

本文旨在指导用户制作可用于
Facerig 的 3D 角色模型，文档版本
号为 1.0

Facerig

模型与材质

文档

制作自定义 3D 角色的规
约

不吃鱼的喵酱

目录

| | |
|--------------------------------|----|
| 1. 提示..... | 1 |
| 2. 新功能与变化..... | 1 |
| 3. 几何信息..... | 2 |
| 3.1. 几何命名..... | 2 |
| 3.2. 默认姿势..... | 3 |
| 3.3. 几何规约..... | 3 |
| 3.4. UV 展开..... | 3 |
| 3.5. 变形 (deformation) | 4 |
| 3.6. 毛发..... | 4 |
| 4. shader (及材质) 命名..... | 5 |
| 4.1. shader 类型..... | 6 |
| 4.1.1. sht_furnormals | 6 |
| 4.1.2. sht_furnonormals | 6 |
| 4.1.3. sht_physfurnormals..... | 6 |
| 4.1.4. sht_metalcloth..... | 6 |
| 4.1.5. sht_eye | 7 |
| 4.1.6. sht_teethblendn | 7 |
| 4.1.7. sht_hair..... | 8 |
| 4.1.8. sht_skinblendn | 8 |
| 4.1.9. sht_skinnoan | 9 |
| 4.1.10. sht_prelit..... | 9 |
| 5. 纹理贴图..... | 9 |
| 5.1. 材质类型参考: | 10 |
| 5.1.1. 漫反射贴图: d..... | 10 |
| 5.1.2. 法线贴图: b..... | 11 |
| 5.1.3. 镜面反射贴图: s..... | 11 |
| 5.1.4. 毛发遮罩贴图: m | 12 |
| 5.1.5. 环境光遮蔽贴图: ao..... | 12 |
| 5.1.6. 颜色遮罩纹理: cm..... | 13 |
| 5.2. 为角色和配件制作的额外皮肤..... | 13 |
| 6. 骨骼..... | 13 |
| 6.1. 伪物理驱动的骨骼..... | 14 |
| 6.2. 动画与姿势 (pose) | 15 |

| | |
|---------------------------------|----|
| 7. 角色配置文件..... | 22 |
| 8. 全身动画制作规范（仅限 Studio 版本） | 23 |
| 9. 参考骨骼树..... | 23 |

1. 提示

请牢记，Facerig 的数据导入脚本需要花费一段时间（通常是 1-2 分钟）来运行，具体时间取决于你的模型与贴图的大小以及你的计算机性能。目前该脚本无法处理含有空格的路径，因此我们强烈建议你将数据存放在不包含空格的路径下。

2. 新功能与变化

新版的导入脚本与旧版的不同之处在于它们**处理动画的方式**。旧版的脚本会对用户动画进行**预处理**以使它们适用于 Facerig。预处理步骤**删去了一些动画**，只导入了那些适用于**特定骨骼**的动画。然而，这种处理依赖于特定的骨骼名称与特殊的动画关系，因此将来无法被简单地更新。（译者注：另一份文档也提及了这一点，看来官方对这里的设计相当不满意。）

新版本的导入脚本，尽管也需要用户制作相应的动画，但是允许用户**自由组织骨骼树**，同时也允许我们在**未来动态添加新的特性**。现在，依赖关系取决于动画的名称、放置于子文件夹的位置以及每个动画中对于每个骨骼的针对性设置。（译者注：这段不好翻译，建议参考原文。）

本文档适用于 Facerig 的所有版本（Steam Facerig Classic/Pro/Studio 或者从官网购买的独立版本），但它们有不同的默认路径，因此后文中我们将使用 [FacerigInstallPath] 这个标记来代指 Facerig/Facerig Studio 的安装路径。默认情况下，安装路径应该是下列中的某一个：

- C:\Program Files (x86)\Steam\SteamApps\Common\FaceRig\
- <Steam 库文件夹>\Steam\SteamApps\Common\FaceRig\
- C:\Program Files (x86)\Steam\SteamApps\Common\FaceRig Studio Personal\
- <Steam 库文件夹>\Steam\SteamApps\Common\FaceRig Studio Personal\
- C:\Program Files (x86)\Holotech\FaceRig Studio\

旧版本所要求的数据主要是 **COLLADA** 和 **Targa** 文件。新版本也需求它们，但是组织的形式有所不同。与之前相同，保存骨骼与几何信息与材质的文件必须放在同一个文件夹下，用户可以通过一个浏览对话框来选择这个文件夹。然而动画要放在子文件夹里。这里有一个例子供你参考。



关于动画的详细描述请参考文档后面的章节。

在 COLLADA 文件附近允许用户自行添加一个 **txt** 文档。如果用户设置了这个文档，那么脚本会自动将其转换成配置文件，用来设定角色的一些参数。配置文件会被自动放置在 “[FaceRigInstallPath]\Mod\VP\PC_CustomData\Objects\[AvatarName]” 路径下，你可以直接对其内容进行更改。它的后缀名是 **.cfg**，并且有一个 “cc” (CustomConfiguration) 前缀。你可以在 “角色配置文件” 那一章找到详细信息。

3. 几何信息

几何信息应该是模型在默认姿势下导出的 COLLADA 文件。(译者注：原文中并没有提及是 T-pose 还是 A-pose，或者对这个姿势有任何描述。)

3.1. 几何命名

包含几何信息的文件必须被命名为 名字+“Geometry” 的形式，其中名字就是你的角色名，“Geometry” 代表此文件包含几何信息而非动画信息。举例：如果角色名是 “Fluffo”，那么几何文件应该命名为 “**FluffoGeometry**”。

如果这个模型是一个已存在角色的 **DLC** (译者注：原文用的是 **prop**，根据意思翻译为 **DLC**)，那么几何文件应该被命名为：角色名+ “_prop_” +DLC 名+ “Geometry”。举例：如果角色 Fluffo 有一个 DLC 名为 “vikingHelmet”，那么几何文件应该被命名为 “**Fluffo_prop_vikingHelmetGeometry**”。

如果你想制作一个**泛用型 DLC** (译者注：原文是 **generic prop**，这里依旧是意译。) 以适配于所有角色，那么这里的名称应该遵循 “_prop_” +DLC 名+ “Geometry” 的规则 (比如它可以是 “_prop_romanHelemetGeometry”)。

如果你的文件夹中没有包含任何的几何信息文件，那么导入脚本就会**放弃这次预处理**，并且此模型**不会被**导入 Facerig。

3.2. 默认姿势

在默认姿势下，停用蒙皮（deactivating skinning）不应该使得模型外观产生任何变化，否则这个姿势就不是我们要求的默认姿势。（译者注：原文的意思可能是指这个姿势是静止不动的，或者是要求同一个角色的不同皮肤之间要保持默认姿势一致，我不确定该如何翻译。）

3.3. 几何规约

几何文件应该遵守以下几条规矩：

- 所有的几何体都必须是多边形拼成的。（即不能是函数绘制，NURBS 曲面，点云，体素，或者其他任何形式。）（译者注：其实就是要求使用面片。）
- 离散的物体（即出现不相连的面）是允许的，不闭合的表面（open surface）也是允许的。
- 模型面数推荐在 50000 个三角形以内，越少越好。虽然在高性能计算机上可以使用比这个更精细的模型，但是大部分情况下高模都会严重影响帧数。
- 至于缩放模型，角色高度应该在 2 单位左右。这个单位是什么并不重要，英寸，厘米，泛指单位皆可。如果导入以后你发现你的模型太小了（你可以通过观察影子来进行判断。这种情况下影子会有一个突兀的边缘（sudden pass in steps）（译者注：这个描述其实挺模糊的。)), 你可以在运行时调整它。如果要这么做，你需要创建一个名为“cc_[AvatarName].cfg”的配置文件，并把它放在 “[FaceRigInstallPath]\Mod\VP\PC_CustomData\Objects\[AvatarName]” 目录下，然后向其中添加 “set_avatar_scale <缩放倍数>” 这一行指令。
- 多边形的变换必须被重置为初始状态。也就是说，多边形的坐标系必须与世界坐标系一致，并且位于原点。缩放应该保持为初始值（在不同建模软件下表现为 1, 1, 1 或者 100, 100, 100）。
- 所有的面都必须有蒙皮（贴图？）。
- 模型至多只能使用 10 种材质。

3.4. UV 展开

- 模型的 UV 展开应该是正规化的（在[0, 1]空间内）。尽管 UV 展开允许出现部分重叠的情况，但是我们推荐你使用不重叠的 UV 展开，因为有些 Shader 要求这种形式的展开，比如次表面散射。

- 所有的多边形都必须分配 UV 展开坐标。

3.5. 变形 (deformation)

所有的变形都由骨骼变形（蒙皮）完成。**Blendshape**（译者注：应该指的是顶点直接变形完成的动画，而不是通过骨骼。）或者置换（译者注：这里讨论的置换可能包括使用置换贴图。）或者其他任何形式的变形都不被支持。

声明：

- 在 DX9 环境下，渲染几何体的一部分的一个 **shader**（整个几何体或许使用了很多个 **shader**）所能使用的最大骨骼数量不应该超过 52。如果超过了这个数量，可能会带来性能的骤降，因为不能使用硬件加速了。
- DX11 环境下单一 **shader** 的骨骼数量限制是 128。如果超过了这个数量，这个模型就不能被处理了。然而，整个模型所能使用的骨骼总数远超 128，只要满足每一个单一的 **shader** 所处理的骨骼数量低于 128。
- 每个顶点只能受到至多 4 个骨骼的影响。任何超过这个限制的骨骼影响都将被忽略。
- 蒙皮的补间必须是线性的，不能是四元组或者其他任何形式。

3.6. 毛发

毛发会被几何信息与纹理贴图同时控制。这一节会解释几何信息控制的那部分。（译者注：纹理控制的那部分应该是毛发的透光性或者看上去的密实程度。）

几何体网格可以借由**顶点色**来传递毛发控制信息。这方面看上去就像法线贴图一样，由颜色通道决定**切线空间下的方向角**，alpha 通道指定**局部长度比 (local length ratio)**。

毛发的**绝对长度**是由 **shader** 决定的。根据这个长度，顶点的 alpha 通道可以指定一个比最大长度更短的毛发长度。值为 255 将使得毛发达到最大程度，而 128 将达到一半长度，当然，0 就是没有毛发。

由于毛发方向是在切线空间中表示的，并且 **Facerig** 使用用户自定义的切线空间，因此在导入过程中**真实颜色会被写入顶点信息**（译者注：原文如此，并不能理解想表达什么。）。为了实现这一点，导入脚本需要一对节点分别表示毛发根部和毛发末梢。这些节点应该被分别命名为“**hairTip**”和“**hairRoot**”，后面跟上节点的索引，比如“**hairTip_01**”和“**hairRoot_01**”。末梢节点**必须被根部节点**链接。末梢节点和根部节点的位置将会被用来计算切线方向，所以务必确保这些位置被正确设置了。根部节点应该尽可能地贴近模型表面。

每一个包含有毛发的面必须被命名为“BlendedMesh”后面加上下标，比如“BlendedMesh_01”。对于刚体网格这种规则也适用。这种名字是用来标识那些顶点色中应该包含有毛发信息的表面的。

在毛发信息写入顶点色后，毛发节点会从模型中移除，这样就不会和骨骼文件堆在一起显得乱七八糟了。

4. shader（及材质）命名

（译者注：shader 非要翻译成着色器的话写起来很不习惯，所以这里保留了原文，反正大家都看得懂。）

材质的命名应该在你所使用的 3D 建模软件（比如 Maya, Cinema 4D, Modo, Blender, 3Ds Max, 水杉, 等）中完成。你需要找到一种至少包含 4 张纹理（比如环境光遮蔽贴图，法线贴图，漫反射贴图，镜面反射贴图）的材质，其余材质（.tga 文件）则需手动添加。

材质名称应该按照以下格式：模型名称+_shader 类型+shader 名称。比如有个角色名字是“Fluffo”，有个材质名称是“Face”，那么 shader 文件的名称就应该是“Fluffo_sht_furnormals_face”。其中“Fluffo”是角色名，“_sht_furnormals”是 shader 的类型名，“face”是 shader 名。

如果你想要在你的 shader 中渲染透明面，那么它的名字应该包含：

- “blend1”表示 alpha 混合
- “blend2”表示 alpha 测试

比如我们现在想做一个透明的围巾，那么 shader 应该起名为“Fluffo_sht_metalcloth_bandana_blend2”。

在 alpha 测试中，你可以使用 **Alpha Reference** 功能（值介于 0 和 1）来设置 alpha 值。比如你将它设为 0.2，这意味着最终输出的图像中 alpha 小于 0.2 的像素将会被丢弃，而大于 0.2 的将会被保留。默认值是 0.5。你可以在模型导入工具中设置 **Alpha Reference** 的值。

对于面剔除也有选项支持（双面剔除或者任一面剔除）。默认值是“逆时针剔除”，这意味着如果观察者是从反面观察这个面的，那么这个面会被剔除。如果你不想使用面剔除功能，那么你的 shader 的名字应该包含“_cull_none_”。比如要制作一个不使用面剔除的围巾，你应该将 shader 命名为“Fluffo_sht_metalcloth_bandana_cull_none”。（译者注：默认的面剔除即背向面剔除，可以节省一部分渲染开销，但是有时候也许会出现错误，比如背向面出现在透明面之后有可能会出现渲染错误。当然，有些情况下也会使用前向面剔除。面

剔除一般只对凸多边形有效。)

所有的 `shader` 名称都必须是小写的，即使角色名的首字母是大写的。这意味着，对于角色“Fluffo”，在 `shader` 名中也只能写作“fluffo”。(译者注：显然，刚才的例子全都没有遵守这一条。)

至于如何制作额外的皮肤(译者注：这里翻译存疑。)，请参考“纹理贴图”那一节。

4.1. `shader` 类型

4.1.1. `sht_furnormals`

这种类型代表它会被用于有毛发的表面。它使用两张法线贴图。其中一张作为基底保存了所有的用户制作的信息，另一张则在特定情况下可变，比如在皱眉的表情中。这种材质总计使用 7 张纹理以及顶点色的全部 4 个通道。

顶点色的颜色通道被用来指定毛发生长的方向(与法线贴图指示方向的原理类似)，而 `alpha` 通道指定毛发长度(0-255 对应 0 到最长毛发长度)。这个正规化了的值是用来控制当前毛发长度与毛发绝对长度的比例的，而后者由 `shader` 指定。

这个 `shader` 使用的纹理：漫反射贴图(“`d`”)，法线贴图(“`b`”)，法线贴图 1(“`b1`”)，镜面反射贴图(“`s`”)，毛发遮罩(“`m`”)，环境光遮蔽贴图(“`ao`”)，着色遮罩(“`cm`”)。

4.1.2. `sht_furnonormals`

与 `sht_furnormals` 有着相同的特性，只是取消了动态法线特性，因此只使用一张法线贴图。

4.1.3. `sht_physfurnormals`

与 `sht_furnormals` 有着相同的特性，并且能指定一个基于物理的毛发运动。

4.1.4. `sht_metalcloth`

为了表现金属与布料而设计的 `shader`。这个 `shader` 使用的纹理包括漫反射贴图(“`d`”)，法线贴图(“`b`”)，法线贴图 1(“`b1`”)，法线贴图 2(“`b2`”)，法线贴

图 3 (“b3”), 法线贴图 4 (“b4”), 镜面反射贴图 (“s”), 镜面反射贴图 2 (“s2”), 环境光遮蔽贴图 (“ao”), 着色遮罩 (“cm”)。其中环境光遮蔽贴图的每个通道都与一张法线贴图相对应:

- 红通道: 法线贴图 1 的环境光遮蔽
- 绿通道: 法线贴图 2 的环境光遮蔽
- 蓝通道: 法线贴图 3 的环境光遮蔽
- Alpha 通道: 默认法线贴图的环境光遮蔽

4.1.5. sht_eye

这个 shader 使用的纹理: 漫反射贴图 (“d”), 法线贴图 (“b”), 法线贴图 1 (“b1”), 镜面反射贴图 (“s”), 虹膜纹理 (“ristext_d”) (“raotext_d”)。纹理应该是正方形的, 并且中间有瞳孔。

虹膜纹理 “ristext_d” 是用来表示虹膜属性的。alpha 通道控制当前像素与虹膜外表面的距离, 0 表示虹膜外表面, 255 表示虹膜中心。因此使用梯度下降的 alpha 值可以做出一个圆穹顶的形状。RGB 通道 (灰度值) 定义了受到瞳孔扩张影响的范围。0 代表不受影响, 255 代表最大影响。用户应该按照虹膜外部是 0, 中心是 255, 线性递减的方式来设置。梯度的变化会影响瞳孔扩张时的真实程度。(译者注: 大概是在说, 瞳孔扩张会有一个阈值, 高于这个阈值的会被认为是瞳孔, 因此数值要从内向外递减。)

虹膜纹理 “raotext_d” 指定了环境光反射、镜面反射遮罩以及着色环境光遮罩。(译者注: 大概是虹膜反光的相关参数。)

4.1.6. sht_teethblendn

口腔内使用的 shader。这个 shader 使用的纹理包括漫反射贴图 (“d”), 法线贴图 (“b”), 镜面反射贴图 (“s”), 环境光遮蔽贴图 (“ao”), 着色遮罩 (“cm”), 次表面散射贴图 (“sc”)。

镜面贴图的各通道使用方式如下:

- 红通道被用来设置光泽度, 粗糙表面接近 0, 光滑表面接近 255。根据经验可以为粗糙表面设置为 1-5, 为极其细微的反光设置为 5-12, 需要更亮的反光则逐级递增直至 255。(译者注: 光泽度贴图一般是粗糙度贴图的倒数, 实践中并没有哪个更好, 只有哪个用起来更习惯。)
- 绿通道用来设置嘴巴完全闭合时候的环境光遮蔽。
- 蓝通道被用来标识那些不受光照影响的区域。当这些区域处于阴影中时, 它

们会微微发光。0 代表完全不发光，255 代表完全发光。

- Alpha 通道表示镜面反射强度，0 代表无镜面反射，255 代表完全的反射。

4.1.7. sht_hair

这个 shader 使用的纹理：漫反射贴图（“d”），法线贴图（“b”），镜面反射贴图（“s”），环境光遮蔽贴图（“ao”）。其中镜面反射贴图的红通道是没用的，应该被默认地设置为 255。

环境光遮蔽贴图的各个通道使用方法如下：

- 红通道不使用，设置为默认值 255。
- 绿通道设置直接光照权重。0 表示无直接光照（只受环境光照和辐照度的影响），255 表示完全的直接光照权重。
- 蓝通道不使用，设置为默认值 255。
- Alpha 通道设置基础的 PPAO（译者注：原文没有注释，不知道是什么的缩写。）

4.1.8. sht_skinblendn

专门为皮肤区域设计的材质。这个 shader 使用的纹理包括漫反射贴图（“d”），法线贴图（“b”），法线贴图 1（“b1”），法线贴图 2（“b2”），法线贴图 3（“b3”），法线贴图 4（“b4”），镜面反射贴图（“s”），环境光遮蔽贴图（“ao”），着色遮罩（“cm”），次表面散射贴图（“sc”）。其中环境光遮蔽贴图的每个通道都与一张法线贴图相对应：

- 红通道：法线贴图 1 的环境光遮蔽
- 绿通道：法线贴图 2 的环境光遮蔽
- 蓝通道：法线贴图 3 的环境光遮蔽
- Alpha 通道：默认法线贴图的环境光遮蔽

镜面反射贴图的各通道数据使用如下：

- 红通道被用来设置光泽度，粗糙表面接近 0，光滑表面接近 255。根据经验可以为粗糙表面设置为 1-5，为极其细微的反光设置为 5-12，需要更亮的反光则逐级递增直至 255。
- 绿通道是 PPAO 的协力者，由法线贴图 4 使用。
- 蓝通道被用来标识那些不受光照影响的区域。当这些区域处于阴影中时，它们会微微发光。0 代表完全不发光，255 代表完全发光。

- Alpha 通道表示镜面反射强度，0 代表无镜面反射，255 代表完全的反射。

次表面散射贴图的各通道数据使用如下：

- 红通道规定次表面散射的半径，0 表示无次表面散射，255 表示最大散射半径。在贴图接缝或者体表饰物附近应该逐步降低至 0。
- 绿通道设置直接光照权重。0 表示无直接光照（只受环境光照和辐照度的影响），255 表示完全的直接光照权重。
- 进入次表面开始散射的光与被材质反射的光之间的比例。0 表示全都反射，255 表示全部进入次表面进行散射。对于皮肤的肉质部来说，一个好的默认值大约是 0.9 左右（229, 229, 229）。在贴图接缝或者体表饰物附近应该逐步降低至 0。
- Alpha 通道设置对象半透明度（比如耳朵、鼻孔）。0 表示完全不会，255 表示完全的半透明。（译者注：也可能是完全的透明。）

4.1.9. sht_skinnoan

与“sht_skinblendn”有相同的用法，除了可移动法线的特性，这将消耗一张额外的法线贴图。

4.1.10. sht_prelit

只使用一张贴图，以及顶点的颜色信息。这个 shader 可以在眼球与眼皮附近渲染一个小环，从而模拟出眼部阴影。使用漫反射贴图，要求使用 64 像素*64 像素分辨率的小图，Alpha 通道的值为 128 或 255。

顶点色的 Alpha 通道会被用来控制透明度，255 不透明 0 完全透明。RGB 通道用来控制整体颜色。

5. 纹理贴图

Shader 使用的贴图是 tga 格式的二进制文件（含 Alpha 通道）。贴图分辨率应该是 2 的整数次方（即 16*16, 32*32, 64*64, 128*128, 256*256, 512*512, 1024*1024 等）。

[AvatarName]+”_”+[ShaderName]+”_”+[TextureName].tga

其中，AvatarName 是你的角色名，ShaderName 是角色的身体部位名称，比

如 `body`, `head`, `arms`, `goggles`, 等（译者注：这里还是应该使用英文。），与材质中的命名保持一致。`TextureType` 是用来指示贴图用途的后缀，比如漫反射 (`d`)，法线贴图 (`b`)，等（参考材质与 `shader` 章节）。

举例：一张贴图名称为“`fluffo_face_d.tga`”，其中 `fluffo` 是角色名，`face` 是 `shader` 名，`d` 是贴图类型，在这里是漫反射贴图。

5.1. 材质类型参考：

- “`d`” 代表漫反射贴图。
- “`b`” 代表法线贴图。（译者注：这里确实是法线贴图，不是凹凸贴图。）
- “`b1-b4`” 代表额外的可变法线贴图。（译者注：翻译存疑。）
- “`s`” 代表镜面反射贴图。
- “`s2`” 代表额外的镜面反射贴图。
- “`sc`” 代表次表面散射贴图。
- “`ao`” 代表环境光遮蔽贴图。
- “`m`” 代表遮罩。（译者注：原文是 `mask`，不太好翻译。）
- “`cm`” 代表颜色遮罩贴图。
- “`iristext_d`” 只被用在“`sht_eye`”中，是虹膜现象的模板。
- “`raotext_d`” 只被用在“`sht_eye`”中，用来模拟虹膜的镜面反射与漫反射。

5.1.1. 漫反射贴图： `d`



以 `RGB` 通道存贮物体的漫反射颜色。`Alpha` 通道被用来表示透明度，`0`（黑色）代表完全透明，`255`（白色）代表完全不透明。

5.1.2. 法线贴图：b



这些贴图被展示在切线空间中并且具有特定用途。更多的相关信息请参考 [GenerateNormalMapsForFaceRig.pdf](#)（译者注：法线贴图以 RGB 通道保存了顶点的法线方向，基准空间为切线空间。法线贴图可以以低成本模拟复杂的表面凹凸光效。总之，法线贴图里保存的不是颜色，是法线向量。）

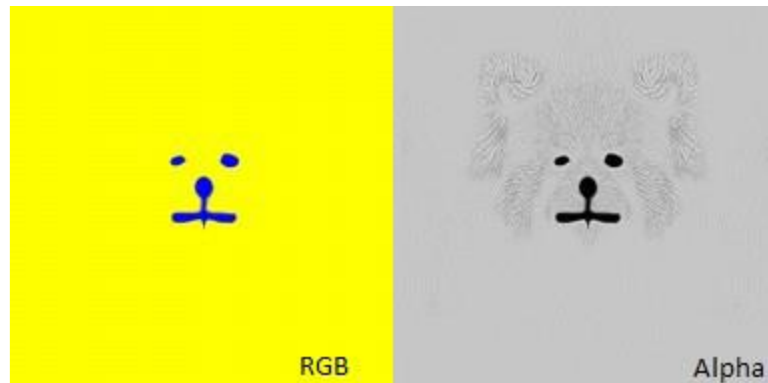
5.1.3. 镜面反射贴图：s



每一个 RGB 通道都包含不同的信息：

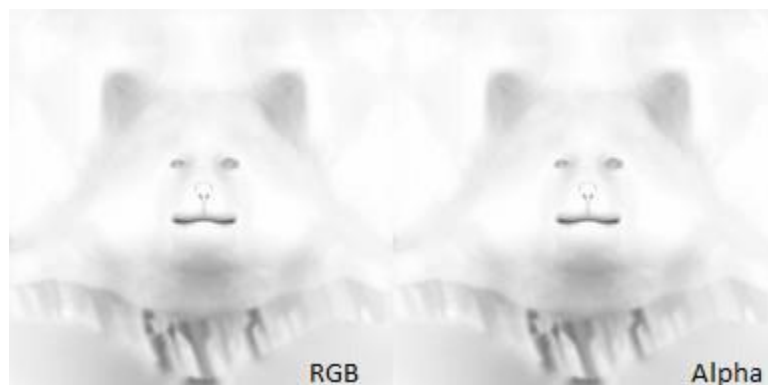
- 红通道代表了光滑度系数，1 表示粗糙表面，255 表示光滑表面。一般使用 1 到 5 之间的系数代表很粗糙的表面，5 到 12 代表极微弱的反光。
- 绿通道被用来设置材质数值（译者注：翻译存疑。），0 表示头发和衣服，255 表示其他材质（比如金属、塑料等）。
- 蓝通道是辐射通道（译者注：名称翻译存疑。），被用来标记那些不受阴影影响的区域（在阴影中发光），0 代表无辐射，255 代表完全辐射。（译者注：这应该是表示物体在多大程度上受到投影的影响。）
- Alpha 通道表示镜面反射强度，0 表示无镜面反射，255 表示最强反射。

5.1.4. 毛发遮罩贴图：m



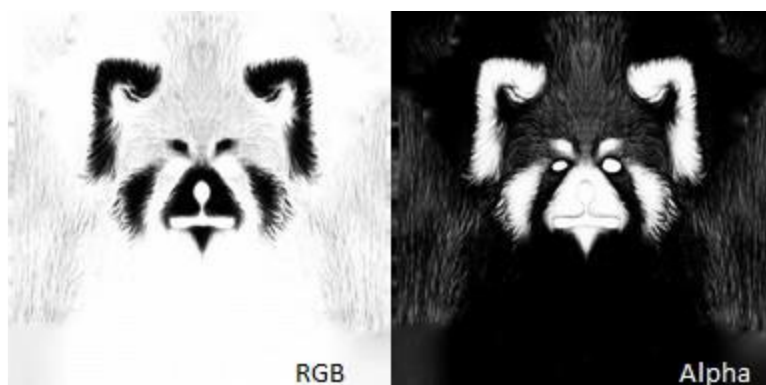
- 红通道：皮肤的镜面反射行为（各向异性的）。255 表示镜面反射类似短毛，0 表示镜面反射类似正常。（译者注：推测是模糊镜面反射光斑的参数。）
- 绿通道：毛发透明度。有时候我们会希望毛发是透明的（不显示毛发），这样就能看到下面的皮肤，比如在脸部。255 表示完全的毛发，0 表示没有毛发。
- 蓝通道：皮肤镜面反射强度。0 代表没有镜面反射（推荐用在皮毛或者头发上），255 表示完全的镜面反射（比如皮肤，舌头，牙齿，指甲）。
- Alpha 通道：毛发的簇，0 代表短毛，255 代表长毛。

5.1.5. 环境光遮蔽贴图：ao



- 红通道应该提供 PPAO（逐像素环境光遮蔽）信息以供绷紧的肌肉使用。
- 绿通道应该提供肌肉松弛时的各向异性光照衰减信息。0 代表无各向异性光照（只有环境光），255 代表最大强度的各向异性光照。
- 蓝通道应该提供肌肉张紧时的各向异性光照衰减信息。0 代表无各向异性光照（只有环境光），255 代表最大强度的各向异性光照。
- Alpha 通道应该提供 PPAO（逐像素环境光遮蔽）信息以供松弛的肌肉使用。

5.1.6. 颜色遮罩纹理: cm



为了能使用户自定义颜色，需要使用一张贴图来标记哪些区域的颜色可以被自定义。被设置为 0 的区域会允许颜色自定义，而被设置为 255 的区域则不受影响，中间值的区域将被插值处理（越接近 0 变化越明显）。你可以同时使用 RGB 通道和 Alpha 通道来控制，这样就可以指定两种颜色。（译者注：具体的混合方式文中没有提及，不过例子中的 Alpha 通道几乎就是 RGB 通道的反相。）

5.2. 为角色和配件制作的额外皮肤

如果你在使用相同 shader 的前提下制作了额外的皮肤，那么至少其中的一张纹理的名字应该包含 “_sk” + 下标。

比如你想给香蕉增加一种额外的颜色，也就是让其中一张纹理——假设是漫反射贴图——有一个额外的版本，那么你应该把它命名为“Fluffo_bandana_sk2_d”之类的。如果这个皮肤是透明的，那么名称应该为“Fluffo_bandana_blend1_sk2_d”，其中“blend1”表示 Alpha 混合，“sk2”表示第二个皮肤。

6. 骨骼

就像我们在引言里提到过的那样，现在允许用户使用自定义的骨骼层级与骨骼名称。然而依旧不允许自定义名称的骨骼有：“Camera”，“BipHead”，“BipLEye”和“BipREye”。这些骨骼会被用于 Facerig 的一些计算步骤，比如视线追随摄像机。另外，摄像机和头部骨骼的 Z 轴必须是前者指向后者的，也就是 Facerig 导入以后的 Z 轴。（译者注：翻译存疑。）有可能用户使用的 3D 软件的在习惯性上标定的 Z 轴与 Facerig 所定义的 Z 轴方向不同，因此照常所谓的 Z 轴在原本的模型空间内实际上是 X 轴或者 Y 轴。如果模型确实被加载了，但是用户在 Facerig

中的屏幕上找不到模型，或许在 3D 软件中旋转模型会帮助解决这一问题。（译者注：这一段说的是，由于 Facerig 和用户使用的 3D 软件的坐标轴定义方式可能会有所不同，所以万一找不着模型，那就尝试把它转一下，来猜一下 3D 软件和 Facerig 的轴对应关系。）首先，尝试把摄像机对准世界坐标系的 x, y, z 三个轴所组成的不同平面（译者注：xOy, yOz 和 xOz 平面。），直到你能把你的模型框在视平面里。如果这时你的模型能出现在 Facerig 的屏幕里了，那你就可以开始收拾其他的摄像机旋转或者移动了。

需要注意的是，对于所有的 Camera 骨骼，FOV（视角）都是固定的 30 度。FOV 不会从外部的设置中导入，也不允许另行更改。因此，为了正确地制作你的角色，应该在 3D 软件中设置 FOV 为 30 度，宽高比为 1.77，以保持和 Facerig 中的设置一致。

为了保证合适的缩放比例，我们强烈建议你将我们的示例角色导入 3D 软件并且用来当模板。测试所用到的数据只有几何信息与 idle 文件，这样就可以快速构建测试用例。

被命名为“props”+什么（比如“propHead”（译者注：原文写的就是 prop，没有 s。））的骨骼是用来表示饰品附件的。这些骨骼可以被自动添加到 prop 骨骼列表中，并且适用于其他角色或模型。

最后，我们强烈建议你逐步导出所需的 idle 文件以及动画文件，并且根据需要做旋转（用导入 Facerig 的方式逐步检查。）大部分程序在导出模型的时候都会对调 Y 轴和 Z 轴，因此逐步检查可以尽早发现问题。

Facerig 支持一些过程式动画，也就是那些可以用特定骨骼控制的动画，比如视线追随摄像机。这些特性依赖 BipHead 骨骼的变换。这根骨骼在 Facerig 中是这样处理的：“BipHead”（既没有影响又没有骨骼树）只被用来指示轴，而你只需要导出最基本的几何信息以及默认动画“idle1”就可以了。在骨骼树中添加一个“BipHead_skin”用来保持骨骼树的连贯吧，它可以和“BipHead”有相同的坐标。如果你对“BipHead”层级的绑骨有疑问，你可以在蒙皮步骤最后再添加它。（译者注：文档这一段到处都是语法问题，实在没法翻译。）

6.1. 伪物理驱动的骨骼

骨骼可以被“碰撞平面”驱动（译者注：物理特性。）在目前的版本中，这些骨骼就像钟摆（比如头发、帽子、胸部之类的骨骼），或者倒立摆（比如“martian antenna”饰品）。

建立物理骨骼的途径如下：这些骨骼直接位于某个被表明父骨骼之下或之上。这些骨骼叫做什么不重要，但是你必须避免它们与之前提到的必需的骨骼名

称相冲突。这里还需要有第三根骨骼，在命名上与物理骨骼的父骨骼相似，但是要添加后缀“_planeNormal”。例如，如果父骨骼名为“BipPhysBoneParent”，第三骨的名称就应该是“BipPhysBoneParent_planeNormal”。第三骨的 Y 轴代表了碰撞平面的法线方向，可以用来限制物理骨骼的运动。比如物理骨骼模拟了一股头发，开发者想要避免头发穿模进入面部，那么碰撞平面就应该被定义为面部的模型面的切平面，法线（Y 轴）朝外。需要注意的是，有些 3D 软件会调换 Y 轴和 Z 轴，所以法线也需要做相应的改变。这种情况下 Z 轴代表法线。

如果碰撞平面不需要，那么它的法线应该指向物理骨骼。另外需要注意的是，尽管碰撞平面的法线是被第三骨定义的，但是它的原点是被物理骨骼的父亲骨骼定义的，所有的变换都发生在父亲骨骼。

总结一下，对于每一根物理骨骼，另外两根辅助骨骼都是需要的：一个父亲骨骼和一个物理碰撞平面。物理骨骼是最终运动的那一个，所以你在刷权重的时候还是要对着它刷。

6.2. 动画与姿势 (pose)

动画必须以 COLLADA 格式提供，帧速率为 30FPS，默认的缩放比例必须一致（取决于你所使用的 3D 软件，1,1,1 或者 100,100,100）。缩放比例在动画中可以调整。

在任何 Y 轴向上的系统中，角色模型应该统一面朝 Z 轴正方向。如果你的系统是 Z 轴向上的，那么模型应该面朝 Y 轴正方向。

Facerig 可以使用基本和添加 (additive) 的动画。总量不多的基本动画，比如 idle1, MouthOpen_base 和 MonthTongueBase，会被要求给出所有骨骼的变换矩阵。而添加的动画则只要求在基本动画的基础上进行变换。它们只会在被添加到某些基本动画后才生效。（译者注：翻译存疑，不过大概意思是模型会先乘算基本动画，然后再累乘添加动画。）

添加动画应该继承它们的基本动画所设置的不动的骨骼（译者注：翻译存疑，也许是在说不应该修改不想动的骨骼？）。比如你在制作一个皱眉的动画，名为“LeftEyebrow_D”，那么你应该只修改眉毛中间的和内部的骨骼，而对于其他骨骼则应该保持和基本动画的第一帧的变换一致，也就是 idle1。这么做是基于两点考虑：第一点，同时也是最重要的一点是，在 Facerig 中，每一项添加动画所做的修改都会累计作用到姿势上。如果你在眉毛动画中对脖子也进行了修改，那么每一次角色皱眉都会对脖子做出改动。第二点，如果骨骼只在他们需要的时候才进行变换，那么脚本就能识别出这些运动并且只对他们做单独的导出，同时忽略剩下的，这会使你的 debug 流程变得更加容易，同时也能避免在添加了大量动

作以后进行矩阵计算所产生的精度问题。（译者注：大概意思是说脚本能自动识别出那些不动的骨骼并抹去。）

原则上来说，所有的添加动画都包含基础动画 `idle1`，更精确地讲，它们保留了对 `idle1` 的引用。然而这里也有些例外：

- 张嘴的动画，也被用来作为“`MouthOpen_base`”基础动画，继而加入 `idle1` 动画。这些动画包括：“`MouthOpen_pursedLips_LR`”，“`MouthOpenLeft_D`”，“`MouthOpenLeft_U`”，“`MouthOpenRight_D`”，“`MouthOpenRight_U`”。
- 当舌头伸出时的舌头动画。这些动画基于“`MouthTongueBase`”，同样也被加入 `idle1` 动画。这些动画包括：“`TongueOut_LR`”，“`TongueOut_UD`”。

动画子文件夹结构：



需要的动画有（第 0 从 `idle1` 开始，除非显式指定从其他地方开始）：

在 `generalMovement` 动画子文件夹：

- `Avatar_FB`：人物在第 0 帧从腰部往上向前俯身，第 15 帧保持 `idle` 姿势，第 30 帧向后仰。（译者注：其实就是一个从向前趴到向后仰的动画，推测 `facerrig` 会计算一个插值，然后从这组动画的合适位置采样得到模型状态。）
- `Avatar_LR`：人物在第 0 帧从腰部往上向左弯曲身体，第 15 帧恢复 `idle` 姿势，第 30 帧向右弯曲身体。
- `Avatar_Twist`：人物在第 0 帧从腰部往上向左扭身，第 15 帧恢复 `idle`，第 30 帧向右扭身。
- `Head_LR`：人物的头部在第 0 帧向左倾斜，第 15 帧恢复 `idle`，第 30 帧向右倾斜。
- `Head_Twist`：人物的头部在第 0 帧向左转动，第 15 帧恢复 `idle`，第 30 帧向右转动。
- `Head_UD`：人物的头部在第 0 帧向上看，第 15 帧恢复 `idle`，第 30 帧向下看。
- `Idle1`：大部分添加动画所依赖的基本动画。这个动画指定了角色自然放置时的姿势，在第 0 帧。它可以包含动画，但应该十分轻微，因为这会被循环播

放并且被表情动画打断。如果 idle 包含了位移，那么它应该是可循环的，这样就能避免在驯化播放动画时产生的不连贯。

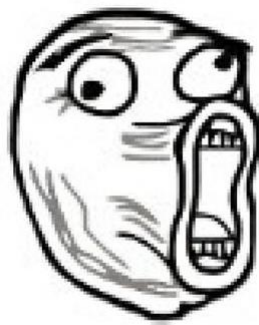
在 EyesAndEyebrows 动画子文件夹中：

- LeftEye_LR: 左眼在第 0 帧向左看，第 15 帧恢复 idle，第 30 帧向右看。
- LeftEye_UD: 左眼在第 0 帧向上看，第 15 帧恢复 idle，第 30 帧向右看。
- LeftEyebrow_D: 左眉毛靠里的一半向下压，就像皱眉的表情。
- LeftEyebrow_D_ext: 左眉毛靠外的一半向下压。
- LeftEyebrow_U: 左眉毛靠里的一半向上升，就像惊讶的表情。
- LeftEyebrow_U_ext: 左眉毛靠外的一半向上升。
- LeftEyeClosed: 左眼上眼睑在第 0 帧位于比 idle 稍微靠上一点的位置，第 10 帧恢复 idle，第 30 帧完全闭上。
- LeftEyeSquint: 左眼下眼睑在第 30 帧轻微闭上，就像眯着眼睛那样。
- LeftEyeWideOpen: 左眼完全睁开，也就是上下眼睑所能分开的最大距离。
- RightEye_LR: 右眼在第 0 帧向左看，第 15 帧恢复 idle，第 30 帧向右看。
- RightEye_UD: 右眼在第 0 帧向上看，第 15 帧恢复 idle，第 30 帧向右看。
- RightEyebrow_D: 右眉毛靠里的一半向下压，就像皱眉的表情。
- RightEyebrow_D_ext: 右眉毛靠外的一半向下压。
- RightEyebrow_U: 右眉毛靠里的一半向上升，就像惊讶的表情。
- RightEyebrow_U_ext: 右眉毛靠外的一半向上升。
- RightEyeClosed: 右眼上眼睑在第 0 帧位于比 idle 稍微靠上一点的位置，第 10 帧恢复 idle，第 30 帧完全闭上。
- RightEyeSquint: 右眼下眼睑在第 30 帧轻微闭上，就像眯着眼睛那样。
- RightEyeWideOpen: 右眼完全睁开，也就是上下眼睑所能分开的最大距离。

在 MouthAndNose 动画子文件夹：

- CheekPuff_L: 左脸鼓起。
- CheekPuff_R: 右脸鼓起。
- Mouth_pursedLips_LR: 这个动画管理的是撅嘴的动作。在第 0 帧它朝左，第 15 帧居中，第 30 帧朝右。（译者注：这个动画我不确定是用来干什么的，可能是配合鼓起脸颊的。）
- Mouth_unveilledTeeth_D: 下嘴唇朝下，露出下牙。
- Mouth_unveilledTeeth_U: 上嘴唇朝上，露出上牙。
- MouthClosedLeft_D: 闭合的左嘴角向下，就像悲伤的表情。

- **MouthClosedLeft_U**: 闭合的左嘴角向上，张开嘴唇并且露出牙齿，就像大笑的表情。
- **MouthClosedLeft_U_visime**: 闭合的左嘴角向上，同时保持嘴唇闭合，就像不明显的笑容。它被叫做 visime 是因为它会和 visime 一起使用，允许用户笑的同时说话。
- **MouthClosedRight_D**: 闭合的右嘴角向下，就像悲伤的表情。
- **MouthClosedRight_U**: 闭合的右嘴角向上，张开嘴唇并且露出牙齿，就像大笑的表情。
- **MouthClosedRight_U_visime**: 闭合的右嘴角向上，同时保持嘴唇闭合，就像不明显的笑容（微笑）。它被叫做 visime 是因为它会和 visime 一起使用，允许用户微笑的同时说话。
- **MouthOpen**: 嘴在第 0 帧位于 idle，第 30 帧完全张开。
- **MouthOpen_base**: 所有“MouthOpen”添加动画的基础动画。它自始至终都是同样的嘴巴完全张开的姿势（从第 0 帧到第 30 帧）。它与 MouthOpen 的最后一帧是相同的姿势。
- **MouthOpen_pursedLips_LR**: 自始至终保持撅嘴的姿势。它应该从张嘴的姿势修改得来。在第 0 帧向左，第 15 帧居中，第 30 帧向右。（译者注：可能是指这个表情：



- **MouthOpenLeft_D**: 从张嘴的姿势开始，到第 30 帧时左嘴角向下。
- **MouthOpenLeft_U**: 从张嘴的姿势开始，到第 30 帧时左嘴角向上，像一个微笑的表情。
- **MouthOpenRight_D**: 从张嘴的姿势开始，到第 30 帧时右嘴角向下。
- **MouthOpenRight_U**: 从张嘴的姿势开始，到第 30 帧时右嘴角向上，像一个微笑的表情。
- **MouthTongueBase**: 嘴从 idle 姿势开始，并且逐渐张大到正好可以吐舌头的程度，最后一帧整个舌头都要露出来。它是舌头动画的基础动画，但是“TongueIdle”除外。

- **NoseWrinker_D**: 角色的鼻孔轻微向下移动。
- **NoseWrinker_U**: 角色的鼻孔轻微向上移动。
- **TongueIdle**: 舌头的闲置动画, 可以包含一些轻微的移动来让舌头看起来更自然而不是完全僵硬的。它从 **idle** 开始, 结束到 **idle**。
- **TongueOut_LR**: 舌头伸出, 在第 0 帧向左移动, 在第 15 帧位于 **MouthTongueBase** 的最后一帧的位置, 在第 30 帧向右移动。
- **TongueOut_UD**: 舌头伸出, 在第 0 帧向上移动, 在第 15 帧位于 **MouthTongueBase** 的最后一帧的位置, 在第 30 帧向下移动。

在 **ShouldersAndHands** 动画子文件夹中:

- **FingerL0_extFlex**: 左手拇指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerL1_extFlex**: 左手食指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerL2_extFlex**: 左手中指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerL3_extFlex**: 左手无名指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerL4_extFlex**: 左手小拇指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerR0_extFlex**: 右手拇指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerR1_extFlex**: 右手食指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerR2_extFlex**: 右手中指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerR3_extFlex**: 右手无名指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **FingerR4_extFlex**: 右手小拇指从伸展 (第 0 帧) 到收起 (第 30 帧), 就像握拳的时候那样。第 15 帧的时候位于 **idle** 位置。
- **HandL_closeDown_LR**: 左臂从较低高度贴近身体从左边移动到右边 (未完全伸展)。
- **HandL_closeMiddle_LR**: 左臂从中等高度贴近身体从左边移动到右边 (未完全伸展)。

- **HandL_closeUp_LR**: 左臂从最大高度贴近身体从左边移动到右边（未完全伸展）。
- **HandL_farDown_LR**: 左臂从较低高度以最大距离远离身体从左边移动到右边（完全伸展）。
- **HandL_farMiddle_LR**: 左臂从中等高度以最大距离远离身体从左边移动到右边（完全伸展）。
- **HandL_farUp_LR**: 左臂从最大高度以最大距离远离身体从左边移动到右边（完全伸展）。
- **HandL_solo_LR**: 左手绕手腕从第 0 帧向左摆动到第 30 帧向右摆动，其中第 15 帧位于 idle 位置。
- **HandL_solo_Twist**: 左手绕手腕从第 0 帧向左旋转到第 30 帧向右旋转，其中第 15 帧位于 idle 位置。
- **HandL_solo_UD**: 左手绕手腕从第 0 帧向上摆动到第 30 帧向下摆动，其中第 15 帧位于 idle 位置。
- **HandR_closeDown_LR**: 右臂从较低高度贴近身体从左边移动到右边（未完全伸展）。
- **HandR_closeMiddle_LR**: 右臂从中等高度贴近身体从左边移动到右边（未完全伸展）。
- **HandR_closeUp_LR**: 右臂从最大高度贴近身体从左边移动到右边（未完全伸展）。
- **HandR_farDown_LR**: 右臂从较低高度以最大距离远离身体从左边移动到右边（完全伸展）。
- **HandR_farMiddle_LR**: 右臂从中等高度以最大距离远离身体从左边移动到右边（完全伸展）。
- **HandR_farUp_LR**: 右臂从最大高度以最大距离远离身体从左边移动到右边（完全伸展）。
- **HandR_solo_LR**: 右手绕手腕从第 0 帧向左摆动到第 30 帧向右摆动，其中第 15 帧位于 idle 位置。
- **HandR_solo_Twist**: 右手绕手腕从第 0 帧向左旋转到第 30 帧向右旋转，其中第 15 帧位于 idle 位置。
- **HandR_solo_UD**: 右手绕手腕从第 0 帧向上摆动到第 30 帧向下摆动，其中第 15 帧位于 idle 位置。

在 visime 动画子文件夹中：

visime 是用于嘴唇同步的动画，每一个动画的名字对应着与其匹配的相关语音。举个例子，visime_new_EH-AE 包含了模型发出 EH 或者 AE 声音时嘴唇与下巴的位置信息。对于每一个 visime，第 0 帧与第 30 帧是完全相同的，都包含了需要变换的骨骼的位置，并且中间没有其他的关键信息（译者注：此处翻译存疑，key 根据上下文无法理解，姑且翻译为关键信息。）。以下是 visime 动画：

- visime_new_AA
- visime_new_AH
- visime_new_AO
- visime_new_AW-OW
- visime_new_CH-J-SH
- visime_new_EH-AE
- visime_new_EY
- visime_new_FV
- visime_new_IH-AY
- visime_new_L
- visime_new_M-P-B
- visime_new_N-NG-DH
- visime_new_OY-UH-UW
- visime_new_R-ER
- visime_new_W
- visime_new_X
- visime_new_Y-IY

译者注：此处的动画对应的是音频跟踪口型的功能，然而注音法实在是看不懂。姑且可以认为这部分动画不重要，制作时可以全用临时素材代替，但是似乎不能跳过。

anim 文件夹中的目录：

- **_frown**：双眉同时皱眉的状态，看起来就像是同时应用了 RightEyebrow_D 和 LeftEyebrow_D。这个动画被渲染节点引用以激活对应的法线贴图。
- **_laugh**：两边的嘴角同时上扬的状态，看起来就像是同时应用了 MouthClosedLeft_U 和 MouthClosedRight_U。这个动画被渲染节点引用以激活对应的法线贴图。
- **_unveilTeeth**：看起来就像是同时应用了两边的露出牙齿的动画。这个动画被渲染节点引用以激活对应的法线贴图。

- **_wonder:** 双眉同时惊奇的状态，看起来就像是同时应用了 **RightEyebrow_U** 和 **LeftEyebrow_U**。这个动画被渲染节点引用以激活对应的法线贴图。
- **cheekL:** 这个动画在用户启用重定向功能以后被用来修正模型嘴部周围的骨骼。由于这些骨骼不会被追踪，所以它们的位置将由这个动画来确定。它在第 0 帧必须是 **idle** 姿势，此后的序列帧一般包含张嘴，笑和噘嘴姿势。这个动画被设置在中间和左边的骨骼上。
- **cheekR:** 与 **cheekL** 功能相同，只是在右边生效。
- **special01:** 这是一个被用来临时替换 **idle** 动画的基础动画，使用无追踪运动以及关键帧动画技术。这个关键帧动画应当被限制于身体、手和头部上，同时保留面部不动。这样角色就可以在保持面部被追踪的同时让身体做一些特殊的动作。
- **special02:** 与 **special01** 相同。
- **special03:** 与 **special01** 相同。
- **special04:** 与 **special01** 相同。
- **special05:** 与 **special01** 相同。
- **special06:** 与 **special01** 相同。

如果你完成了全部的动画后发现角色的运动依旧有问题，你可以重复以上过程，并且在 **BipRoot** 骨骼下面添加一根骨骼。你可以叫它“**bipRoot_head**”（这根骨骼只能在包含几何信息以及动画“**idle1**”的导出文件中找到），并且你可以在“**idle1**”动画中点击它以获得对应的坐标。

如果导入以后发现你的角色太小（你可以通过观察阴影来发现这一点，它会突然向着某个方向延伸一截（译者注：翻译存疑，原文是 *you can see this looking at the shadows, they will have a sudden pass in steps*，姑且怀疑这里的 *pass* 有特殊用法。)), 你应该缩放它。你必须创建一个“**cfg**”文件来完成这一点。它应该叫做“**cc_avatarName**”（如果你不想这么做的话，那也可以复制一份由导入工具生成的“**cfg**”文件，重命名它并且清空里面的内容）。在这份文档中，你需要输入命令“**set_avatar_scale XX**”。

7. 角色配置文件

角色配置文件允许用户设定角色的一些参数。如果放置在 **COLLADA** 文件的地方同时放置了一份 **txt** 文件，那么导入脚本就会自动把它当做配置文件，并且和其他的角色数据一同拷贝。

| | | | | | | |
|--|--|--|--|--|--|------------------|
| | | | | | | BipLEyelid_up1 |
| | | | | | | BipSnout |
| | | | | | | BipLEye |
| | | | | | | BipREye |
| | | | | | | BipREyebrow_0 |
| | | | | | | BipREyebrow_1 |
| | | | | | | BipREyebrow_2 |
| | | | | | | BipREyebrow_3 |
| | | | | | | BipREyebrow_4 |
| | | | | | | BipREar |
| | | | | | | BipREyeCornerIn |
| | | | | | | BipREyeCornerOut |
| | | | | | | BipREyelid_down1 |
| | | | | | | BipREyelid_up1 |
| | | | | | | BipRNose |
| | | | | | | BipRCheek_2 |
| | | | | | | BipRCheekMiddle |
| | | | | | | BipRCheek_1 |
| | | | | | | BipRCheekUpper |
| | | | | | | BipRCheek_3 |
| | | | | | | BipRLipCorner |
| | | | | | | BipRLipUpper_2 |
| | | | | | | BipRLipUpper_1 |
| | | | | | | BipRLipUpper_3 |
| | | | | | | BipRLipUpper_4 |
| | | | | | | BipLHead_D |
| | | | | | | BipLHead_U |
| | | | | | | BipRHead_D |
| | | | | | | BipRHead_U |
| | | | | | | BipLClavicle |
| | | | | | | BipLUpperArm |
| | | | | | | BipLForearm |
| | | | | | | BipLHand |
| | | | | | | BipLFinger4 |

| | | | | | | | | | BipRFinger01
| | | | | | | | | | BipRFinger02
| | | | | | | | | | BipRWristCorr
| | | | | | | | | | BipRWrist
| | | | | | | | | | BipRShoulder1
| | | | | | | | | | BipLShoulder2
| | | | | | | | | | BipLShoulder3
| | | | | | | | | | BipRShoulder2
| | | | | | | | | | BipLShoulder4
| | | | | | | | | | BipRShoulder3
| | | | | | | | | | BipRShoulder4
| | | | | | | | | | BipLThigh
| | | | | | | | | | BipLCalf
| | | | | | | | | | BipLFoot
| | | | | | | | | | BipLToe
| | | | | | | | | | BipRThigh
| | | | | | | | | | BipRCalf
| | | | | | | | | | BipRFoot
| | | | | | | | | | BipRToe

Camera