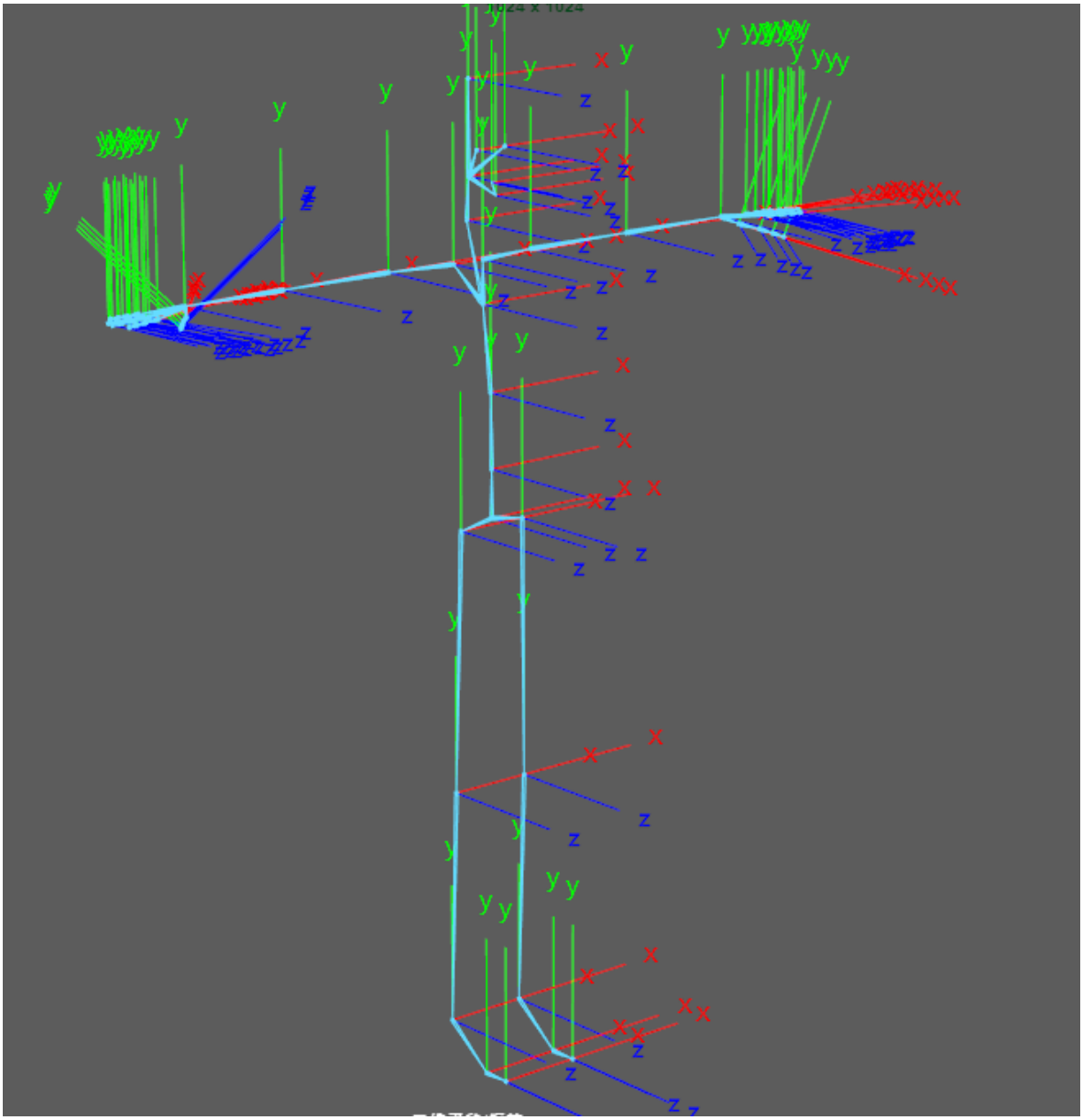
下载标准骨骼

可任意添加附属骨骼，但不能改变原骨骼层级的父子关系。

2、骨骼处理

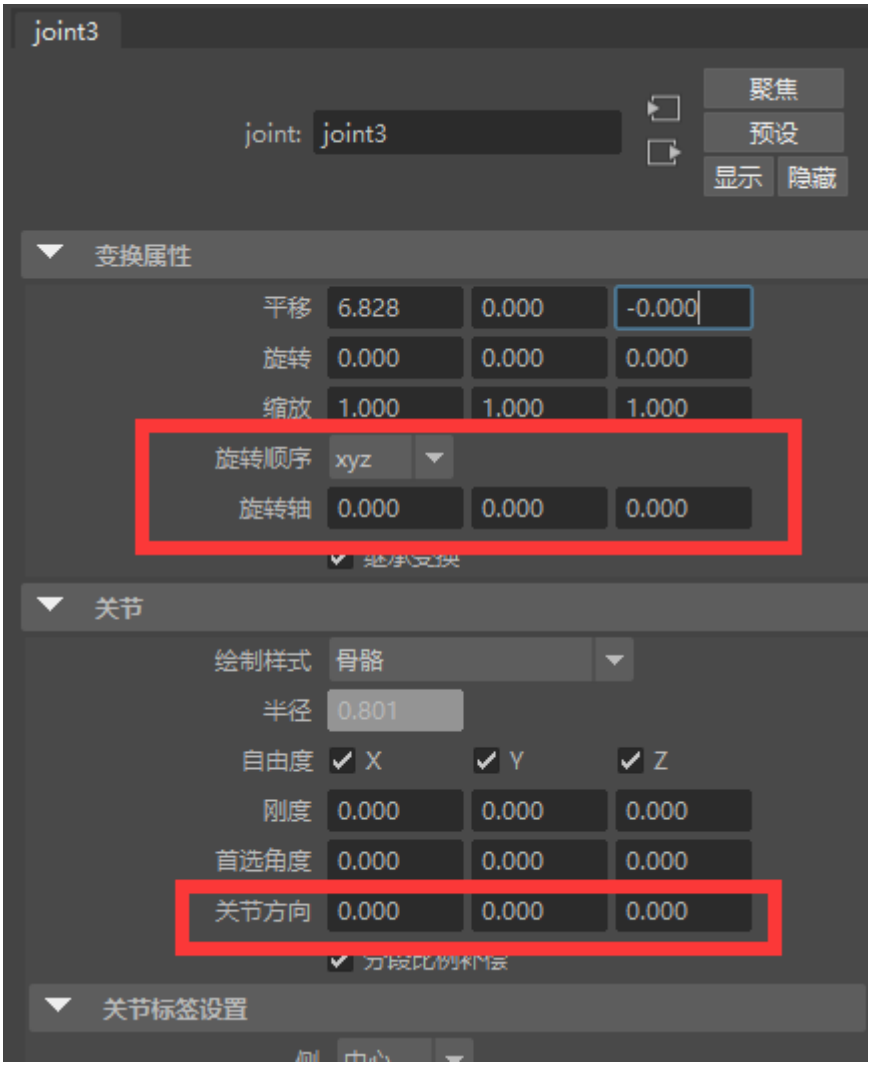
统一标准为T Poss，确认全部关节方向为世界方向。 全部关节方向一致，不允许左右对称。不允许使用





在T Poss之下，除去手部大拇指的特殊处理，其它关节不允许有旋转值。（如带了旋转值，在动捕前程序会默认清0）

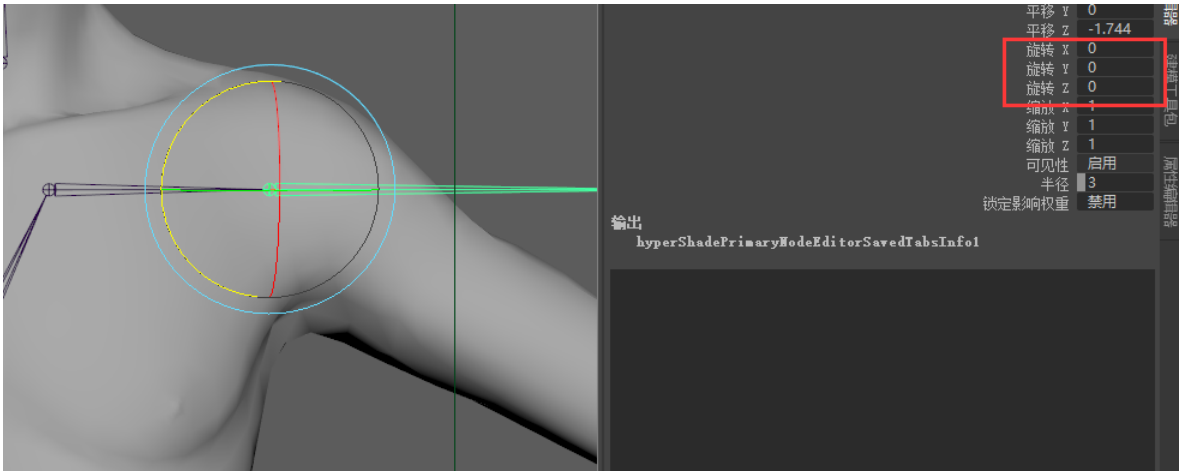
旋转顺序为"XYZ",旋转轴不可以有数值，关节方向不可以有数值。



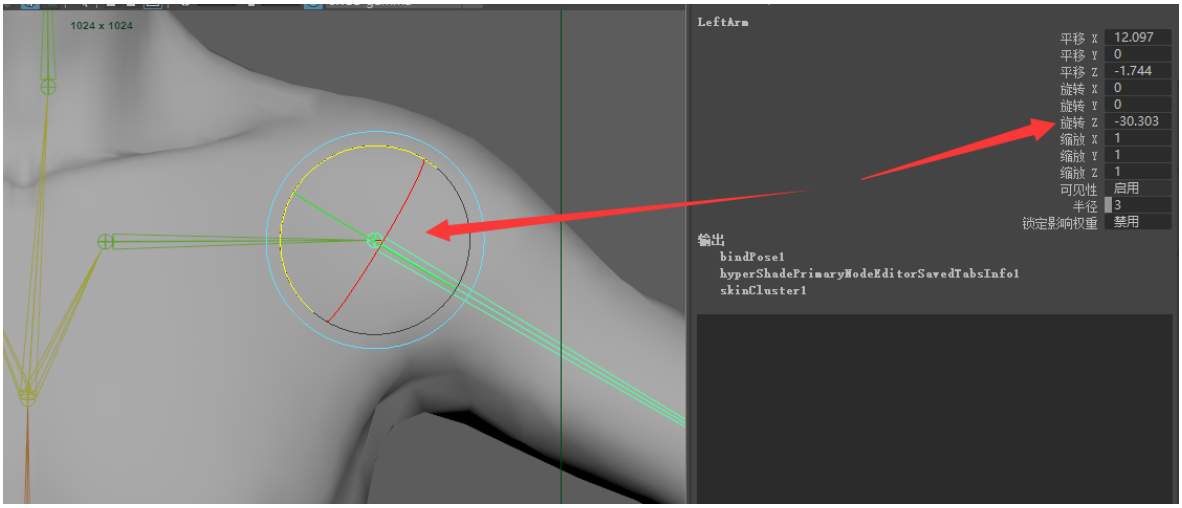
建议完整使用我们提供的骨骼去直接适配角色。按本规范执行。

3、模型不是标准TPose的情况

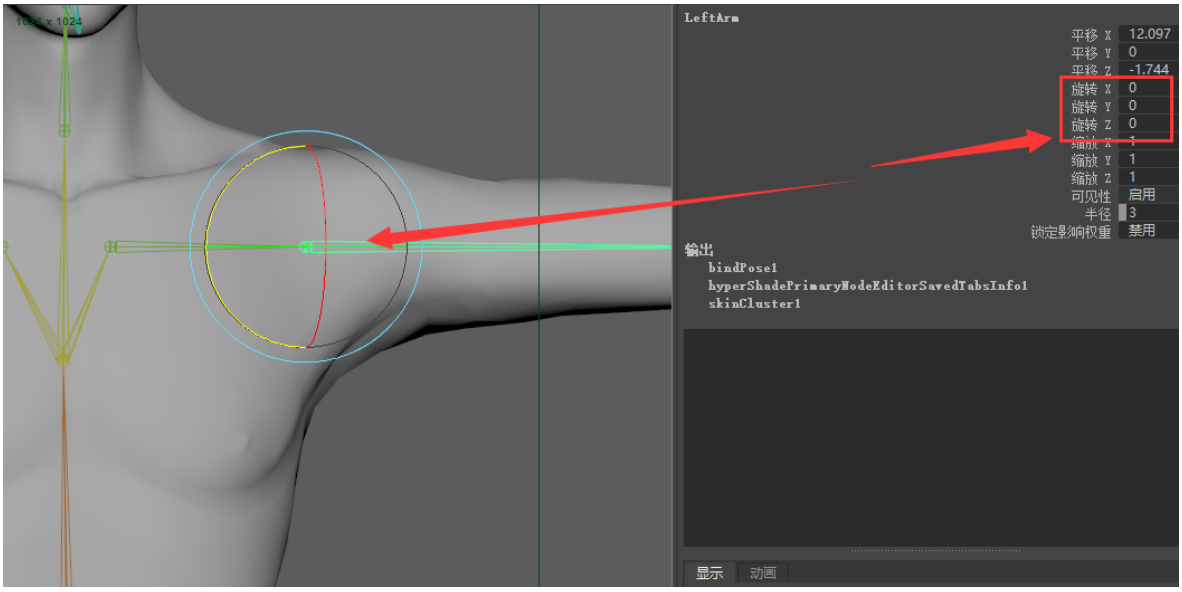
如果模型不是T Pose，请用T Pose骨骼旋转匹配模型姿势，刷好权重后，去掉旋转值，让模型姿势变成 T Pose。以下举例。



首先，没有旋转的情况下，匹配位置。



然后，旋转以匹配模型。

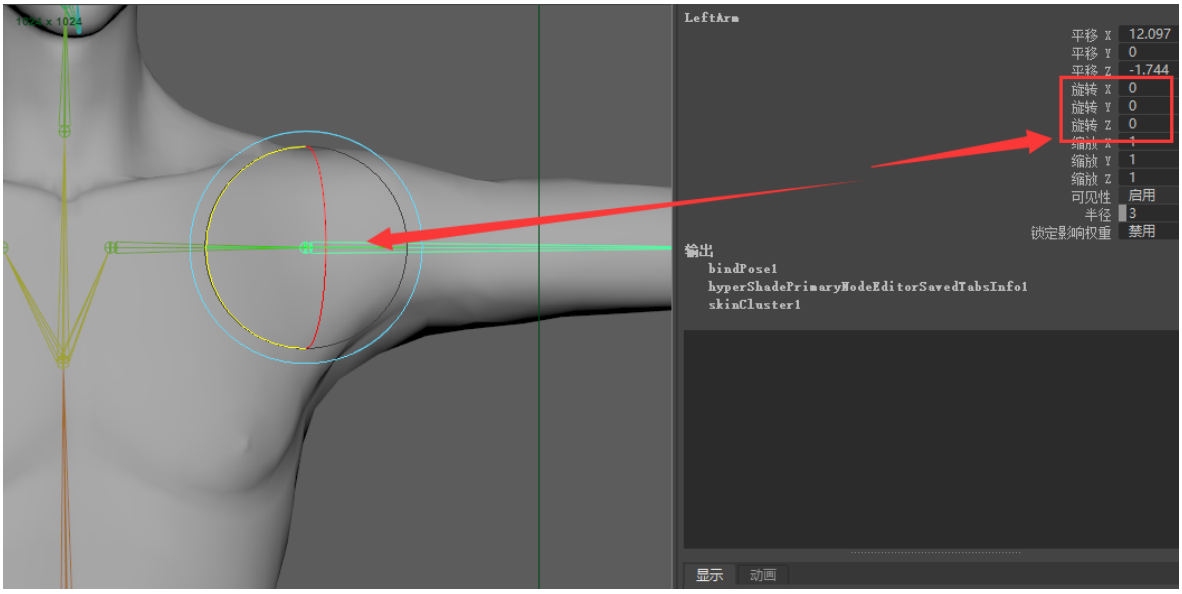


刷好权重之后，去掉旋转值，将模型变成标准T Poss。

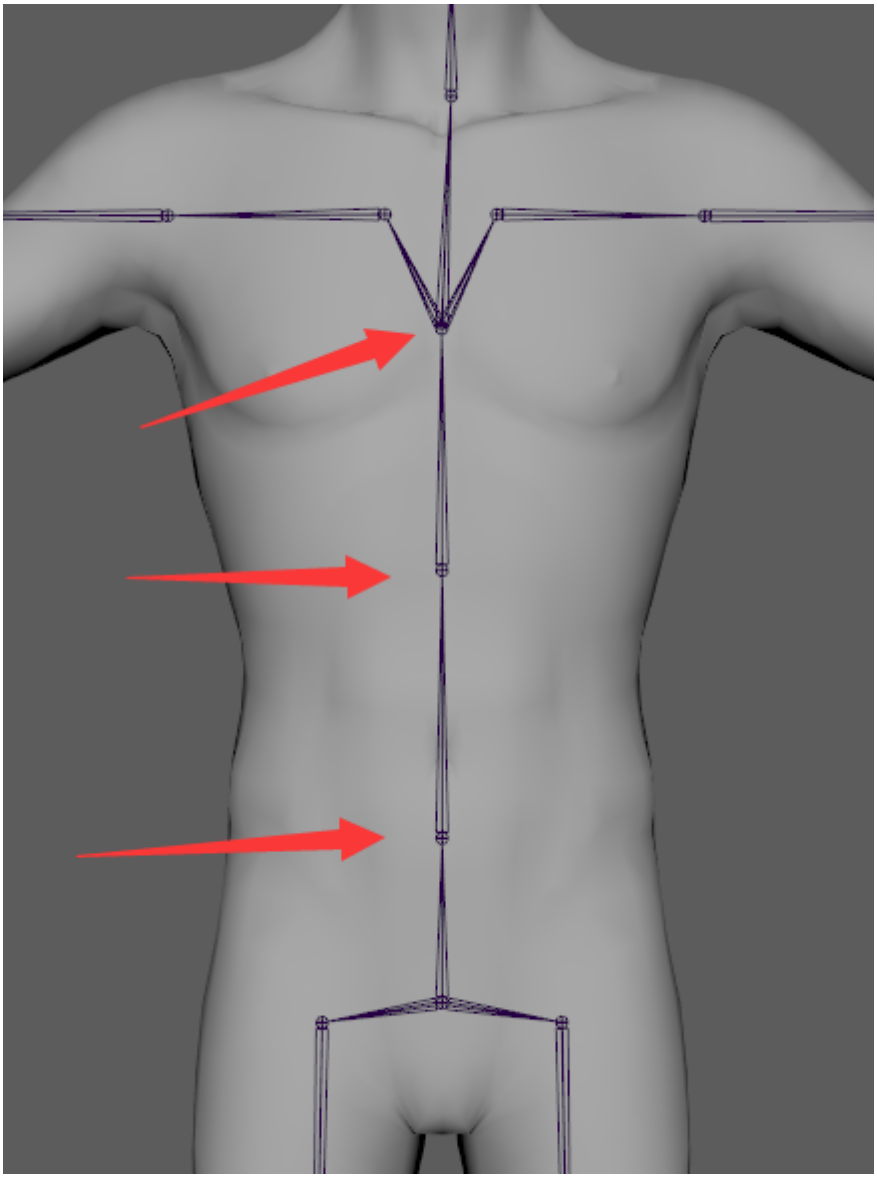
4、骨骼位置建议

肩膀与大臂

肩膀与大臂距离不宜设置过长和过短，根据模型肩宽来决定。



脊柱3节骨骼位置



分别放至：

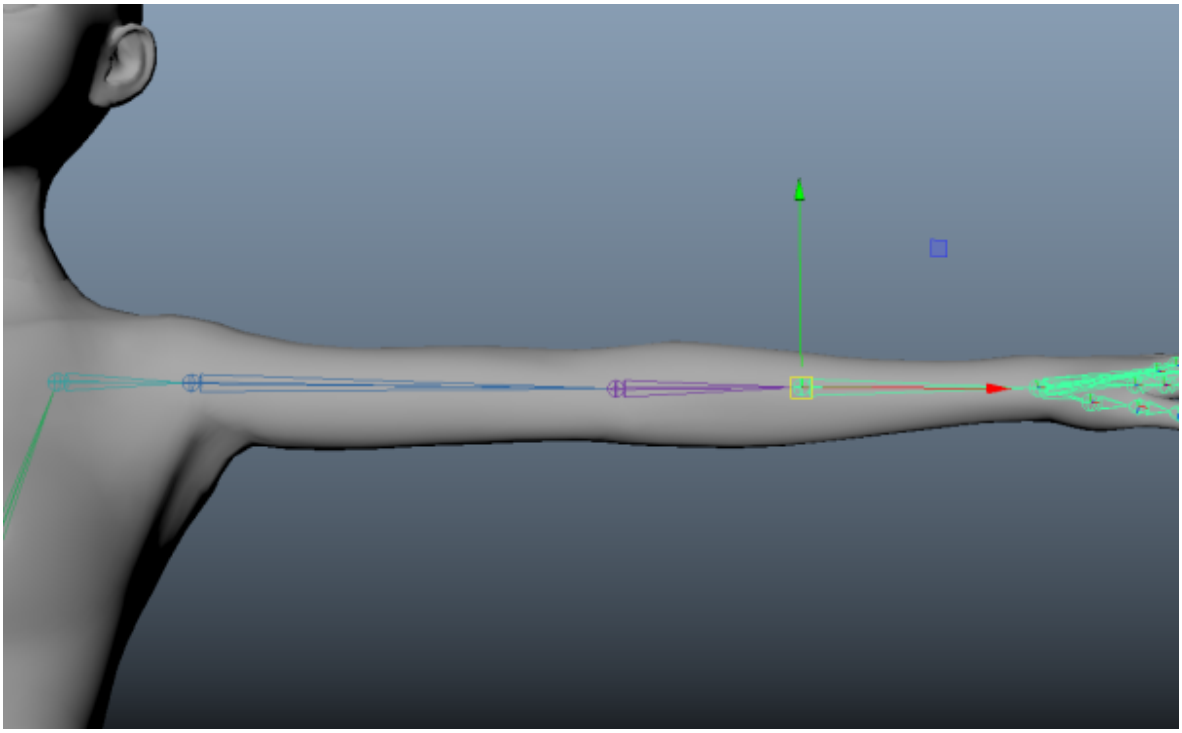
胯部体块上部交界处；

胸部体块下部交界处；

胸部体块中心。

弯腰动作测试合理即可

旋转骨摆放位置

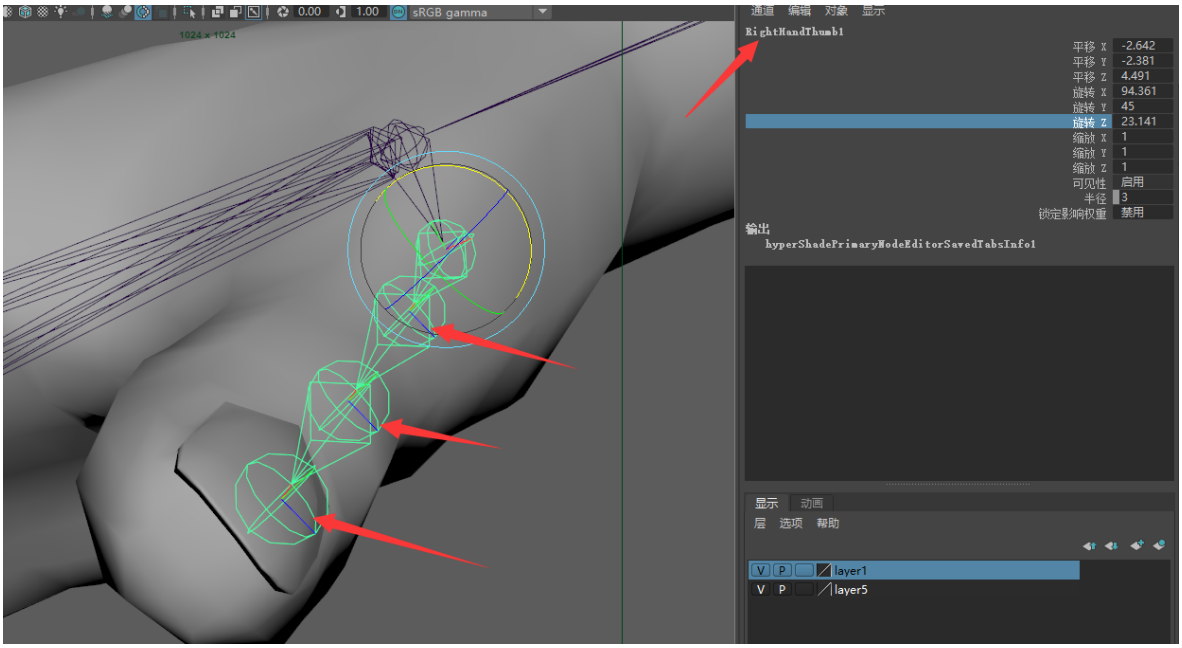


大致放在手肘和手腕中间位置即可

适当刷一下权重，不要让手腕旋转扭曲即可

手指拇指关节处理

这部分容易出错，有疑问请主动联系支持人员。



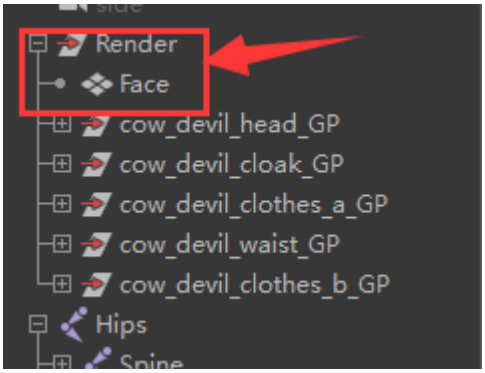
如图，用HandThumb1来匹配位置与旋转，使3节子关节蓝轴（Z轴）平行于指甲盖。

动捕时，右手拇指沿+Z轴实现内弯，左手则是沿-Z轴。

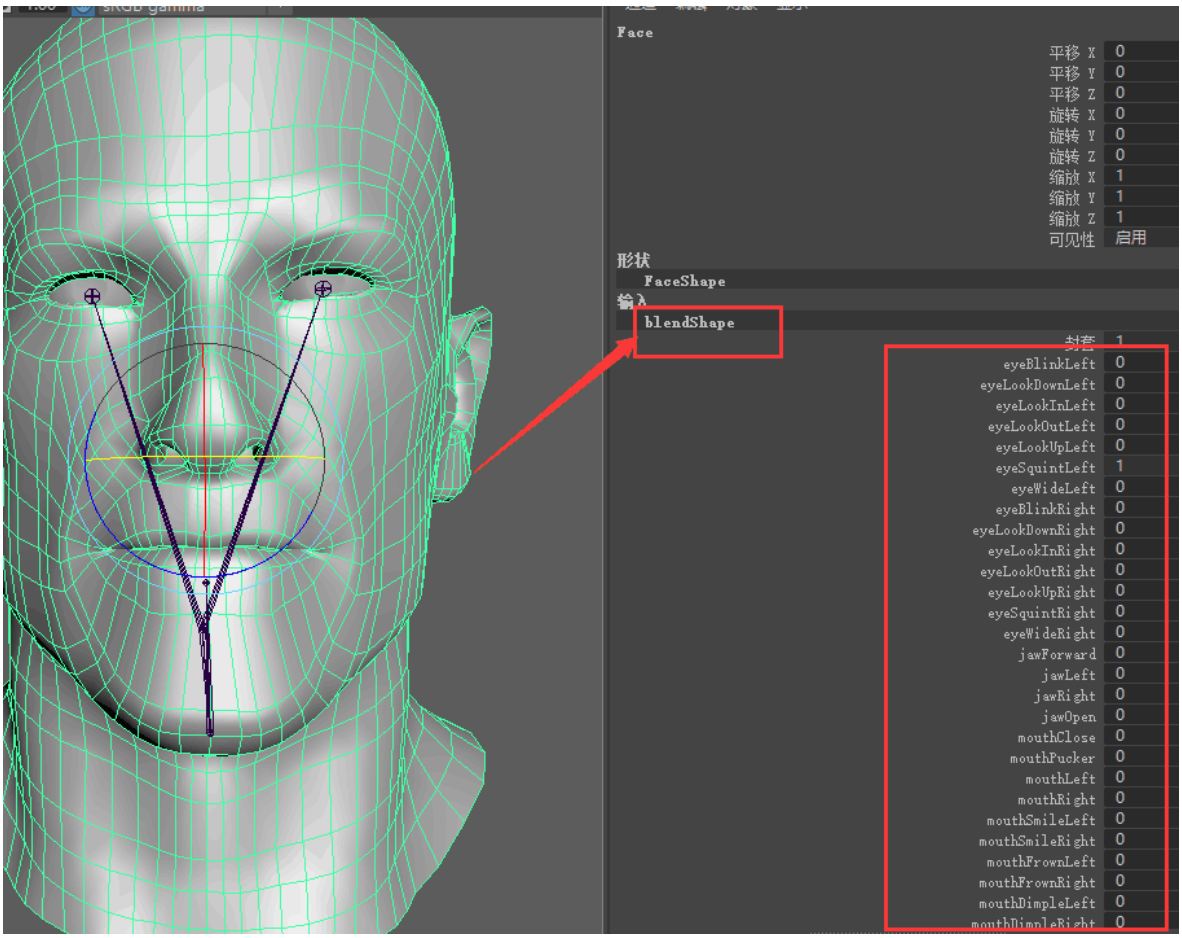
DM面部绑定规范

一、面部绑定注意事项

1.眉毛、睫毛、眼球、舌头、牙齿等参与混合变形的附属模型，应与面部合并为一个模型，并且命名为"Face"（24卡通规范命名为"Face_24",32卡通规范命名为"Face_42"），放置于Render下。如图。父层级一定要，命名为Render。



2.所有BlendShape需按照顺序统一指定给Face，属性中的BlendShape命名后面不应有任何后缀，必须是"blendShape"，如图。



3.面部绑定BlendShape原型数量有3种规范以下说明。

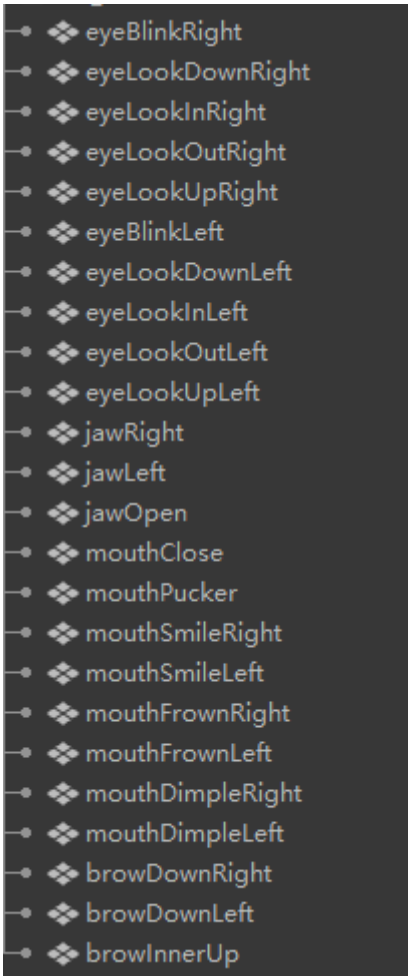
- 24个原型适合面部结构简单的符号化角色。面部所在Mesh命名为"Face_24"。
- 32个原型适合面部结构不复杂的卡通角色。面部所在Mesh命名为"Face_32"。
- 42个原型适合面部结构复杂的卡通角色和写实角色。面部所在Mesh命名为"Face_42"。
- 51个原型适合写实角色。面部所在Mesh命名为"Face_51"。
- Mesh命名无后缀------"Face"，默认以42个原型的情况处理。

面部BlenShape排列顺序

注意：顺序与命名必须完全按照要求。

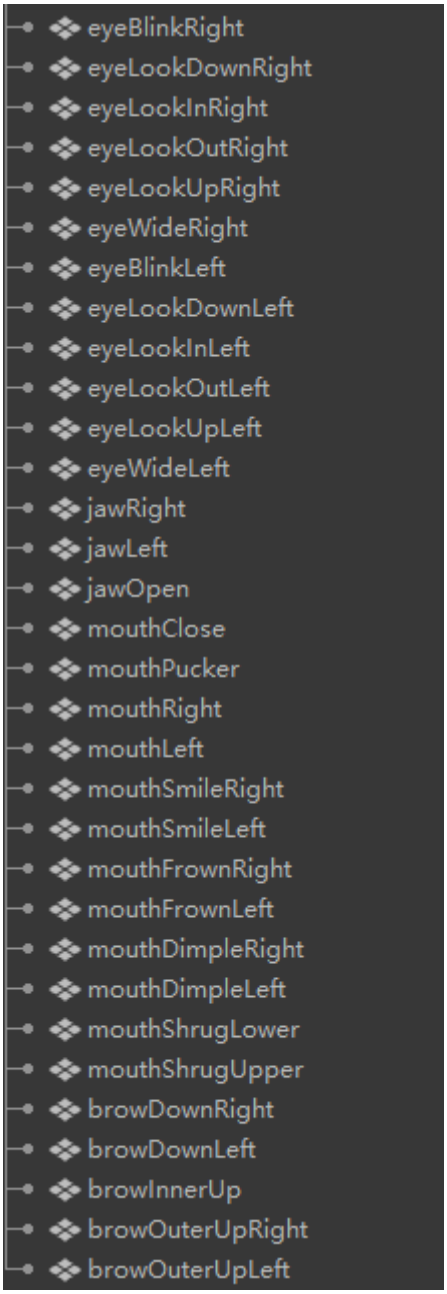
有4种规范，以下分别列举。

24个原型------"Face_24"以下列举.。



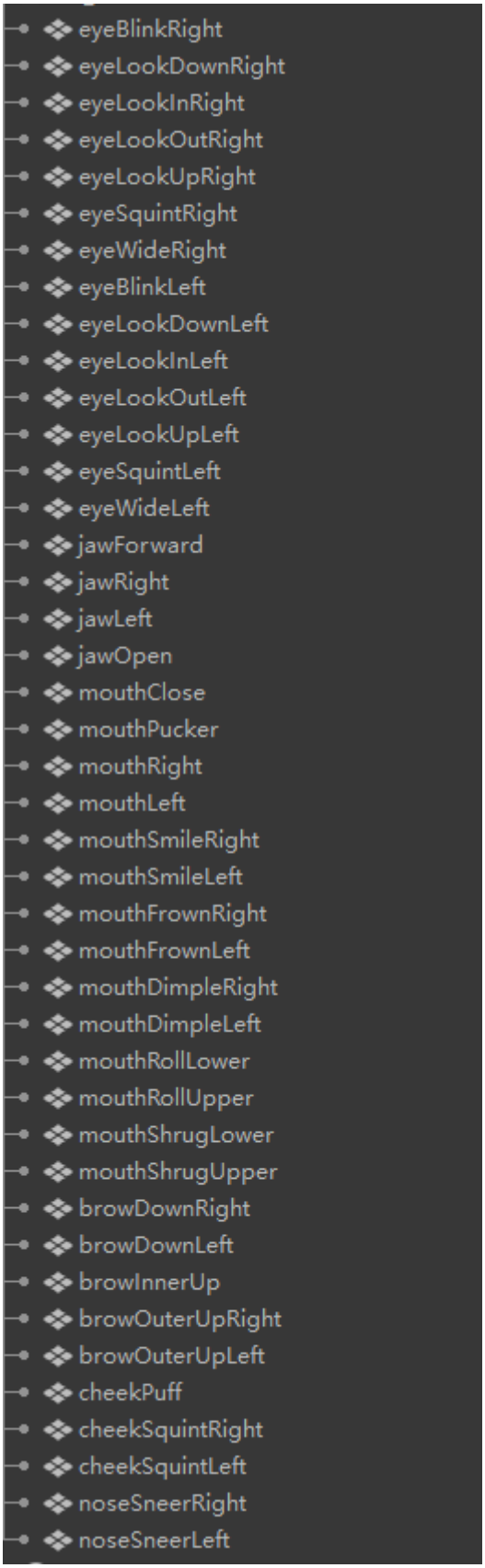
1.eyeBlinkRight 右眼闭眼
2.eyeLookDownRight 右眼向下看
3.eyeLookInRight 右眼向内看

4.eyeLookOutRight 右眼向外看
5.eyeLookUpRight 右眼向上看
6.eyeBlinkLeft 左眼闭眼
7.eyeLookDownLeft 左眼向下看
8.eyeLookInLeft 左眼向内看
9.eyeLookOutLeft 左眼向外看
10.eyeLookUpLeft 左眼向上看
11.jawRight 下颌向右
12.jawLeft 下颌向左
13.jawOpen 下颌向下开口
14.mouthClose 闭嘴
15.mouthPucker 噘嘴
16.mouthSmileRight 右嘴角微笑
17.mouthSmileLeft 左嘴角微笑
18.mouthFrownRight 右嘴角下挑
19.mouthFrownLeft 左嘴角下挑
20.mouthDimpleRight 右嘴角向后动
21.mouthDimpleLeft 左嘴角向后动
22.browDownRight 右眉毛凶眉
23.browDownLeft 左眉毛凶眉
24.browInnerUp 内挑眉
32个原型------"Face_32"以下列举。



1.eyeBlinkRight 右眼闭眼
2.eyeLookDownRight 右眼向下看
3.eyeLookInRight 右眼向内看
4.eyeLookOutRight 右眼向外看
5.eyeLookUpRight 右眼向上看
6.eyeWideRight 右眼瞪眼
7.eyeBlinkLeft 左眼闭眼
8.eyeLookDownLeft 左眼向下看
9.eyeLookInLeft 左眼向内看
10.eyeLookOutLeft 左眼向外看
11.eyeLookUpLeft 左眼向上看
12.eyeWideLeft 左眼瞪眼
13.jawRight 下颌向右

14.jawLeft 下颌向左
15.jawOpen 张嘴
16.mouthClose 闭嘴唇
17.mouthPucker 噘嘴
18.mouthRight 嘴唇向右上挑
19.mouthLeft 嘴唇向左上挑
20.mouthSmileRight 右嘴角上挑
21.mouthSmileLeft 左嘴角上挑
22.mouthFrownRight 右嘴角下挑
23.mouthFrownLeft 左嘴角下挑
24.mouthDimpleRight 右嘴角向后动
25.mouthDimpleLeft 左嘴角向后动
26.mouthShrugLower 下嘴唇向外移动
27.mouthShrugUpper 上嘴唇向外移动
28.browDownRight 右眉毛凶眉
29.browDownLeft 左眉毛凶眉
30.browInnerUp 内挑眉
31.browOuterUpRight 右眉毛挑眉
32.browOuterUpLeft 左眉毛挑眉
42个原型------"Face_42"以下列举

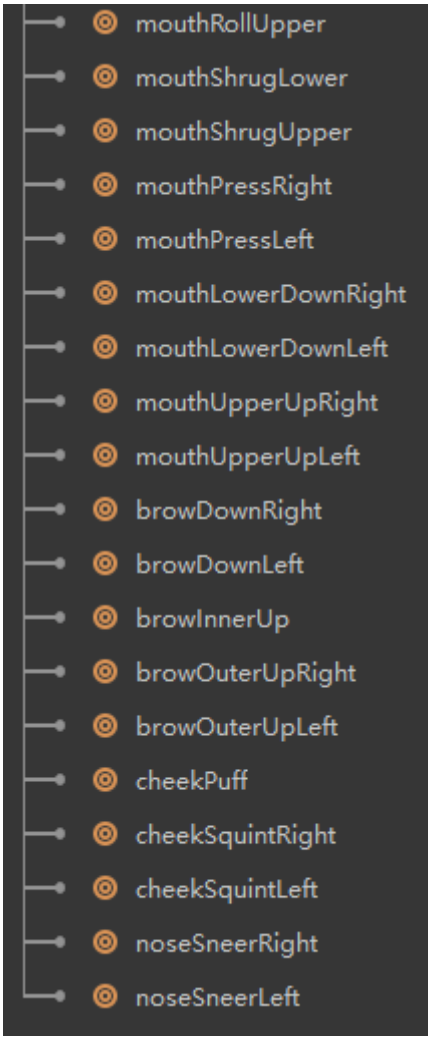


1.eyeBlinkRight 右眼闭眼

2.eyeLookDownRight 右眼向下看

3.eyeLookInRight 右眼向内看

4.eyeLookOutRight 右眼向外看
5.eyeLookUpRight 右眼向上看
6.eyeSquintRight 右眼眼睑收缩
7.eyeWideRight 右眼瞪眼
8.eyeBlinkLeft 左眼闭眼
9.eyeLookDownLeft 左眼向下看
10.eyeLookInLeft 左眼向内看
11.eyeLookOutLeft 左眼向外看
12.eyeLookUpLeft 左眼向上看
13.eyeSquintLeft 左眼眼睑收缩
14.eyeWideLeft 左眼瞪眼
15.jawForward 下颌向前
16.jawRight 下颌向右
17.jawLeft 下颌向左
18.jawOpen 张嘴
19.mouthClose 下颌向下同时闭嘴唇
20.mouthPucker 噘嘴
21.mouthRight 嘴唇向右上挑
22.mouthLeft 嘴唇向左上挑
23.mouthSmileRight 右嘴角上挑
24.mouthSmileLeft 左嘴角上挑
25.mouthFrownRight 右嘴角下挑
26.mouthFrownLeft 左嘴角下挑
27.mouthDimpleRight 右嘴角向后动
28.mouthDimpleLeft 左嘴角向后动
29.mouthRollLower 下唇向嘴内侧移动
30.mouthRollUpper 上唇向嘴内侧移动
31.mouthShrugLower 下嘴唇向外移动
32.mouthShrugUpper 上嘴唇向外移动
33.browDownRight 右眉毛外部向下挑
34.browDownLeft 左眉毛外部向下挑
35.browInnerUp 内挑眉
36.browOuterUpRight 右眉毛外部向上挑
37.browOuterUpLeft 左眉毛外部向上挑
38.cheekPuff 两个脸颊向外运动
39.cheekSquintRight 右脸颊向上移动
40.cheekSquintLeft 左脸颊向上移动
41.noseSneerRight 鼻孔右侧抬高
42.noseSneerLeft 鼻孔左侧抬高
51个原型------"Face_51"以下列举

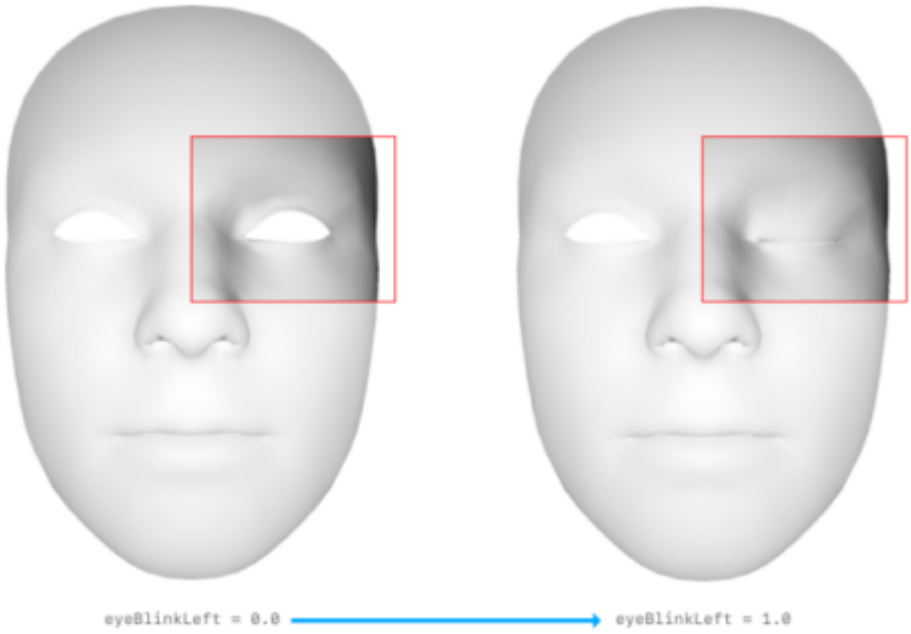


1.eyeBlinkRight 右眼闭眼
2.eyeLookDownRight 右眼向下看
3.eyeLookInRight 右眼向内看
4.eyeLookOutRight 右眼向外看
5.eyeLookUpRight 右眼向上看
6.eyeSquintRight 右眼眼睑收缩
7.eyeWideRight 右眼瞪眼
8.eyeBlinkLeft 左眼闭眼
9.eyeLookDownLeft 左眼向下看
10.eyeLookInLeft 左眼向内看
11.eyeLookOutLeft 左眼向外看
12.eyeLookUpLeft 左眼向上看
13.eyeSquintLeft 左眼眼睑收缩
14.eyeWideLeft 左眼瞪眼
15.jawForward 下颌向前
16.jawRight 下颌向右
17.jawLeft 下颌向左
18.jawOpen 张嘴
19.mouthClose 下颌向下同时闭嘴唇

- blendShape
  - eyeBlinkRight
  - eyeLookDownRight
  - eyeLookInRight
  - eyeLookOutRight
  - eyeLookUpRight
  - eyeSquintRight
  - eyeWideRight
  - eyeBlinkLeft
  - eyeLookDownLeft
  - eyeLookInLeft
  - eyeLookOutLeft
  - eyeLookUpLeft
  - eyeSquintLeft
  - eyeWideLeft
  - jawForward
  - jawRight
  - jawLeft
  - jawOpen
  - mouthClose
  - mouthFunnel
  - mouthPucker
  - mouthRight
  - mouthLeft
  - mouthSmileRight
  - mouthSmileLeft
  - mouthFrownRight
  - mouthFrownLeft
  - mouthDimpleRight
  - mouthDimpleLeft
  - mouthStretchRight
  - mouthStretchLeft
  - mouthRollLower

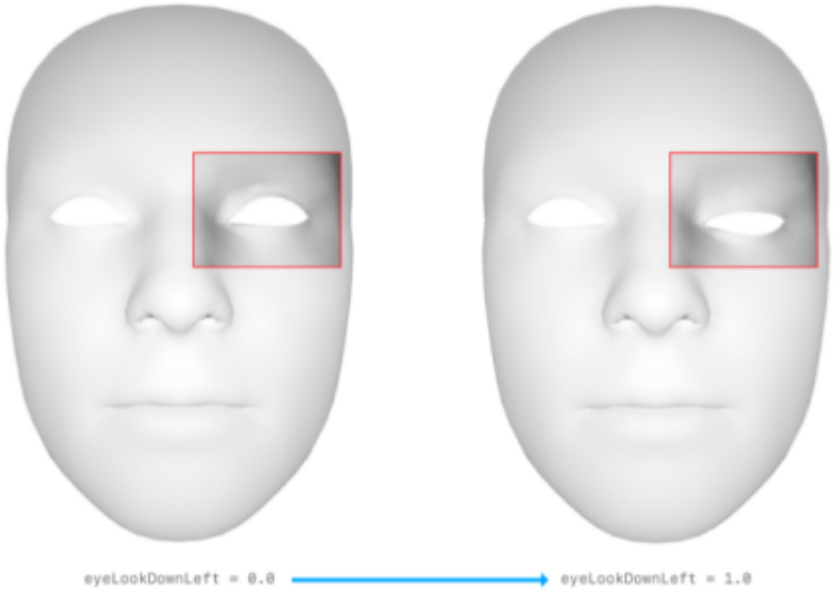
20.mouthFunnel 双唇收缩成开放形状
21.mouthPucker 噘嘴
22.mouthRight 嘴唇向右上挑
23.mouthLeft 嘴唇向左上挑
24.mouthSmileRight 右嘴角上挑
25.mouthSmileLeft 左嘴角上挑
26.mouthFrownRight 右嘴角下挑
27.mouthFrownLeft 左嘴角下挑
28.mouthDimpleRight 右嘴角向后动
29.mouthDimpleLeft 左嘴角向后动
30.mouthStretchRight 右嘴角向左移动
31.mouthStretchLeft 左嘴角向右移动
32.mouthRollLower 下唇向嘴内侧移动
33.mouthRollUpper 上唇向嘴内侧移动
34.mouthShrugLower 下嘴唇向外移动
35.mouthShrugUpper 上嘴唇向外移动
36.mouthPressRight 右侧下唇向上压缩
37.mouthPressLeft 左侧下唇向上压缩
38.mouthLowerDownRight 右侧下唇向下运动
39.mouthLowerDownLeft 左侧下唇向下运动
40.mouthUpperUpRight 右侧上唇向上运动
41.mouthUpperUpLeft 左侧上唇向上运动
42.browDownRight 右眉毛外部向下挑
43.browDownLeft 左眉毛外部向下挑
44.browInnerUp 内挑眉
45.browOuterUpRight 右眉毛外部向上挑
46.browOuterUpLeft 左眉毛外部向上挑
47.cheekPuff 两个脸颊向外运动
48.cheekSquintRight 右脸颊向上移动
49.cheekSquintLeft 左脸颊向上移动
50.noseSneerRight 鼻孔右侧抬高
51.noseSneerLeft 鼻孔左侧抬高

全部面部BlendShape原型详细说明



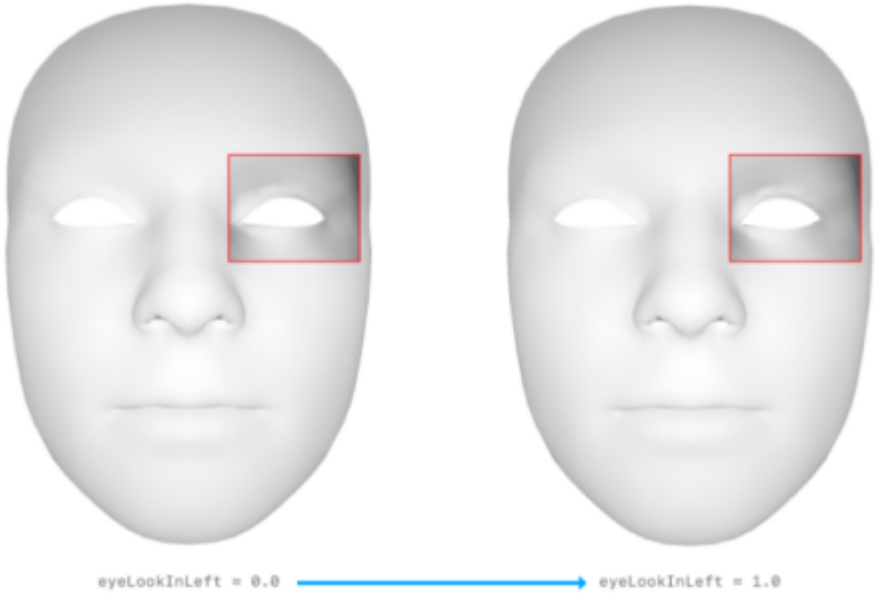
1.

左眼上眼睑闭合。



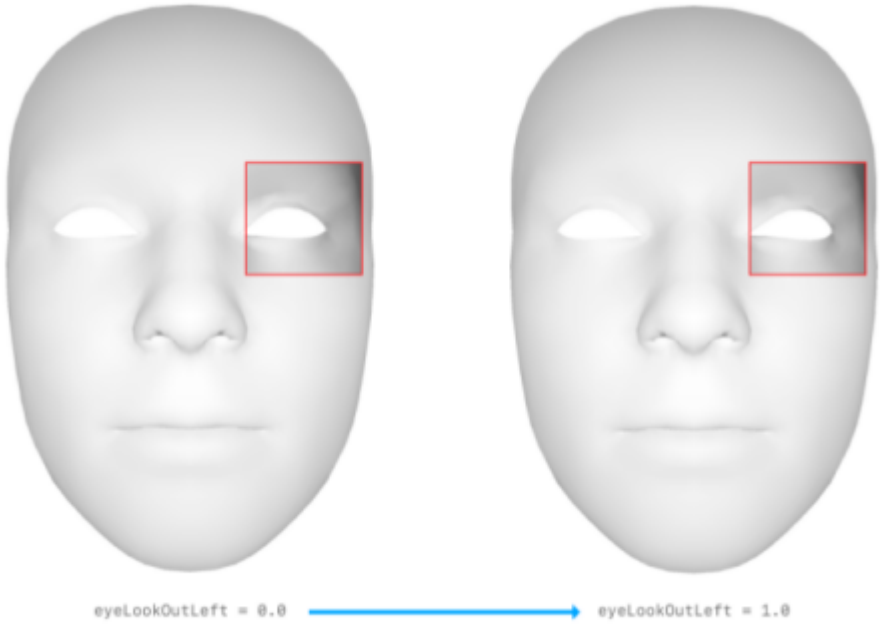
2.

左眼睑向下凝视。



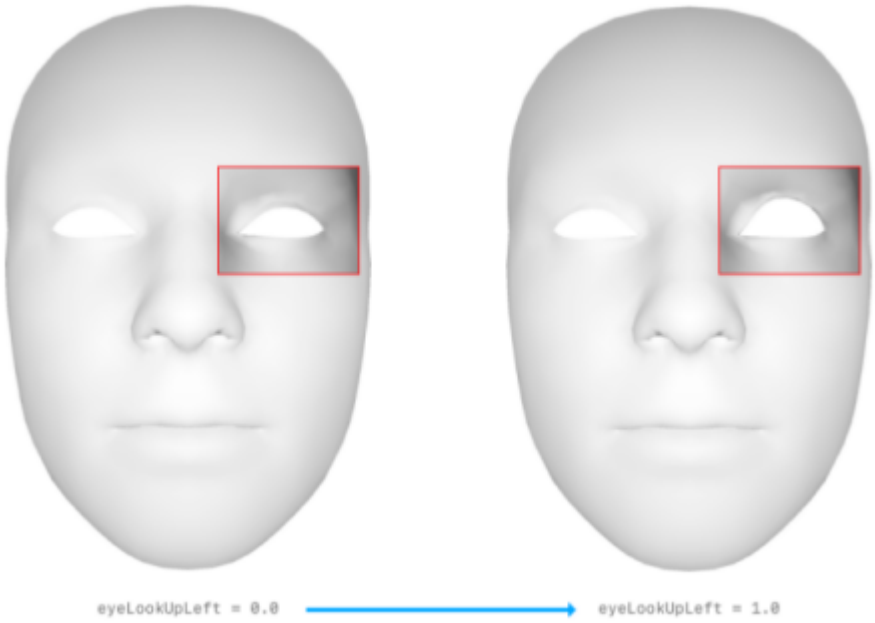
3.

左眼睑向内凝视。



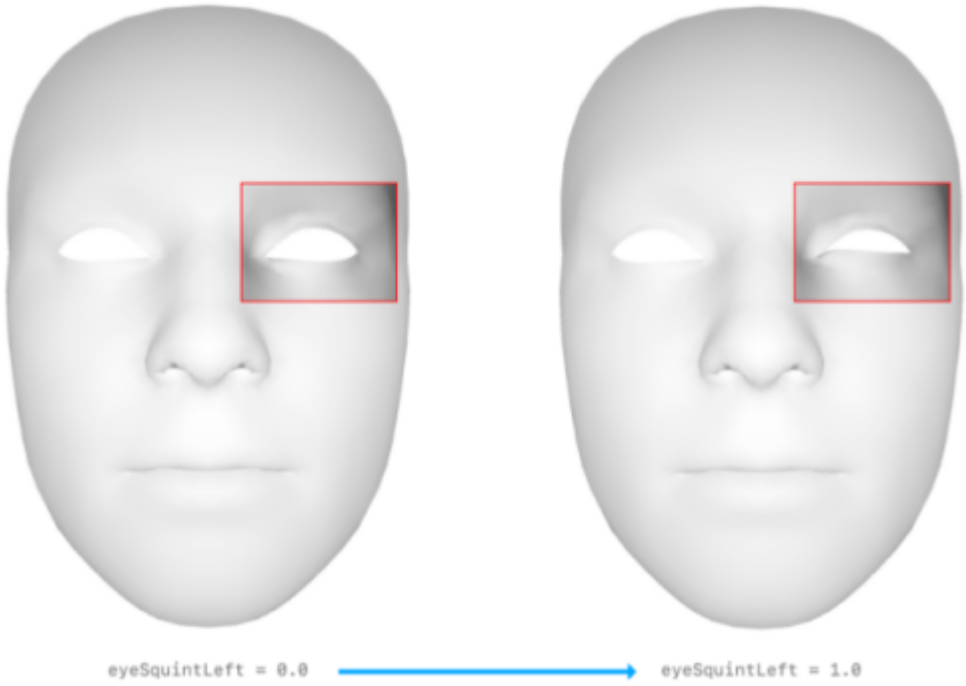
4.

左眼睑向外凝视。



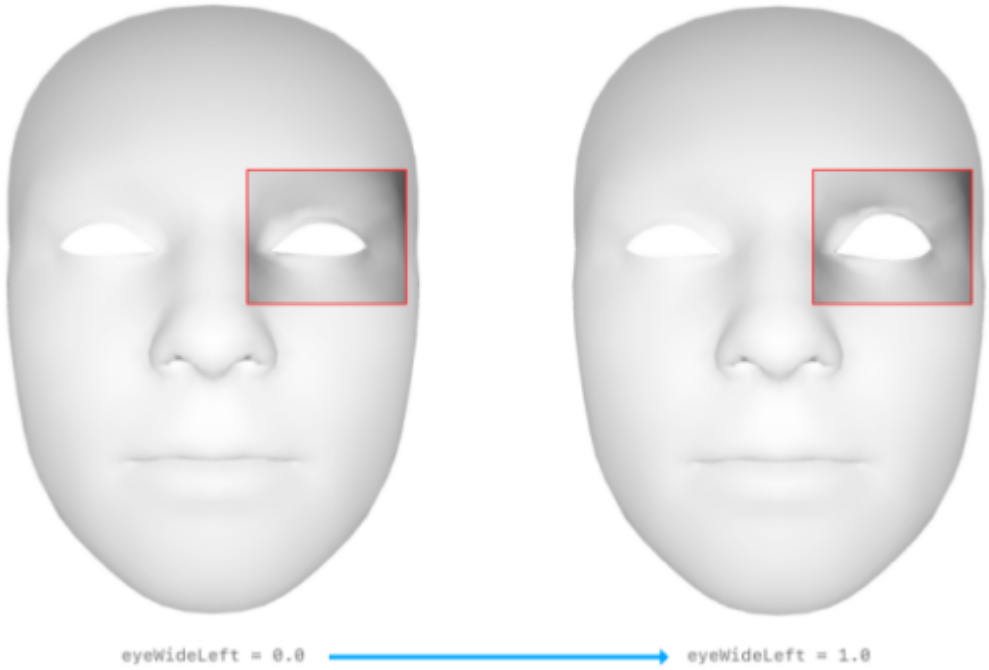
5.

左眼睑向上凝视。



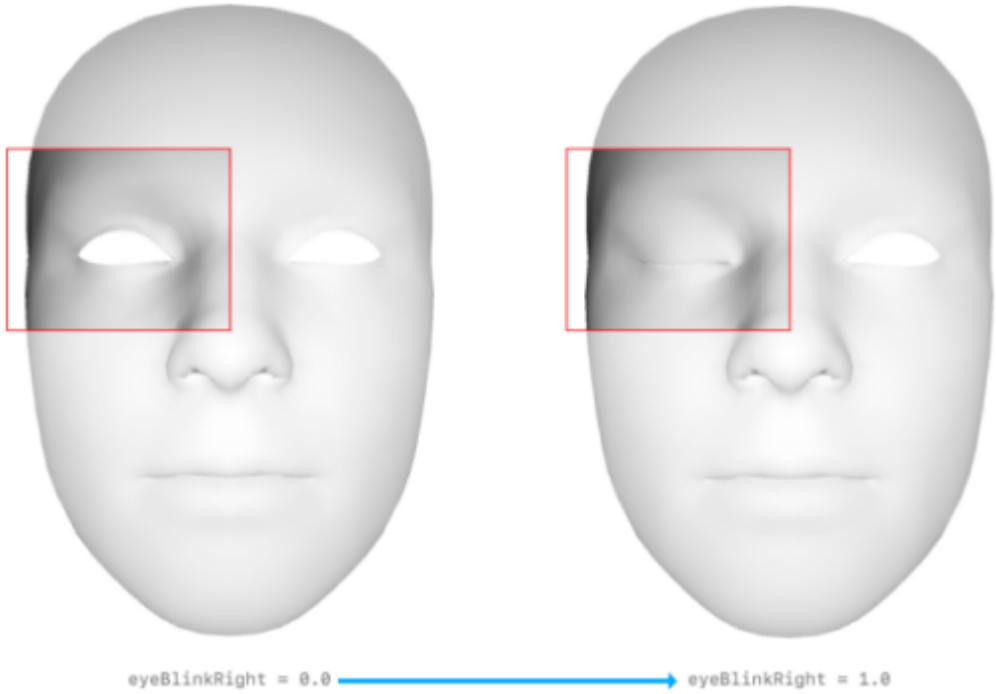
6.

左眼睑收缩。(眯眼)



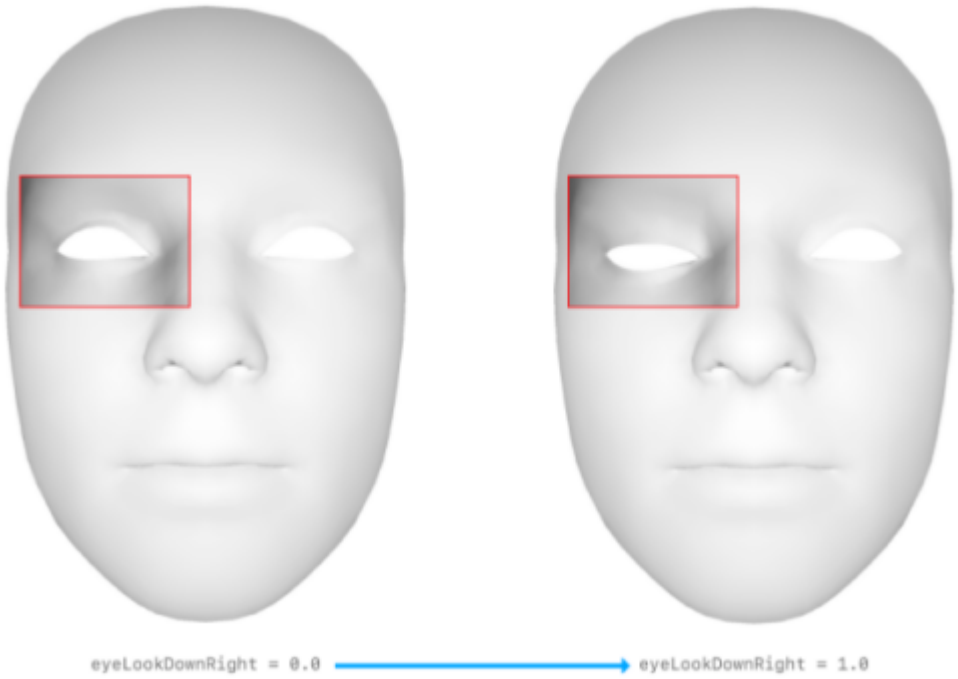
7.

左眼瞪眼。



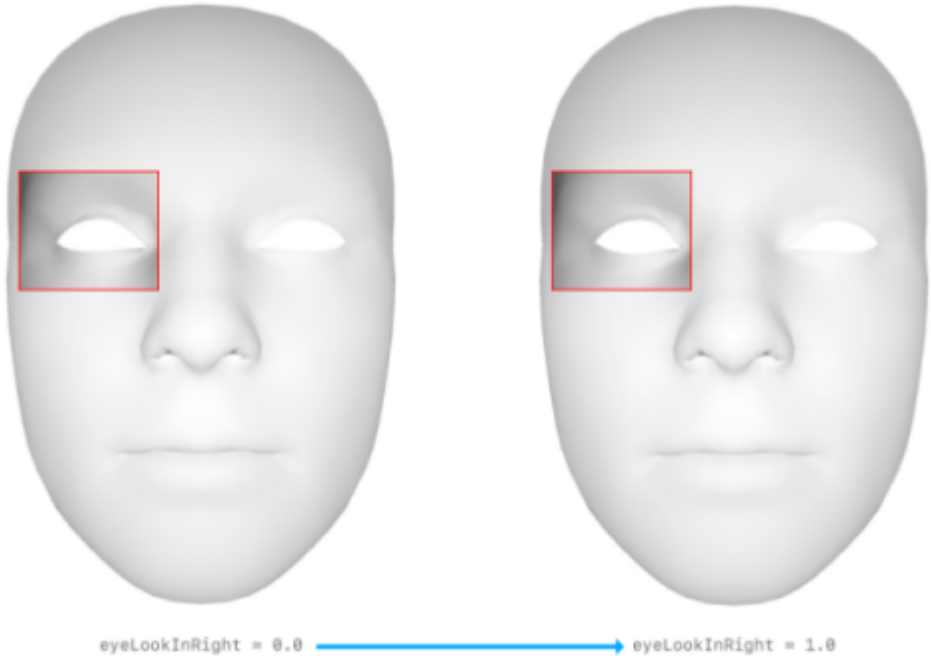
8.

右眼上眼睑闭合。



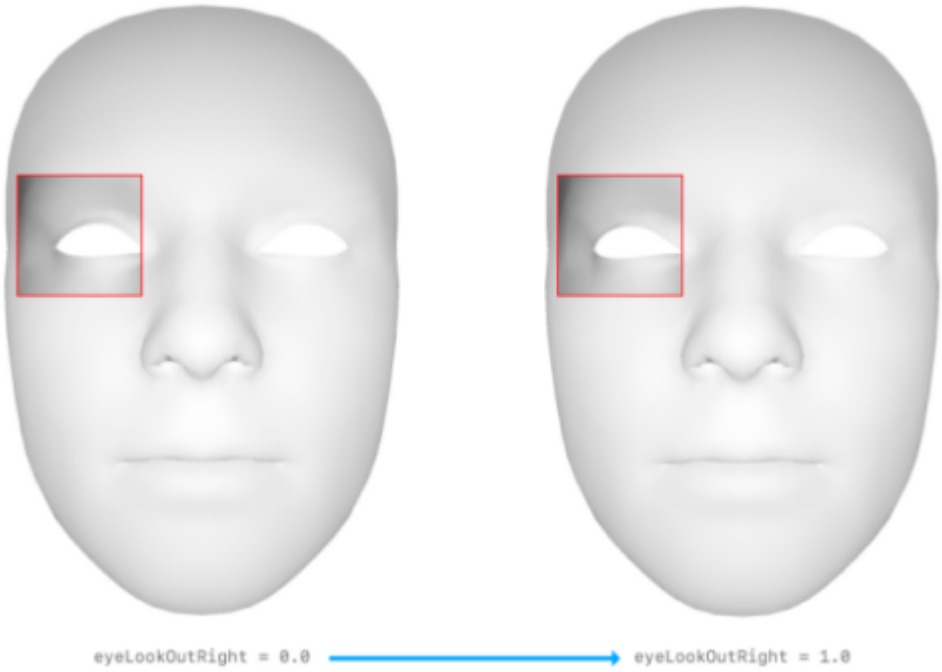
9.

右眼睑向下凝视。



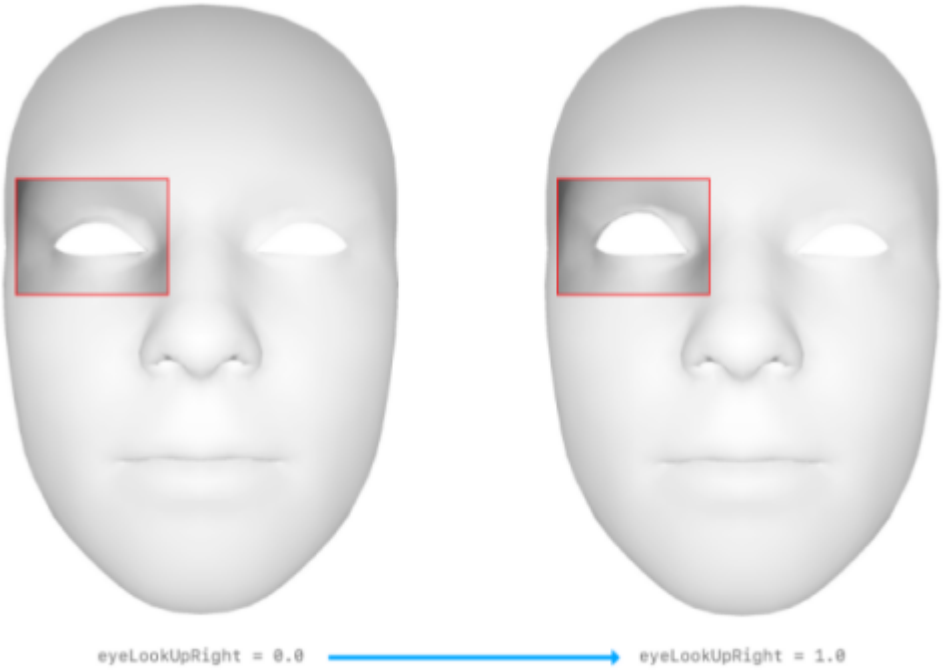
10.

右眼睑向内凝视。



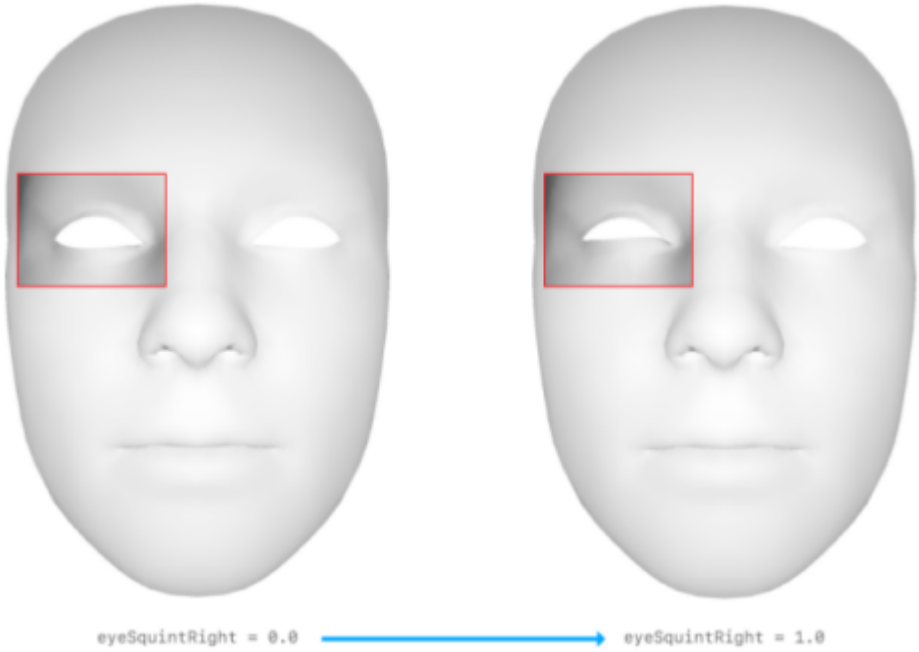
11.

右眼睑向外凝视。



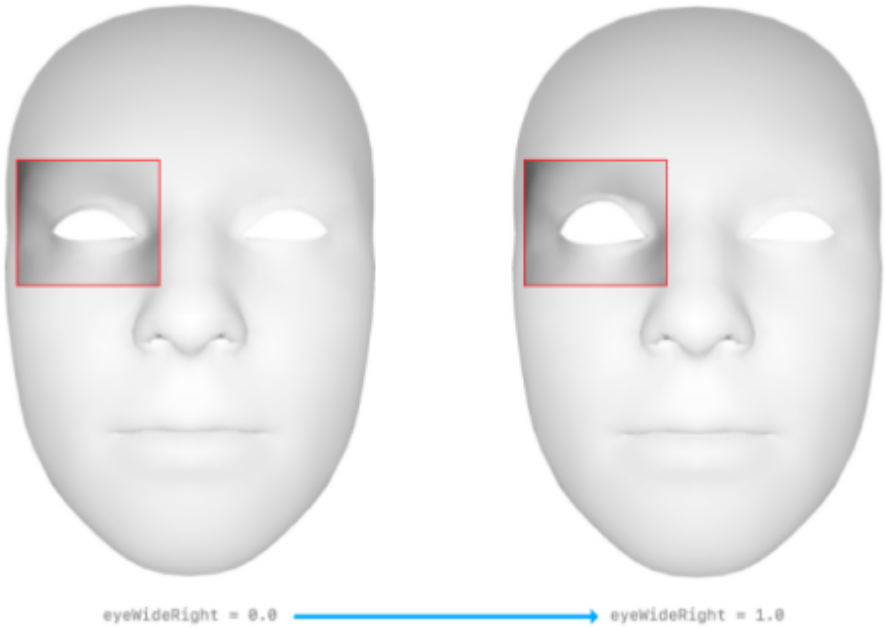
12.

右眼睑向上凝视。



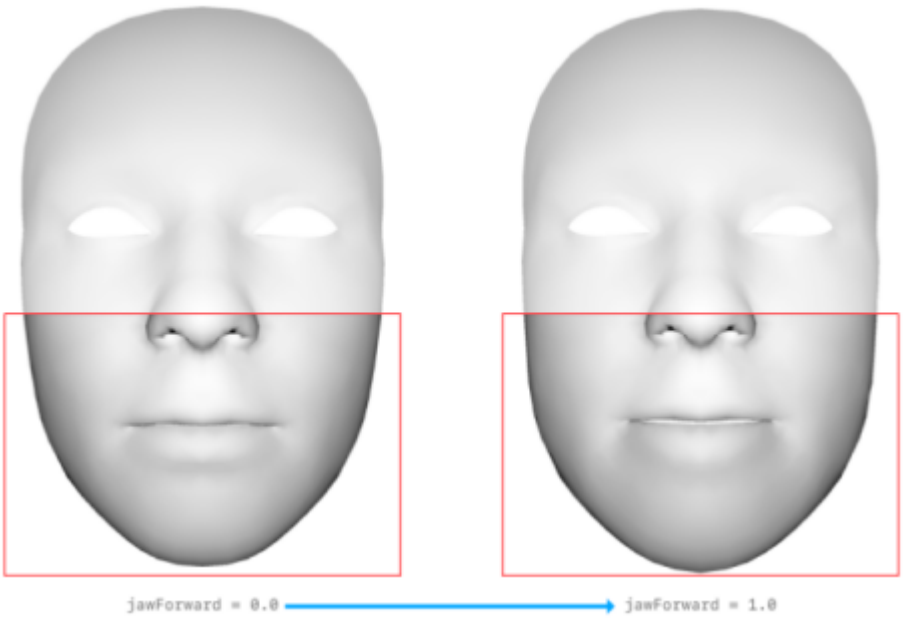
13.

右眼睑收缩。(眯眼)。



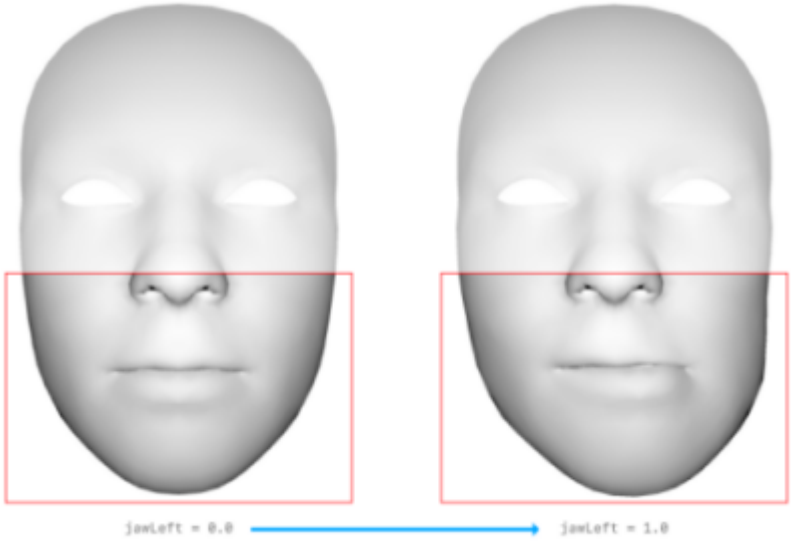
14.

右眼瞪眼。



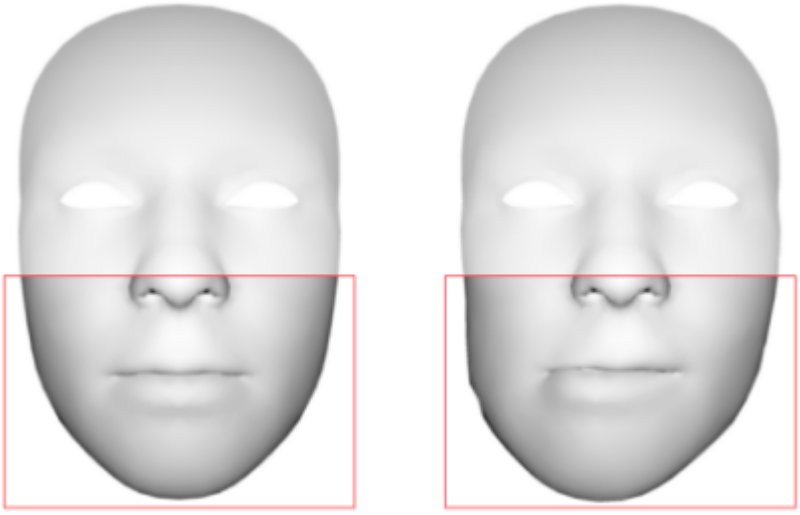
15.

下颌向前运动。



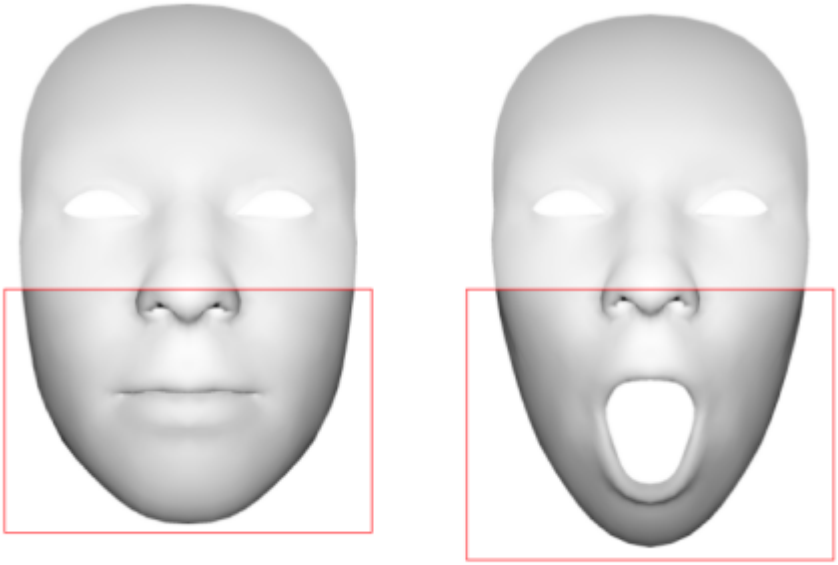
16.

下颌向左运动。



17. jawRight = 0.0 → jawRight = 1.0

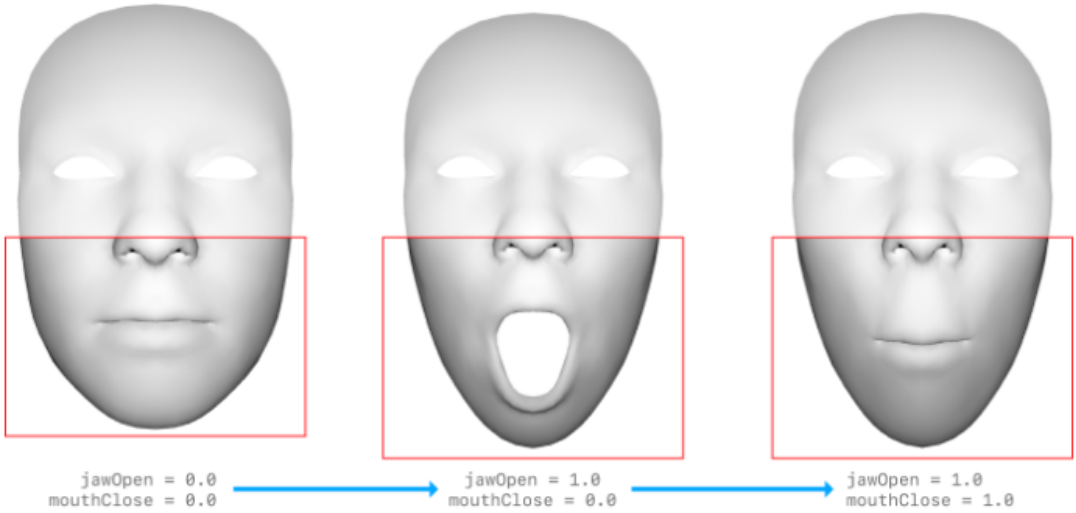
下颌向右运动。



18. jawOpen = 0.0 → jawOpen = 1.0

下颌打开。（张嘴）

19.



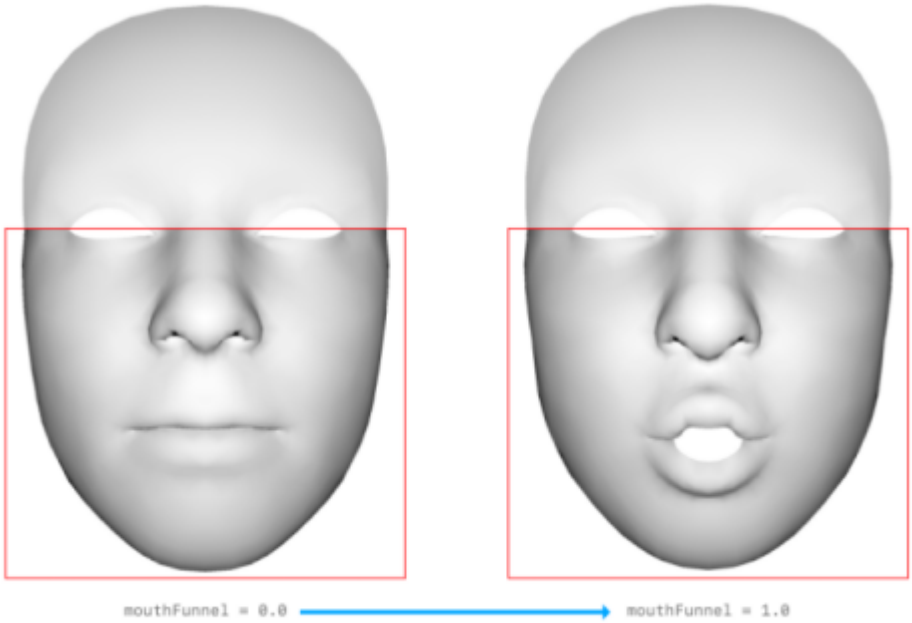
嘴唇闭合。(下颌打开的同时，合上嘴唇)

上图显示了三种状态下的表情：

1.中性面（所有ARFaceAnchor.BlendShapeLocation系数值均为0.0，包括jawOpen和mouthClose）。

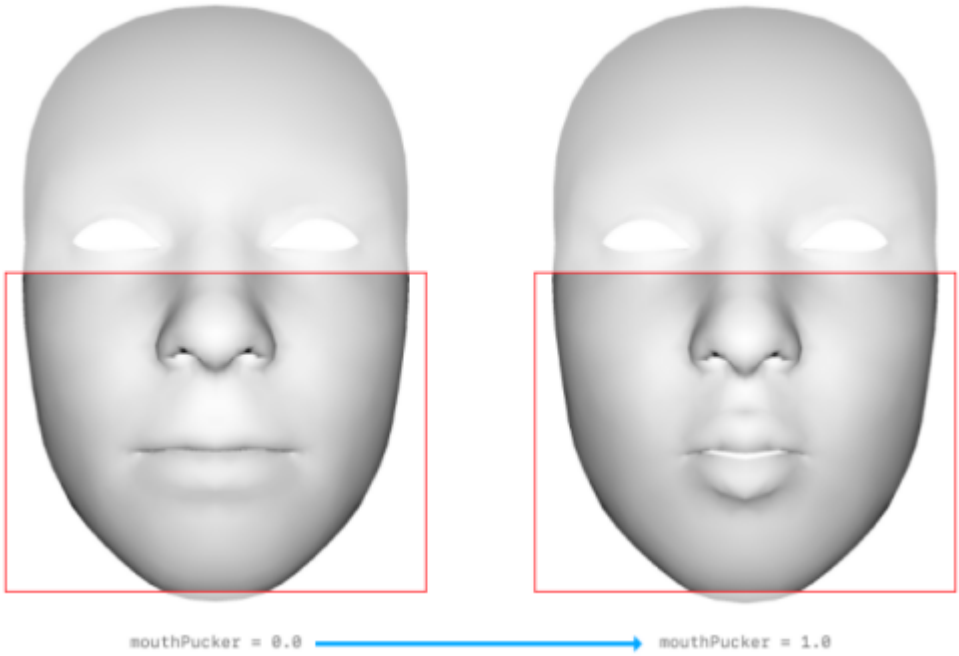
2.仅将jawOpen系数设置为1.0，同时将所有其他系数值（包括mouthClose）保持为0.0。

3.将jawOpen和mouthClose系数都设置为1.0，同时将所有其他系数值保持为0.0。



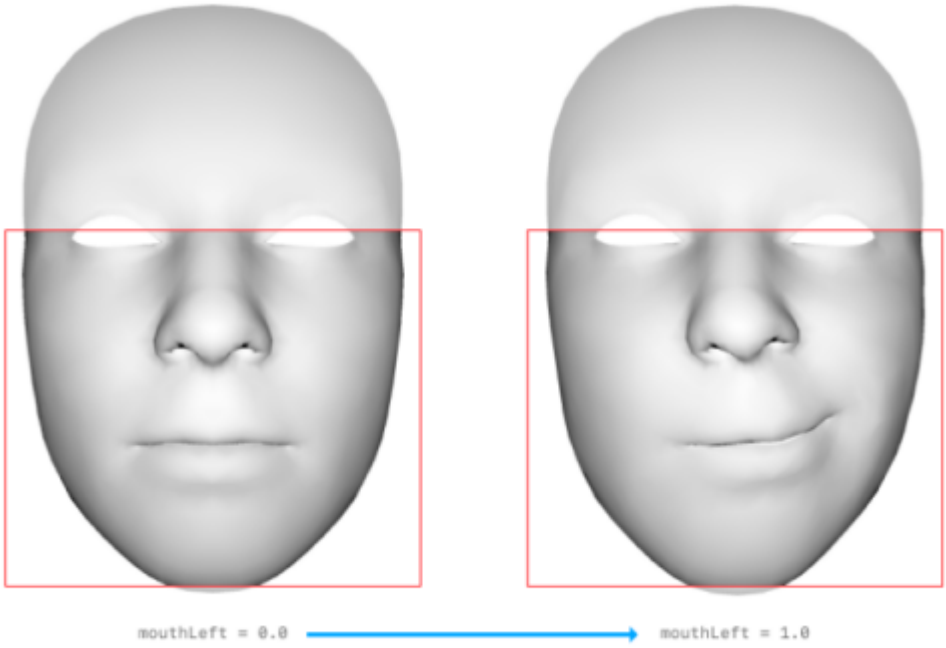
20.

双唇收缩成开放形状。



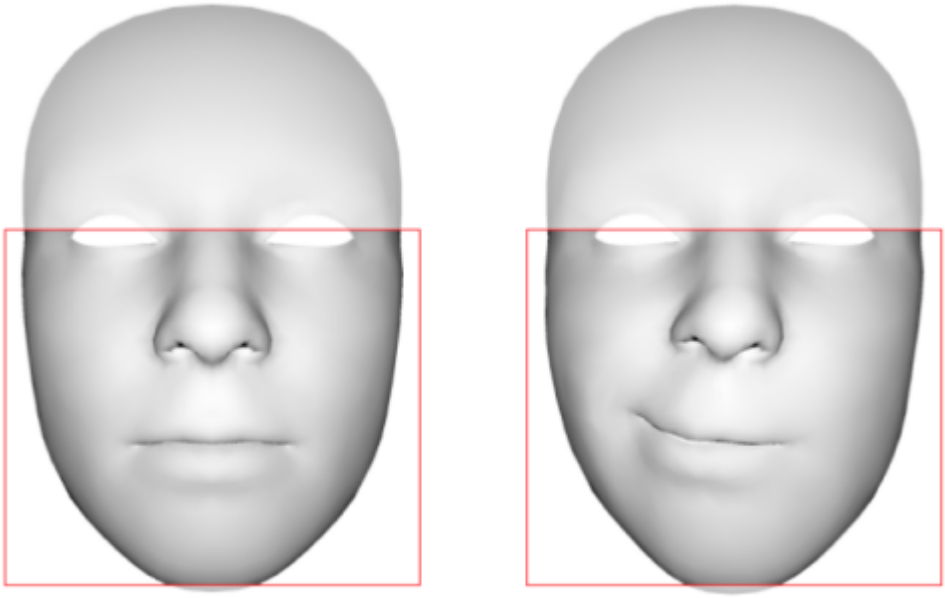
21.

闭合嘴唇的收缩。（发wu的音。）



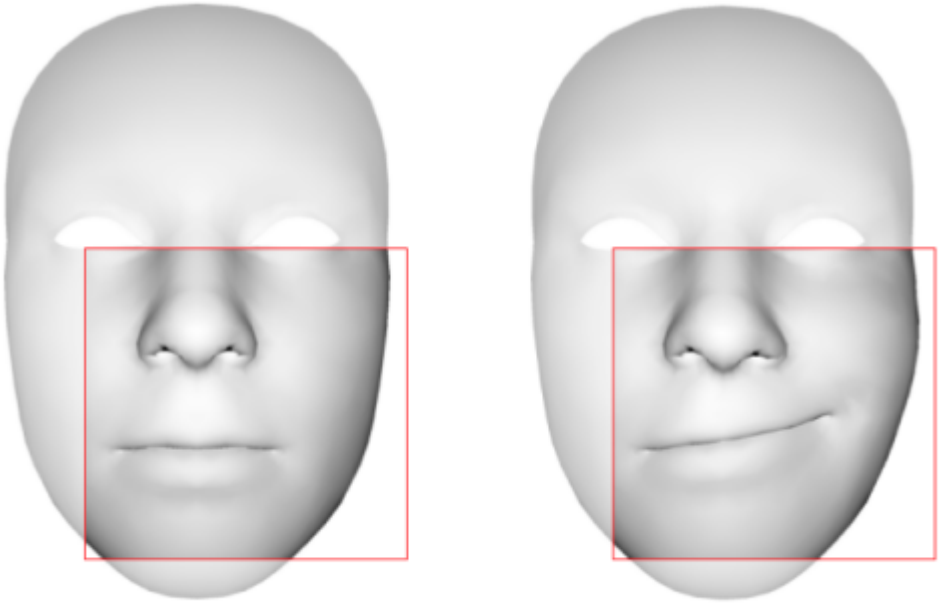
22.

两个嘴唇向左移动。



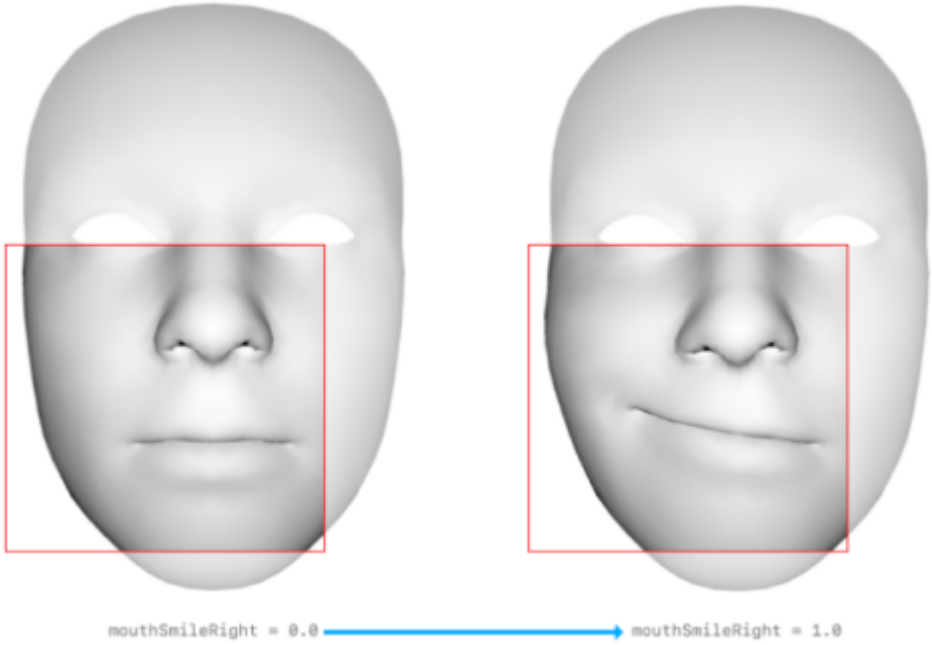
23. mouthRight = 0.0 → mouthRight = 1.0

两个嘴唇向右移动。



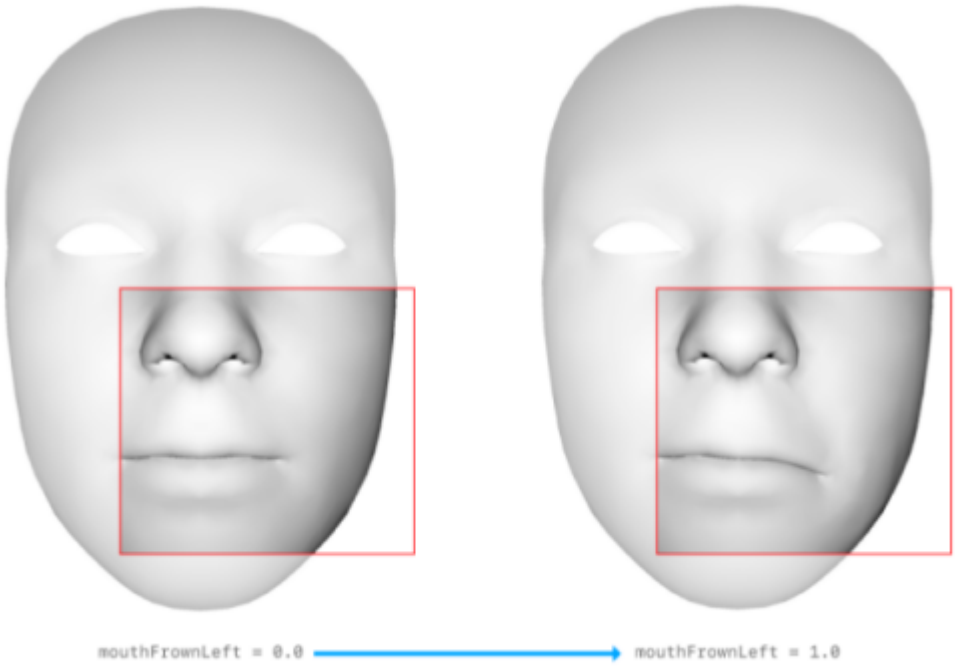
24. mouthSmileLeft = 0.0 → mouthSmileLeft = 1.0

嘴左角向上运动。（左边微笑）



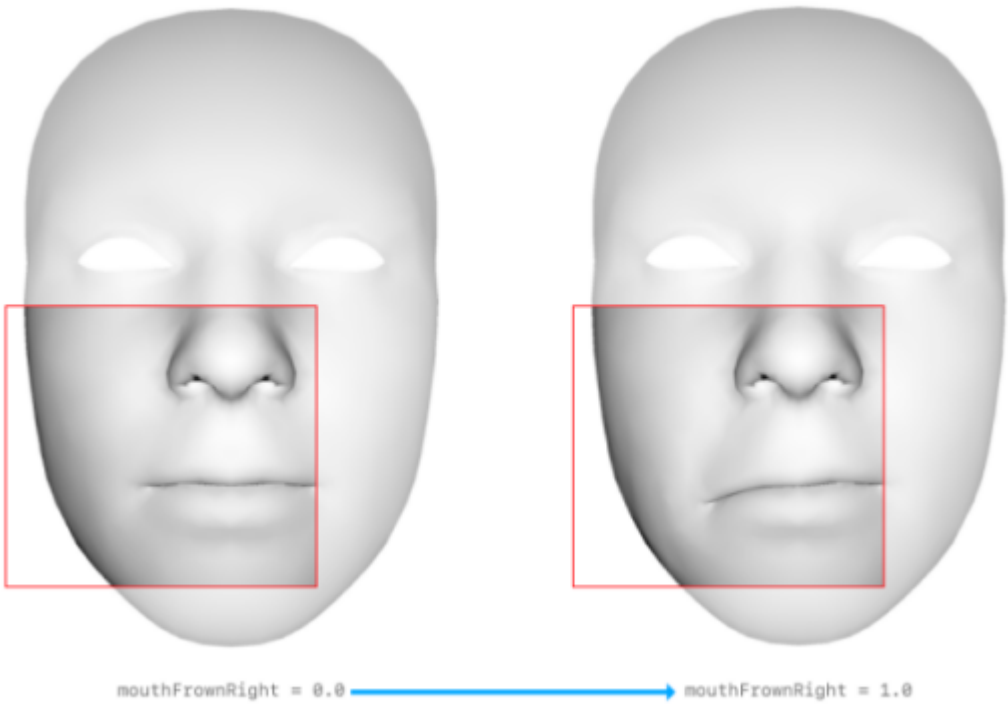
25.

嘴右角向上运动。（右边微笑）



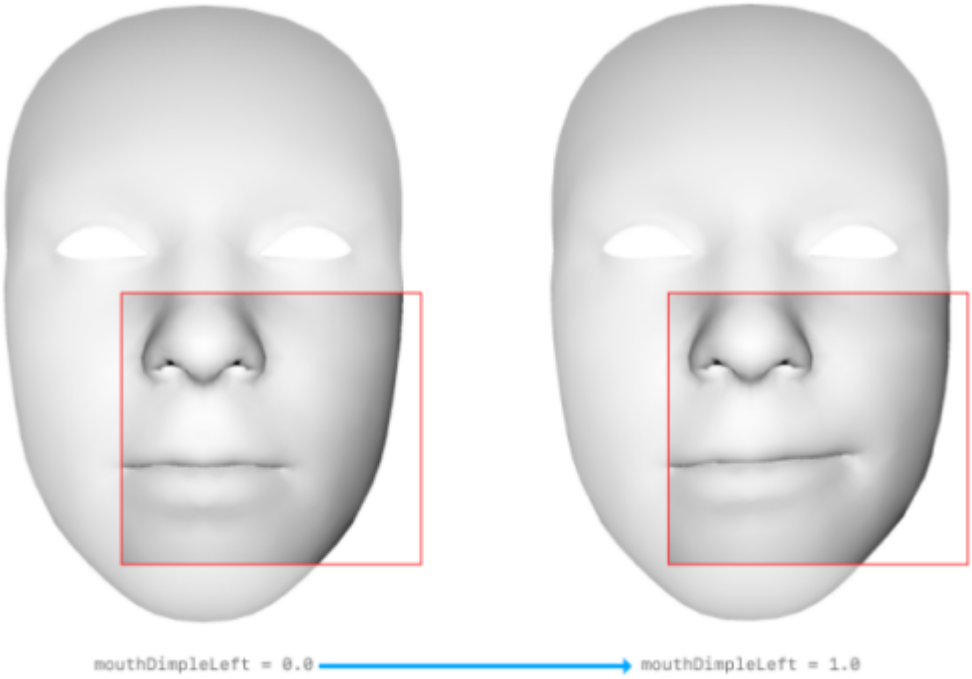
26.

嘴左角向下运动。



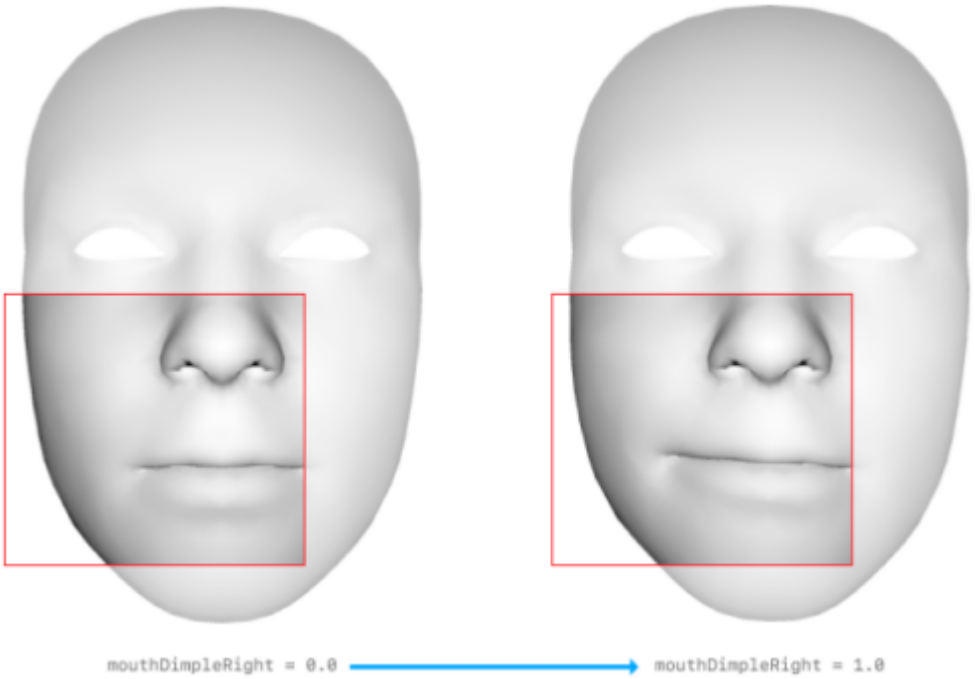
27.

嘴右角向下运动。



28.

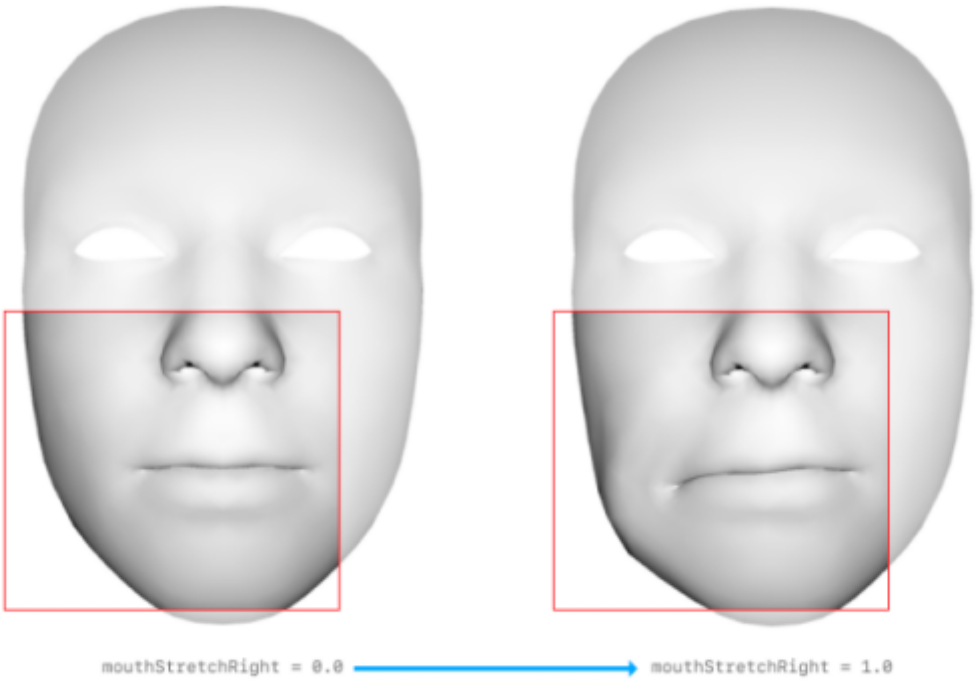
嘴左角向外运动（同时适当向后）。



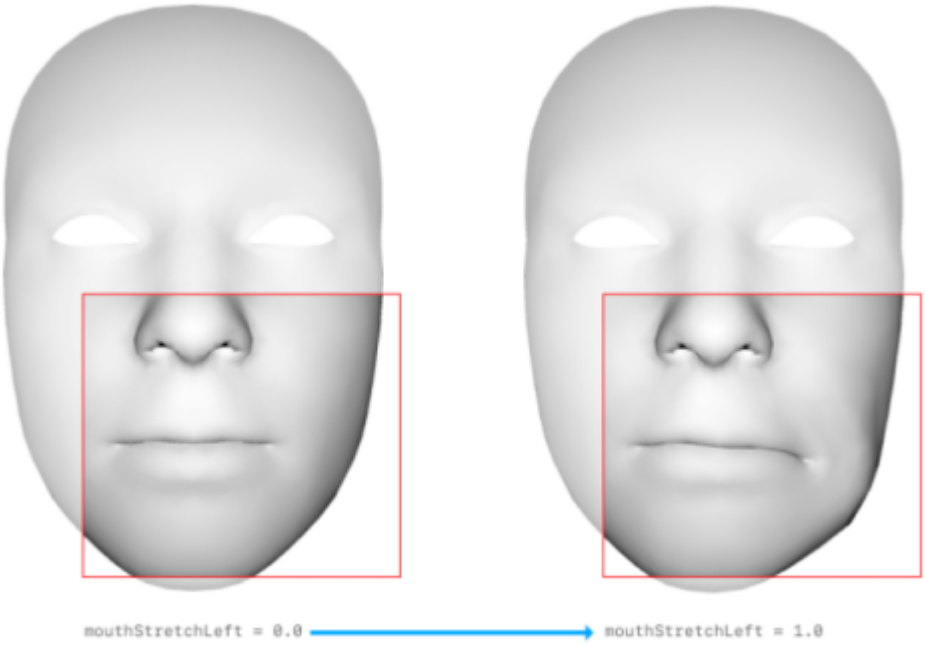
29.

嘴右角向外运动（同时适当向后）。

30.



右嘴角向右移动。

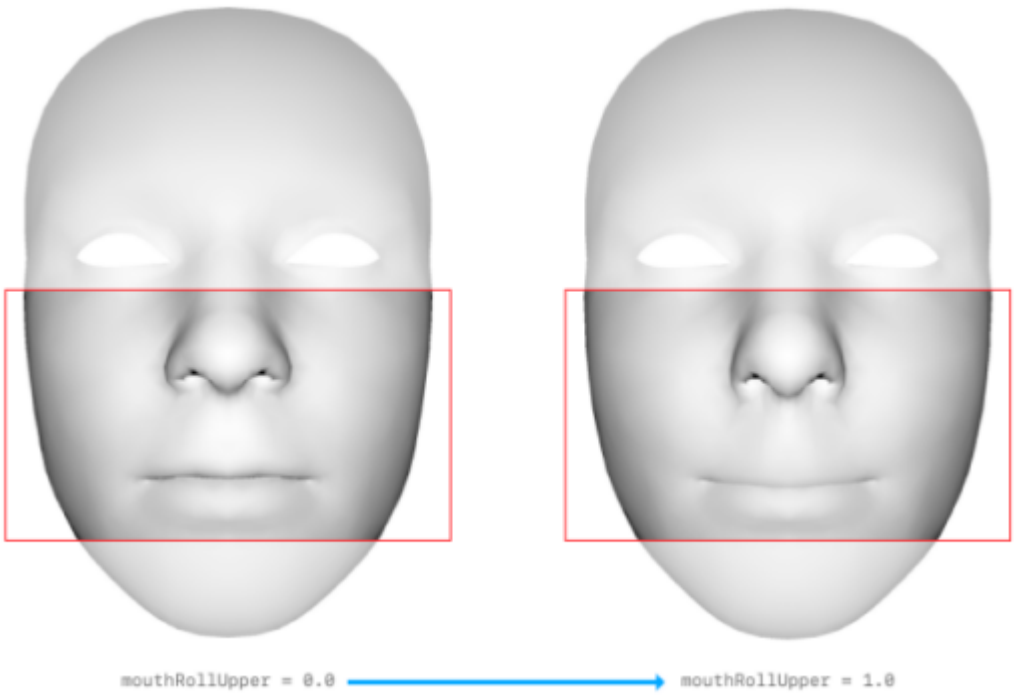


31.

左嘴角向左移动。

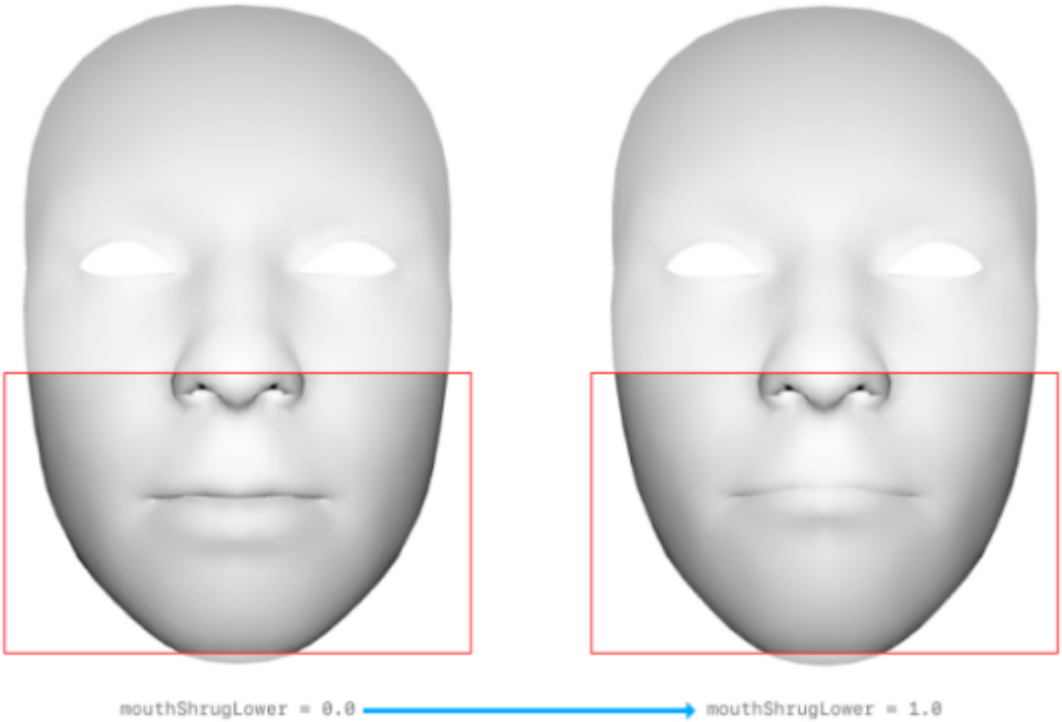
32.

下嘴唇向内侧移动。（内翻）



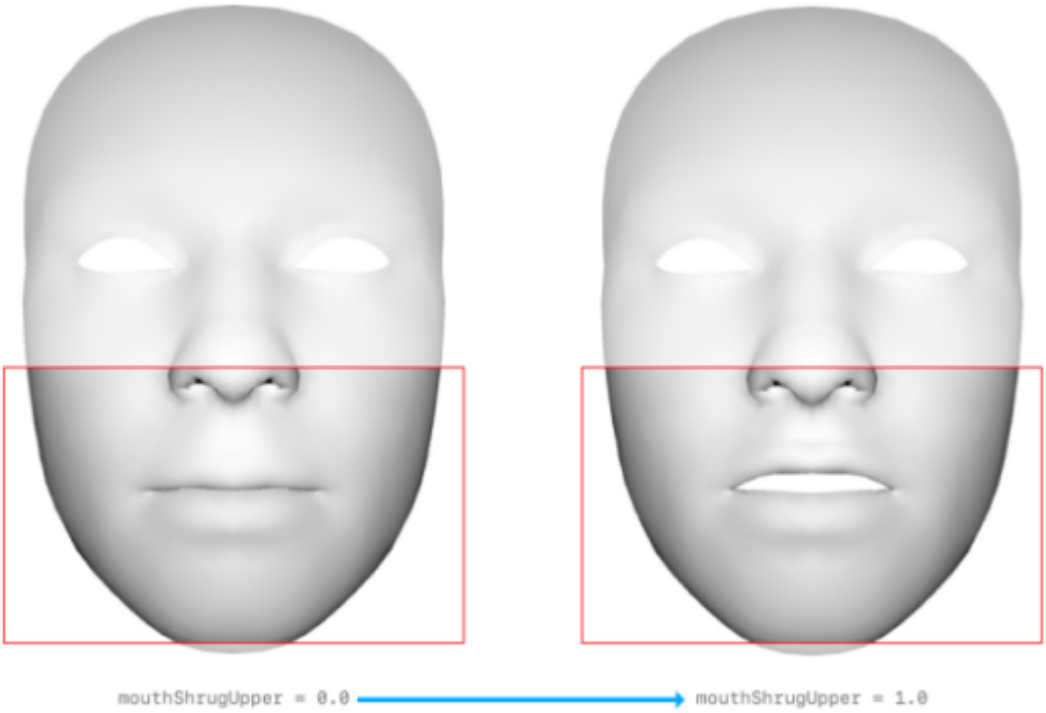
33.

上嘴唇向内侧移动。



34.

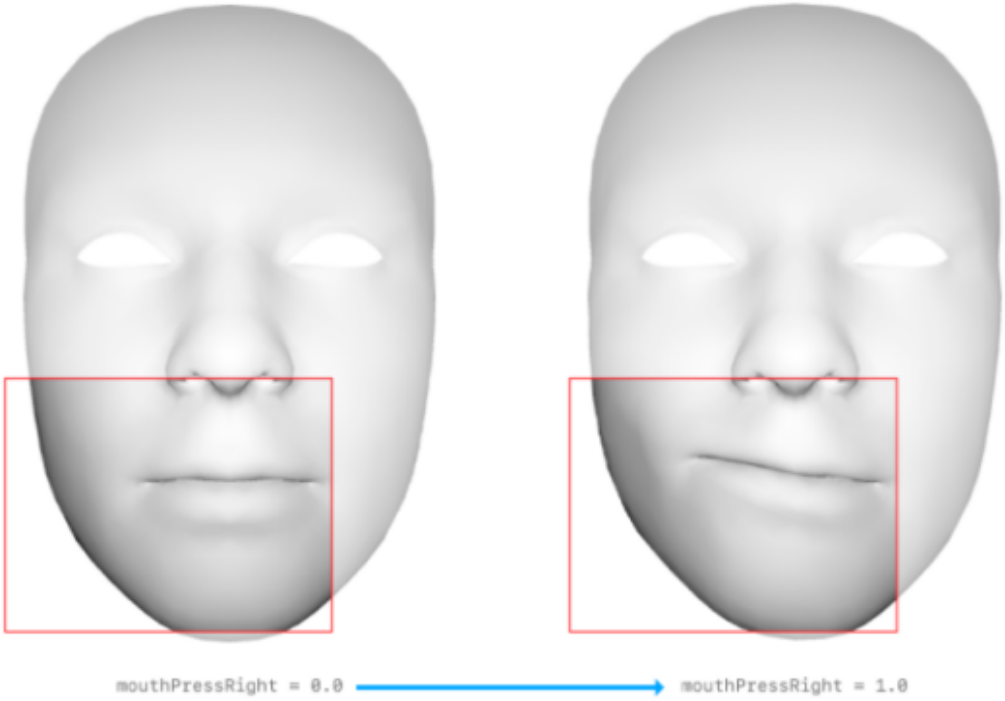
下嘴唇向外，并向上运动。（挤压上唇）



35.

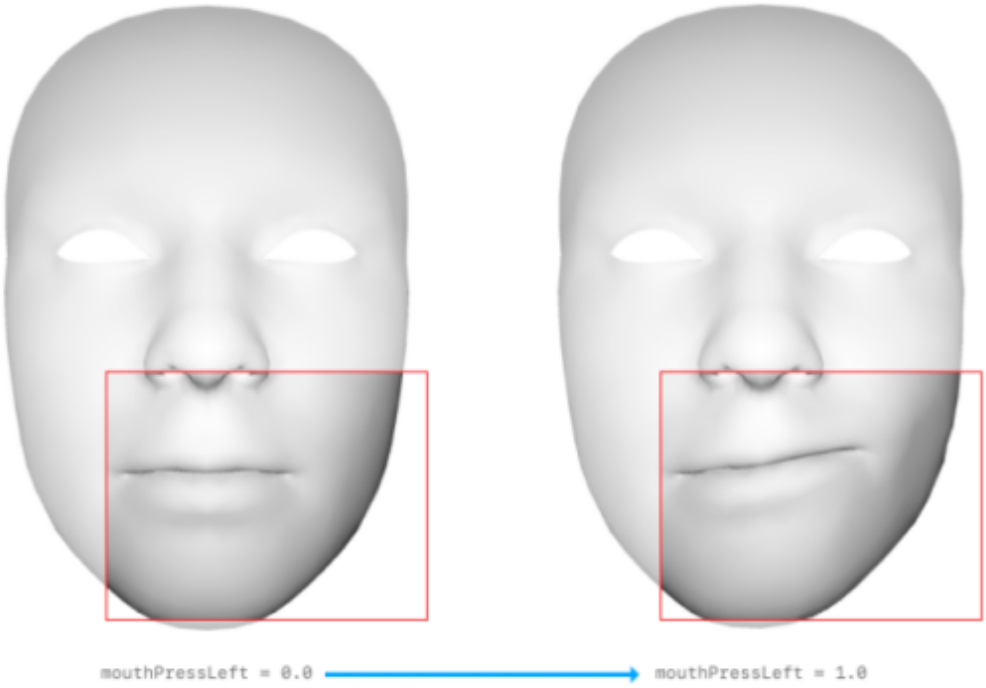
上嘴唇向上运动。（可适当外翻）

36.



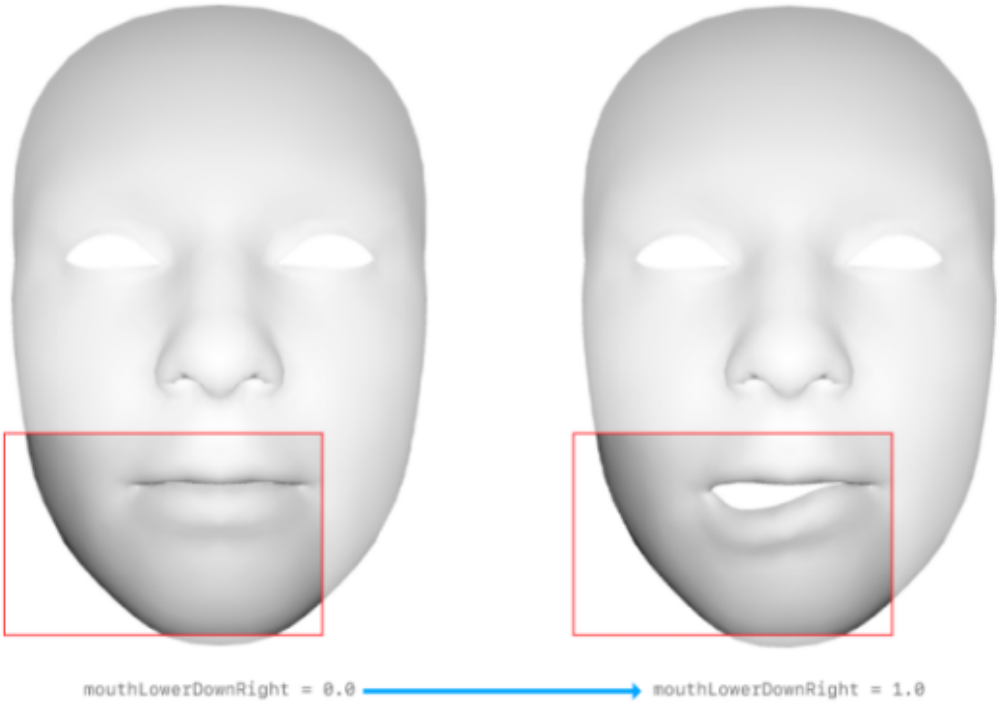
右侧下唇向上压缩。

37.



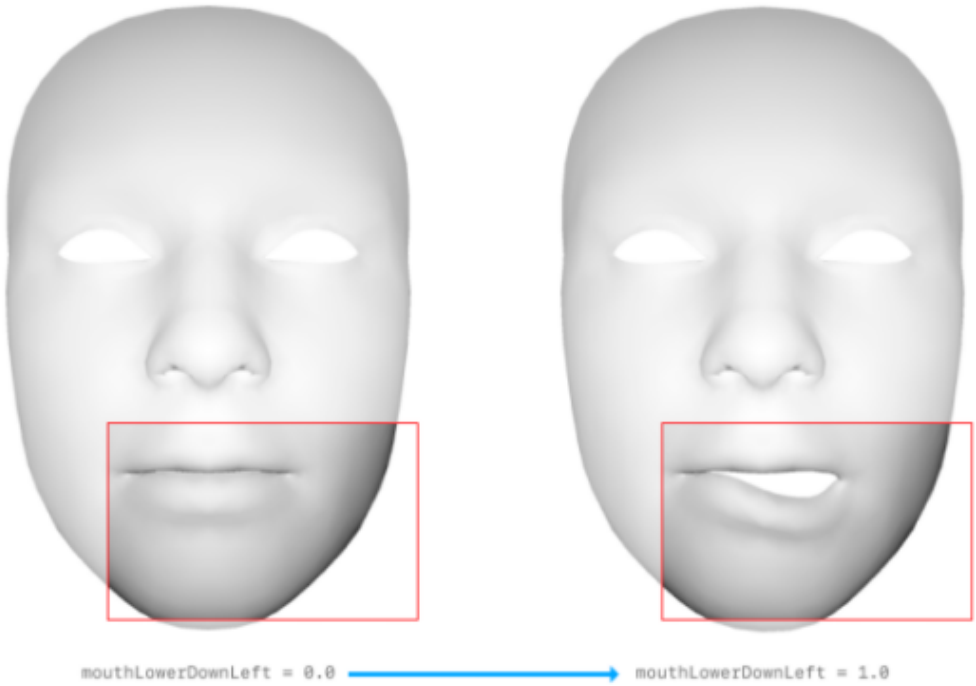
左侧下唇向上压缩。

38.



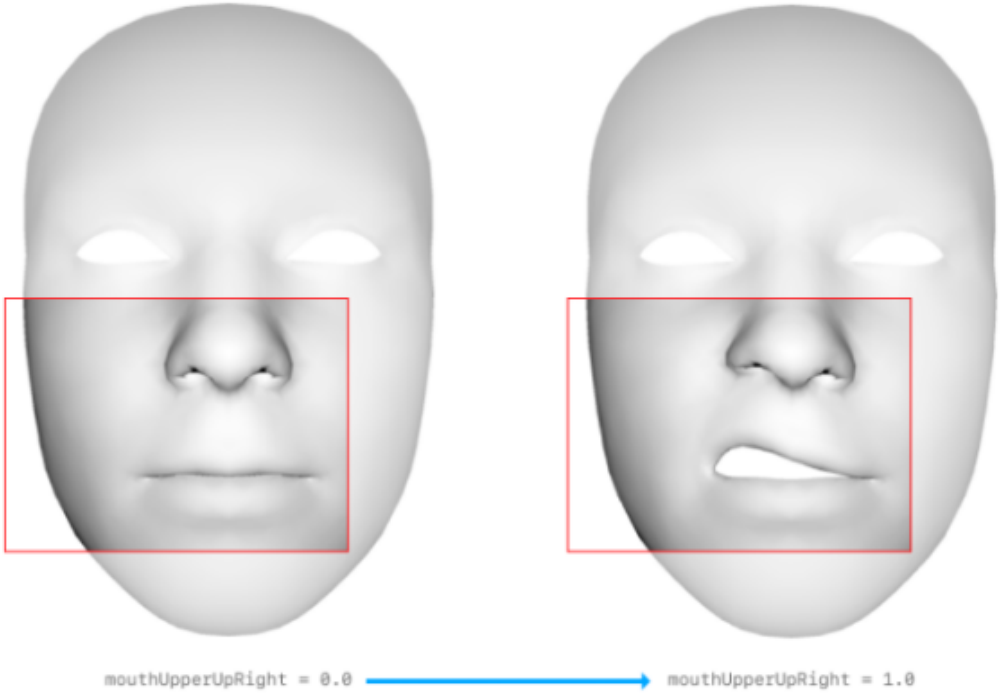
右侧下唇向下运动。

39.



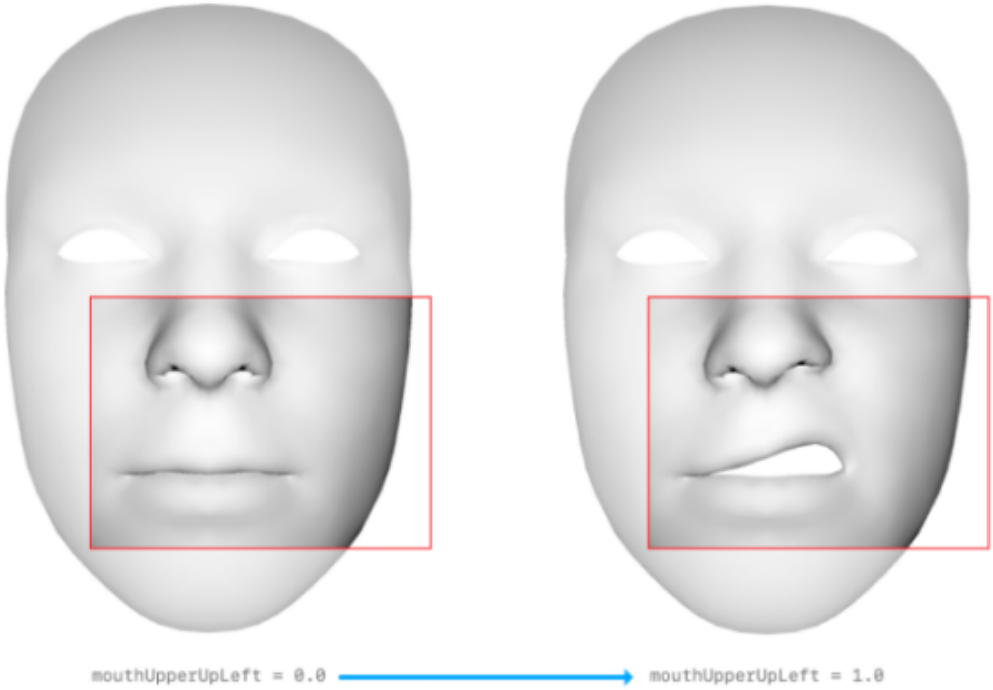
左侧下唇向下运动。

40.

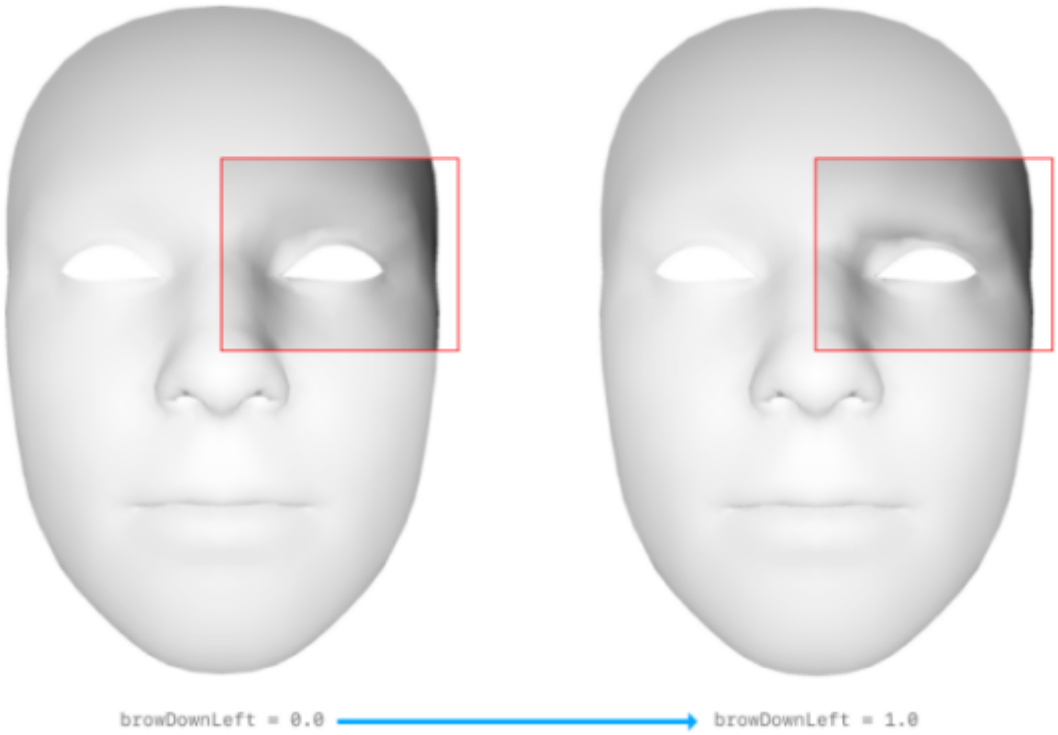


右侧上唇向上运动。

41.

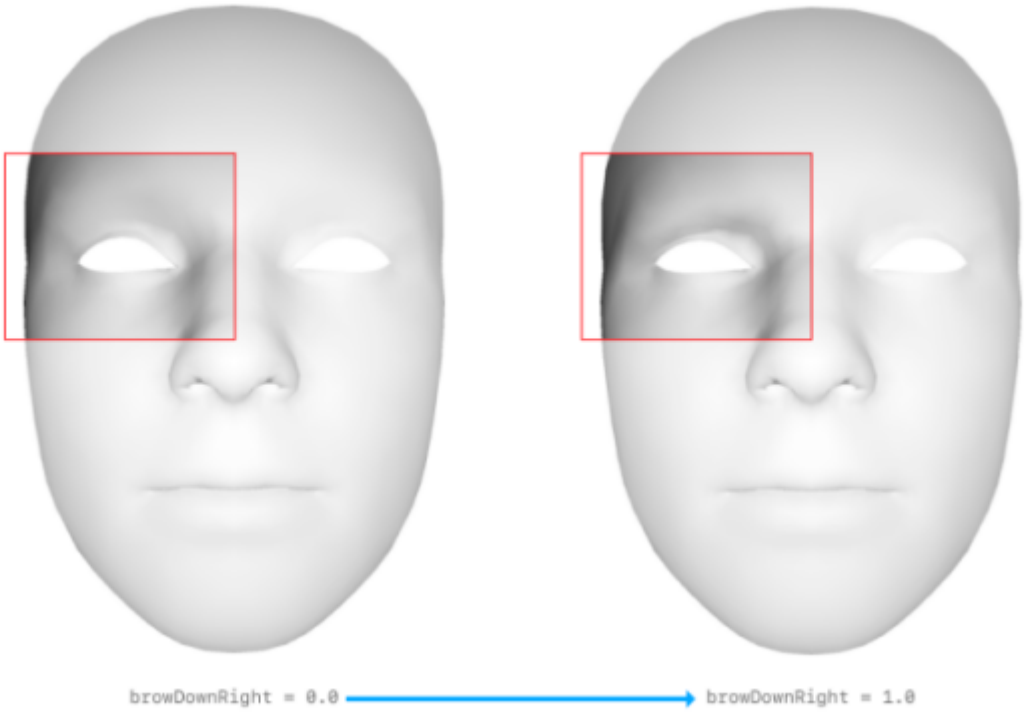


左侧上唇向上运动。



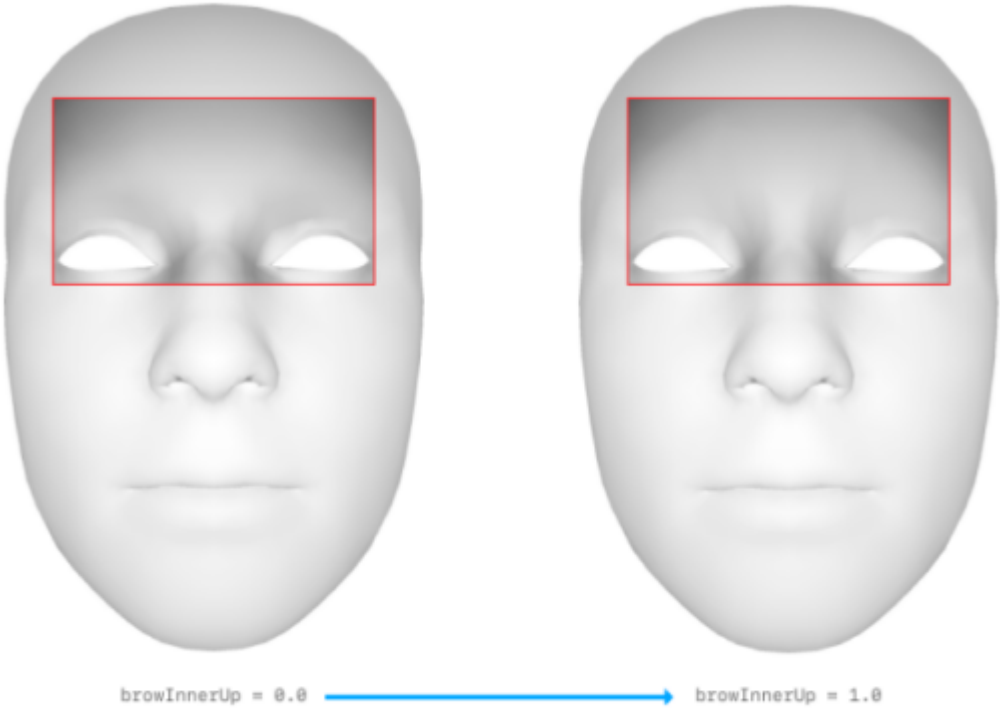
42.

左眉毛向下运动。



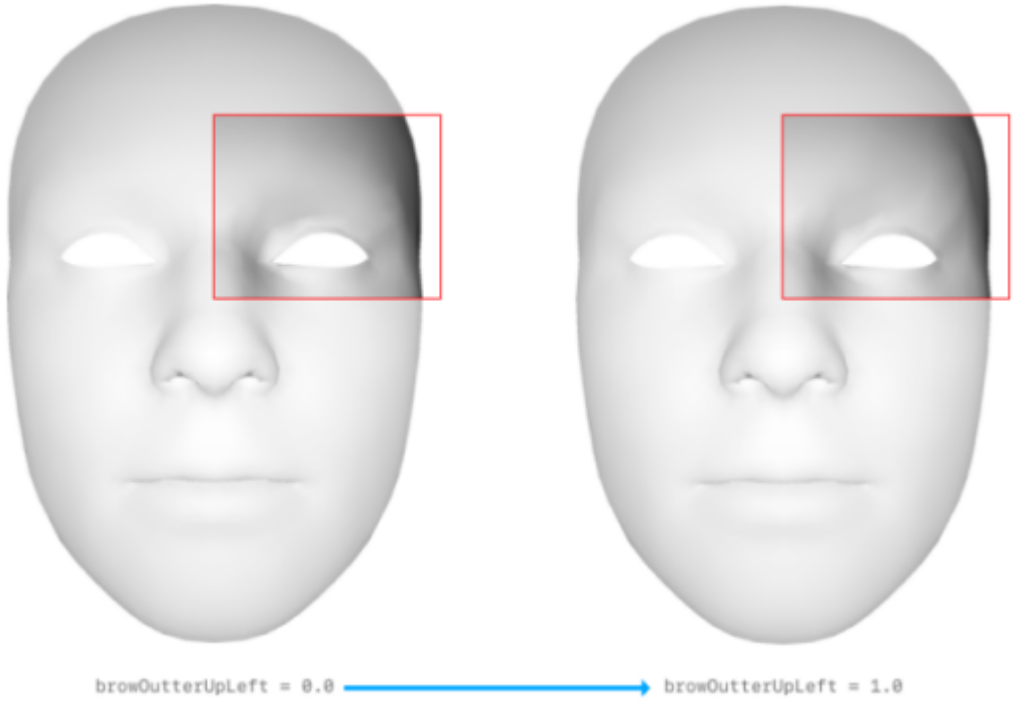
43.

右眉毛向下运动。



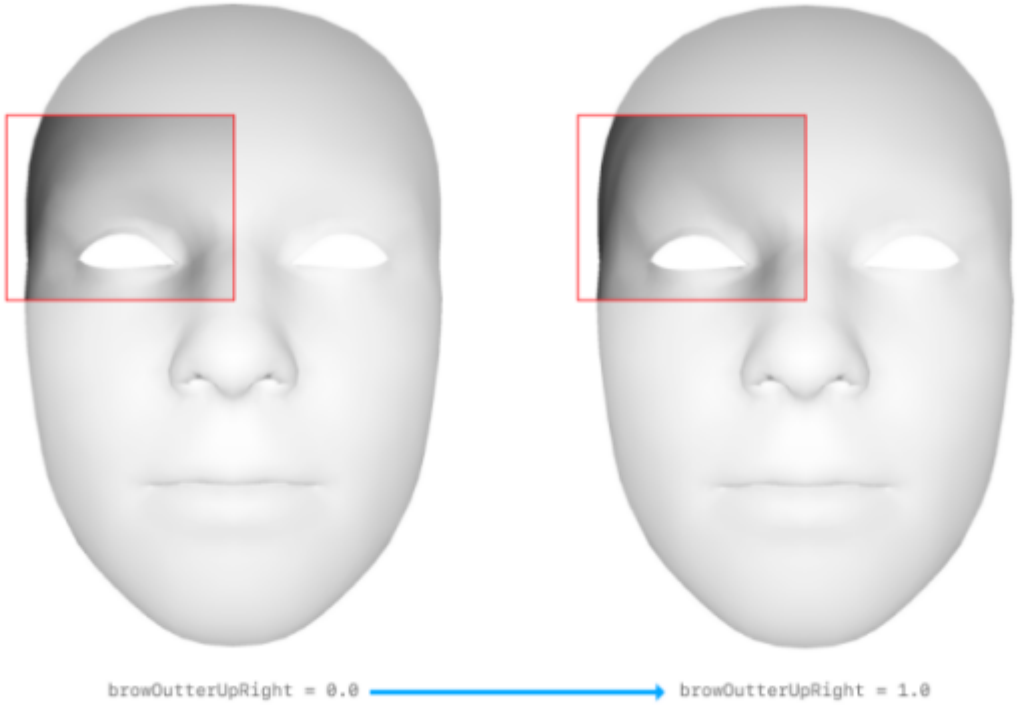
44.

两个眉毛内部向上运动。



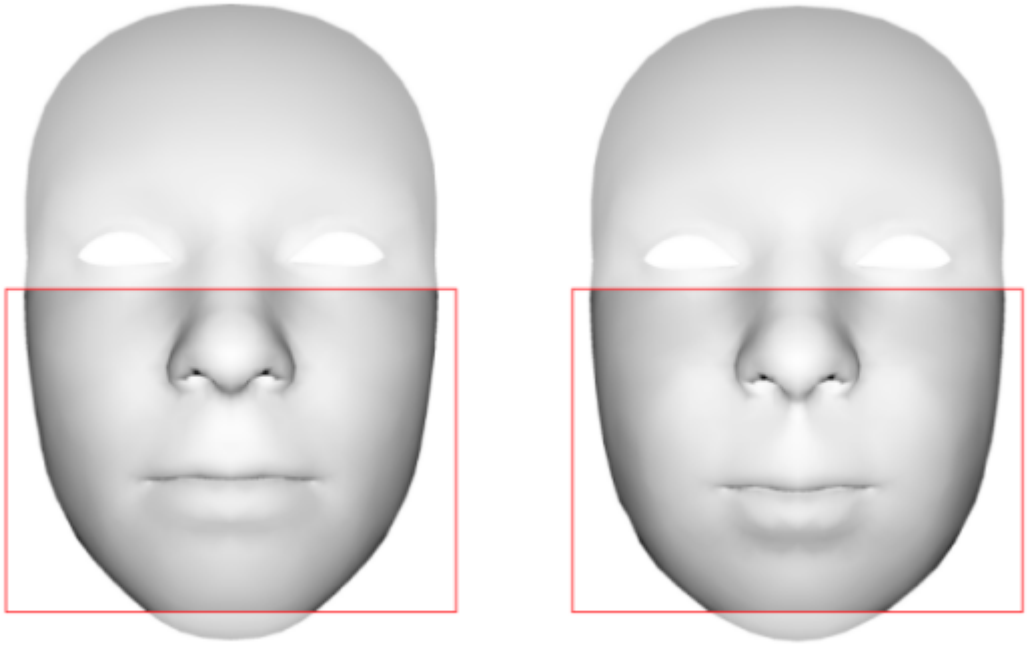
45.

左眉毛外部向上运动。



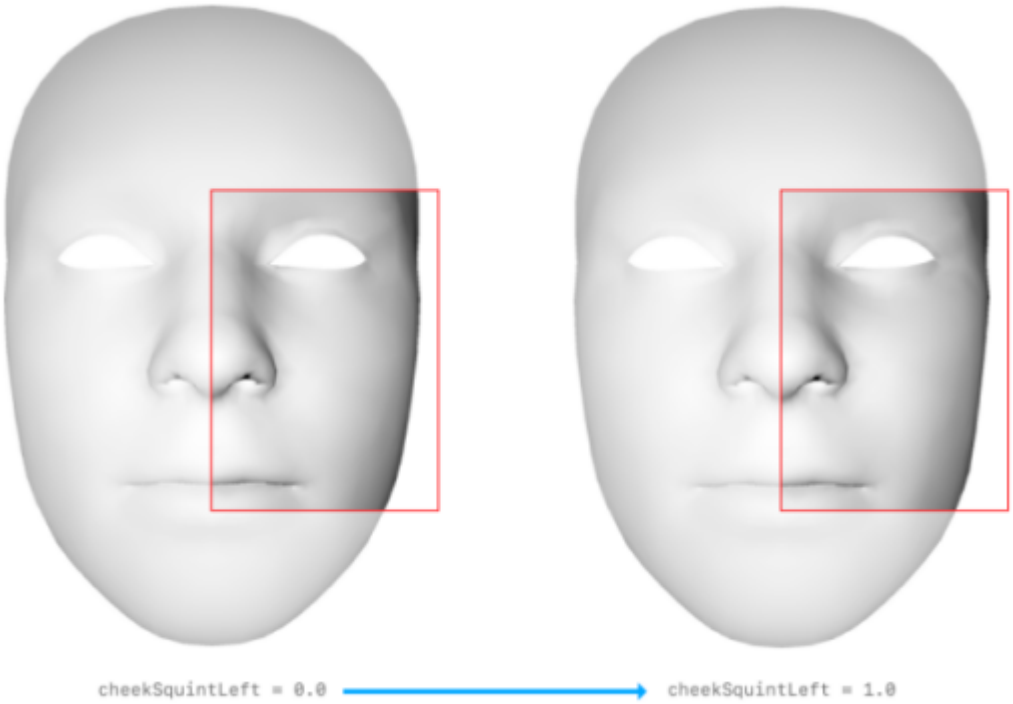
46.

右眉毛外部向上运动。



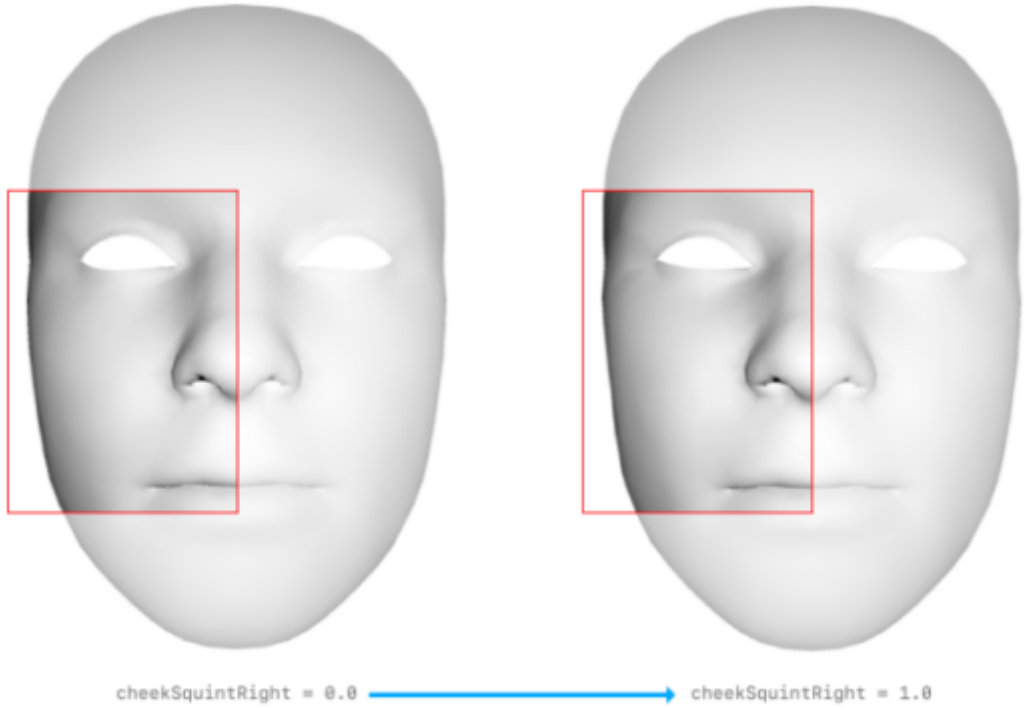
47.

两个脸颊向外运动。（鼓腮）



48.

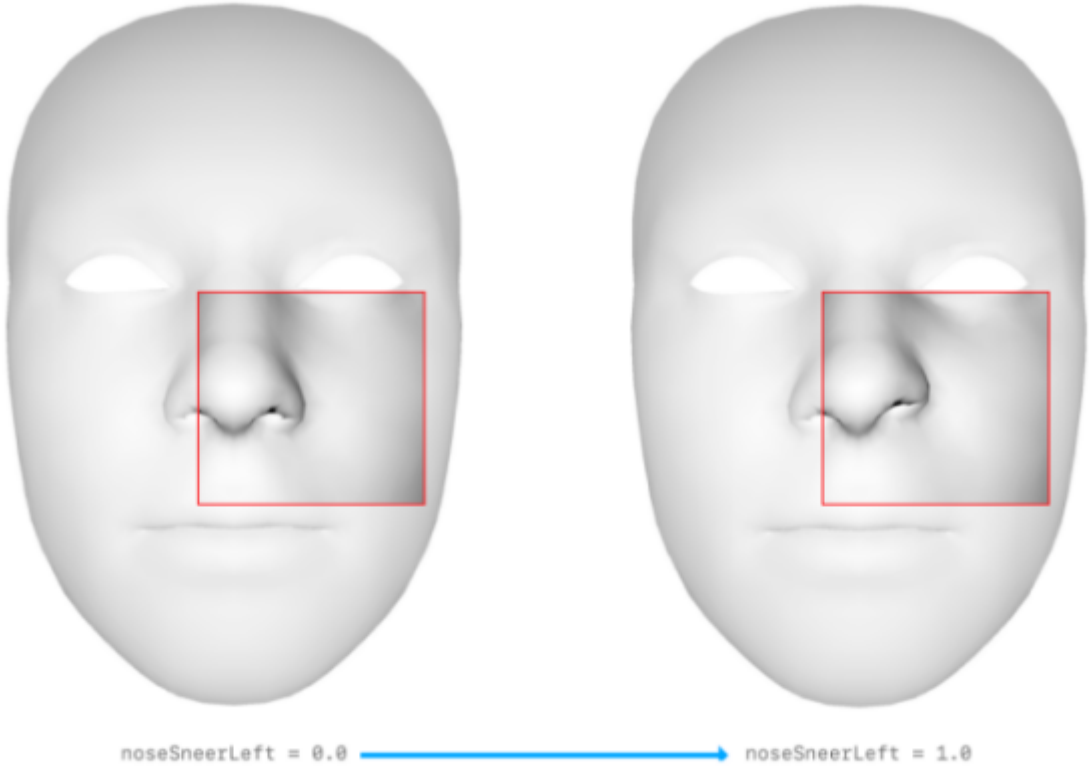
脸颊在左眼周围和下方向上移动。



49.

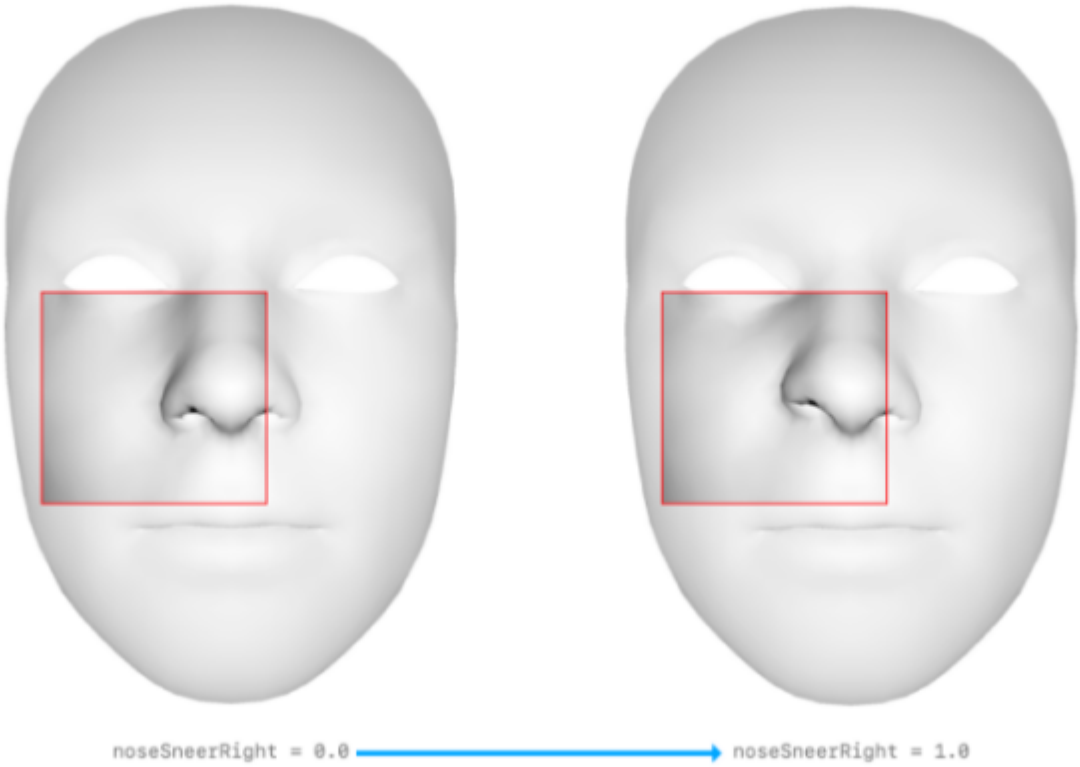
脸颊在右眼周围和下方向上移动。

50.



鼻孔左侧鼻子抬高。

51.



鼻孔右侧鼻子抬高。