

Harmony22

**Toon Boom
Harmony22**

游戏



**toon
boom**

TOON BOOM ANIMATION INC.

4200 St.Laurent Blvd, Suite 1020
Montreal, Quebec, Canada
H2W 2R2

+1 514 278 8666

contact@toonboom.com
toonboom.com

法律声明

Toon Boom Animation Inc.
4200 Saint-Laurent, Suite 1020
Montreal, Quebec, Canada
H2W 2R2

电话: +1 514 278 8666
传真: +1 514 278 2666

toonboom.com

免责声明

本文件的内容属于 Toon Boom Animation Inc. 的财产, 受版权保护。严禁复制本指南的全部或部分内容。

本文件的内容由适用的许可协议提供特定的有限保证并规定赔偿责任的排除和限制, 该许可协议的附件包含针对 Adobe® Flash® 文件格式 (SWF) 的特殊条款和条件。有关详情, 请参考许可协议以及上述特殊条款和条件。

本文档中的某些图标由 Font Awesome 的 Font Awesome Free 5.6.1 提供。这些图标根据 CC BY 4.0 许可证提供。有关 Font Awesome 的更多信息, 请访问: <https://fontawesome.com>。有关 Font Awesome Free 的许可证的信息, 请访问: <https://fontawesome.com/license/free>。

本文档中的某些图标由 Glyphicons 的 Glyphicons Halflings 字体提供。有关 Glyphicons 的更多信息, 请访问: <https://www.glyphicons.com/>。

商标

Toon Boom® 是注册商标。Harmony™ 和 Toon Boom 徽标是 Toon Boom Animation Inc. 的商标。所有其他商标都是其各自所有者的财产。

出版日期

02/22/2023

版权所有 © 2023 Toon Boom Animation Inc., Corus Entertainment Inc. 旗下公司 公司。保留所有权利。

目录

目录	3
关于游戏制作	5
Harmony Gaming SDK 2022 版本说明	6
第 1 章: 导出 Harmony 游戏制作游戏	1
第 2 章: 关于游戏资产创建	4
游戏绑定准则	5
游戏变形准则	7
关于游戏骨骼变形	8
游戏切割器准则	9
创建元数据注释	10
游戏动画小贴士	12
制作多个镜头的动画	13
工作流 1: 分离场景	14
工作流 2: 使用场景标记分离	15
关于正射摄像机	16
第 3 章: 关于导出到 Unity	17
设置锚点	18
导出 Sprite Sheet	19
精灵分辨率	21
使用 Bake_Group	23
导出至 Easel JS	24
色板变体	25
第 4 章: 关于 Harmony Unity SDK	27
Harmony Previewer	31
资产操控	33
关于 Unity 示例项目	39
关于 Unity 界面	40
Harmony 文件导入 Unity	42
Unity 中的 TBG 文件工作流	44
Unity 中的 XML 文件夹工作流	50
向空游戏对象添加 Harmony Renderer	54
在 Unity 设置碰撞	56

游戏术语	57
------------	----

关于游戏制作

Harmony 允许导出 Harmony 内创建的动画到外部游戏引擎。这样您就可以设计带简单动作、可用于游戏引擎的角色。

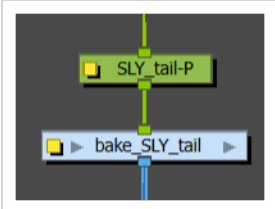
可合并 Harmony 数据的游戏引擎之一是 Unity。Toon Boom 在 Unity Asset Store 提供 Harmony SDK 下载，用于控制导入到 Unity 的角色。

本指南演示如何从 Harmony 导出动画。还会详细说明 Harmony Game SDK 与 Unity 的集成。

Harmony Gaming SDK 2022 版本说明

从 Harmony 导出

功能	描述
<p>新增导出器选项</p>	<p>为使游戏编辑器提供更好的支持, Export to Sprite Sheet (导出到Sprite Sheet) 界面增加了一些功能, 可在导出文件打包更多信息。</p>  <p>新选项</p> <ul style="list-style-type: none"> • Palette selection(色板选择) 选项卡 - 此功能可排除部分不希望导出文件包含的色板, 涵盖所有可能的调色板变体, 使角色呈现各种颜色 • Encode to TBG(编码成 TBG) - 将 XML 文件压缩为 TBG 自定义文件格式, 使之可以通过 Unity Custom Importer 导入到 Unity。文件导入后视为 Unity 的资产, 包含纹理、精灵和动画等 sub-asset。这样可以改进角色数据创建与移动的人体工程学, 以及检查源代码管理的变更。 • “Save and Export”(保存并导出) 性能提升 - 导出过程时间通常只有 Harmony 21 的 1/3, 同时适用于 XML 文件夹和 .tbg 文件。
<p>“bake_”组</p>	<p>目前有很多节点无法在 Game SDK 中渲染。新增功能可在导出加入这些节点的值, 将绘图及对其有影响的节点分为带“bake_”前缀的新组(如 bake_hips)。</p>

功能	描述
	 <p>任何以“bake_”前缀命名的组都将进行分析,用于效果、移动和绘图替换的特殊动画帧。导出到 spritesheet 时,该组中的每个特殊帧都作为精灵导出。如果时间轴有许多帧在该组内没有任何变化,在 Sprite Sheet 内将视为同一导出精灵</p>

在 Unity 导入 TBG 文件

功能	描述
Automatic Importing Process	<p>Unity 会自动检测 TBG 文件的更改,并使用自定义导入器重新导入。然后将该文件作为 prefab 拖到场景中,类似 FBX 文件。</p> <p>纹理、精灵、动画和 TbgStore 资产将作为主 TBG prefab 资产下的 sub-asset 生成并提供。这些资产无需再源代码管理中单独管理,只放在 Unity 的库缓存中。AnimatorController 和 SpriteSheet 资产无法作为 sub-asset 运行,因此在角色旁边生成。</p>
Importer Settings	<p>“导入”流程可在 inspector view 的设置中修改。</p> <p>动画设置</p> <ul style="list-style-type: none"> • Discretization Step - 精灵在进行骨骼变形时分割成的列数/行数。 • Framerate - 动画运行时每秒的帧数。 • Stepped - 创作的帧间禁用插值。 • Create Animator Controller - 创作的帧间禁用插值。 • AnimatorController - 提供一个在特定条件下动画相互过渡的工作区。可以在游戏脚本中引用以在游戏过程中触发动画。 • 保留复制剪辑的曲线:从 .tbg 文件中的剪辑 sub-asset 复制(并在 AnimatorController 中引用)的曲线将被更新的 .tbg 文件的新数据覆盖。 <p>材质设置</p> <ul style="list-style-type: none"> • Shader - 用于渲染 prefab 内的所有精灵 • SRGBTexture - 从纹理映射伽玛颜色空间到线性颜色空间渲染 • FilterMode - 纹理的相邻像素如何插值 • Mipmap 启用 - 生成较低分辨率的纹理,增加渲染距离 • CreateSpriteAtlas - 如果下方未引用 SpriteAtlas,将在项目中、.tbg 文件旁边新建一个 SpriteAtlas 资产

功能	描述
	<ul style="list-style-type: none"> SpriteAtlas - .tbg 文件中的所有精灵都可以添加到 SpriteAtlas 中，完成后可将多个新的 SpriteRenderer 组合成一个绘制调用，提高渲染性能

在 Unity 渲染 TBG 文件

功能	描述
Prefab Hierarchy	<p>Unity 中生成的 prefab 的结构应与 Harmony 时间轴中设置的分级结构相匹配。添加到输出节点的 Harmony 的合成的节点视图结构，导出期间会被排除。</p> <p>新 GameObjects 可以附加到以上任意子定位钉变换。因此允许附加程序设备。</p> <p>此外，运行时可启用或禁用任意 GameObject，因此角色的可选视觉添加可以禁用，需要时再启用。该层级运行时也可完全分解、重组或复制。因而可以选择爆开角色或将某些图画作为抛射体射击。</p>
TBG Renderer 设置	<p>分辨率、色板、材质、颜色和蒙皮可通过检查器和 GameObject 的自定义脚本编辑。SpriteRenderer 的材质和颜色也可以每个绘图单独设置，因此可以更改角色配饰的颜色，或通过自定义 shader 为某些绘图添加特殊特效。</p>
2D Animation Package 集成	<p>使用 Unity 的“2D Animation Package”，在 Harmony 中由游戏骨骼变形的绘图在 Unity 中将转换为 SpriteSkin 骨骼。此包为 SpriteRenderer 提供多线程高性能网格变形的支持，并提供在 Scene view 中移动骨骼的 UI 图柄。可以使用 Unity 的“IK Manager 2D”对骨骼链制作程序动画，在完成动画 Update(更新) 步骤后混合瞄准、倾斜、摇晃、挥动等特效。</p>
动画	<p>所有动画数据均从 Harmony 逐定位钉导出，因此所有中间旋转和缩放都在 Unity 的动画窗口中展示。动画关联的 AnimationClip 资产从 TBG sub-asset 复制到项目中的新资产后，动画可以修改，从而允许在 Unity 中添加事件、提供额外曲线。</p> <p>AnimationClip 通过 AnimatorController 选择，提供动画过渡方法，可以直接过渡，也可以通过平稳混合定位钉变换来过渡。</p>
Shader Graph 集成	<p>SDK 中包含兼容的 shader graph shader，可复制和修改，形成简便界面，为角色添加自定义的游戏特定视觉特效。</p>

HarmonyRenderer 改进(XML 文件夹)

功能	描述
HarmonyProject 预览	在 Unity Asset 文件夹中选择角色后, preview window 会显示角色的预览。Preview window 播放角色动画的预览, 在其中就可以更改蒙皮, 无需将角色拖到 scene view 中。

修复

此次版本的 Harmony Game SDK 修复了以下问题：

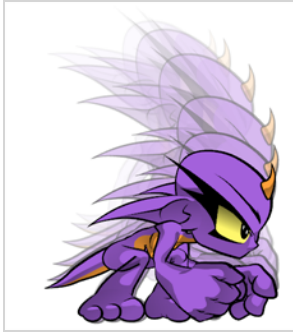
HarmonyRenderer

- 以前更新锚点变换时泄漏内存的锚点。更新 HarmonyRenderer C++ 插件后解决了此问题。
- HarmonyRenderers 渲染性能提升。现在从 HarmonyRenderer C++ 插件接收新网格, 这些插件可以通过更新的 Unity 网格生成功能直接传递到 Unity 的 C++ 后端。大大减少了 Harmony Renderer 角色动画的更新时间, 允许屏幕同时显示更多角色。

第 1 章：导出 Harmony 游戏制作游戏

从 Harmony 导出数据到您的游戏引擎主要有两种工作流程：

- 第 1 页上的 [导出 Harmony 游戏制作游戏](#)
- 第 1 页上的 [导出 Harmony 游戏制作游戏](#)



Harmony 中创建动画

为游戏创建角色绑定和动画之前需考虑几点。艺术家和程序员应协作，使需求都得到满足。

- 游戏将针对哪些平台创作？ Windows、Mac、手机、iOS、PS、XBOX 等
- 使用什么游戏引擎？
- 游戏使用什么动画风格？ 手绘、cut-out、带纹理等

开始创作前需考虑以上等问题。它们将影响设计、构建和制作动画角色的方式。

例如，如果计划开发一款智能手机游戏，游戏大小要控制在 50 MB 以下，便于没有 Wi-Fi 的情况下下载。这种情况下，最重要的是使用非常紧凑的 **Sprite Sheet** 创建高效的角色，并大量重复使用动画，缩小文件大小。期间您需要：

- 在 Harmony 绑定和动画化角色。
- 提取 Harmony 数据。
- 将 Harmony 数据导入游戏引擎。

记住，如果您使用定制引擎，也可以处理导出的 Harmony 数据，应用于定制引擎。如果您需要数据适应引擎方面的帮助，请通过 store.toonboom.com/contact/support 联系。

如果为 PS 或 Xbox 等游戏机制作游戏，那么完全可以制作大型纹理。您可以逐帧制作动画，使用 cut-out 角色，或两者兼具。

如果只按逐帧序列导出，可以不受限地使用 Harmony 的所有工具。然后您可以将导出的图像序列处理成一个 **Sprite Sheet**。

原始游戏数据导出(导出到 Sprite Sheet)



如果希望文件大小尽可能小,宜导出原始游戏数据。Toon Boom允许您直接从 Harmony 场景转换数据,并入游戏引擎。您可以提取骨骼信息、绘图信息、关键帧动画数据、变形(仅骨骼和关节)、切割器、透明度节点以及计时列。

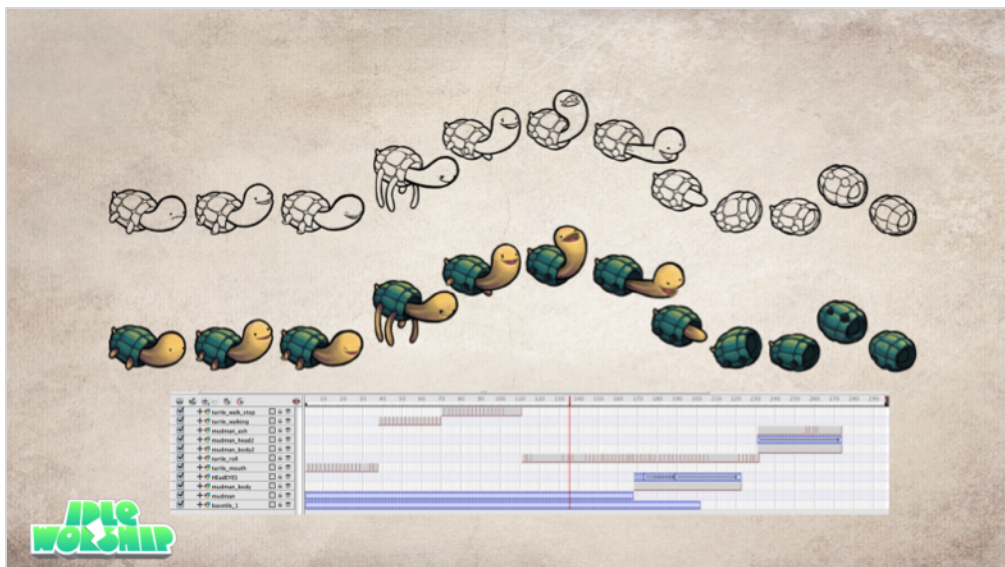
- **利:**最轻量级的导出,减少文件大小,适合移动应用程序。
- **弊:** Harmony中的工具使用受限。可以使用变型以及曲线和封套变形器等工具,但需要烘焙成绘图,游戏引擎才能正确解释。可以使用切割器特效(遮罩),但不能进行级联,也就是说层级链中只能有一个。绑定上可以使用游戏骨骼变形器,无需烘焙到绘图。

即便有诸多限制,您仍可以在 Harmony 创作出很好的 cut-out 角色动画并提取全部兼容数据。移动、旋转、缩放和倾斜不同的绘图图层,可以创建高级外观的动画。

提取数据后的 Sprite Sheet 只包含 Harmony 场景文件所使用的的身体部位的绘图。重复使用相同骨骼和绘图支持创作不同动画,例如空闲、奔跑和跳跃。

Toon Boom已将此解决方案完全集成到 Unity 游戏开发渲染引擎。如果在 Unity 中创作游戏,将无缝衔接流程,无需重新处理游戏引擎的数据。

逐帧导出(导出到 Easel JS)



可使用图像序列从 Harmony 导出,导出后可以重新编译为 Sprite Sheet。

利:制作动画时随意使用 Harmony 的工具！

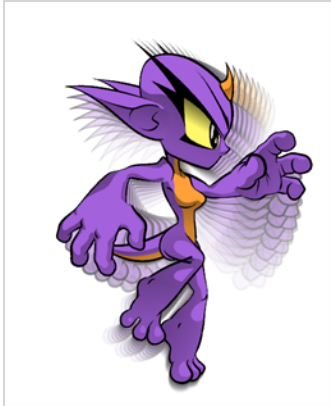
弊:这些 **Sprite Sheet** 很大，导入游戏变成大型纹理，处理起来复杂。对于主机游戏等类型没什么问题，但如果是开发移动设备（如 **iOS** 或 **Android**）游戏，则要求小纹理空间。而且文件大小应保持在 **50 MB** 以内，便于无 **Wi-Fi** 的情况下下载。

Harmony 提供一个将逐帧动画导出到 **EaselJS** 游戏引擎的脚本。**EaselJS** 游戏引擎兼容 **javascript web**。该脚本采用 **json** 数据结构将每一帧匹配到 **Sprite Sheet** 的一个条目。其数据结构易于理解，需要时可以适应或转化到不同游戏引擎。

如需导出到 **EaselJS**，应在“脚本编辑”工具栏添加脚本（**TB_ExportToEaselJS**）。Harmony 包含此脚本。参见脚本编辑指南进一步了解如何向脚本编辑工具栏和接口参数添加脚本。

第 2 章：关于游戏资产创建

本节提供绑定、变形、切割器的指南和许多有用的动画小贴士。



游戏绑定准则

下面列出绑定角色时需牢记的准则。为游戏规划角色绑定时，请遵循角色风格，定制色板。但是如打算提取游戏数据，需要考虑进行一些限制：

- 将Harmony场景设置为正方形分辨率(如 1024 x 1024)。可在 **Scene Setting**(场景设置对话框)中操作—参阅 参考指南。
- 只在线条艺术和色彩艺术图层上绘制。顶层和底层包含的信息不会导出到 **Unity**。但可以将其作为参考图层，只是不会导出内容。
- 若需要重新排序图层，请在 **Z** 空间中微移图层。但角色绑定中不支持较大 **Z** 偏移。
- 将游戏引擎中的每个图层制作成Harmony中的分隔图层。如有两个不同深度的角色，请放入不同的场景文件。
- Harmony场景根级组的角色绑定在 **Unity** 中将渲染到单个面，但使用单独的 **Sprite Sheet** 和动画数据集。对于可能有多个角色交互的场景，请留意这一点。
- 不使用 **3D** 空间。如果要在 **3D** 空间中设置，可以在游戏引擎操作。
- 使用“旋转”工具在“定位钉”图层上设置轴心点，使之成为整个图层的轴心点。建议使用定位钉轴心点而非绘图轴心点。即使不制作动画，绘图图层也应设置轴心点，后续若在绘图层上放置锚点，就可以在游戏引擎内检索信息。
- 不使用变型。游戏引擎暂不支持。
- **Morphing**(变型)和**Deformations**(变形)都可以应用后烘焙为单独绘图。这些绘图之后将在 **Sprite Sheet** 中显示为新绘图。不要经常这样操作，它会增大纹理空间。
- 导出时留意角色的位置。导出的游戏对象的主轴心点就是 **Harmony** 场景的中心 (0,0)。
- 层级最后要有一个“显示”。
- 绑定时，请记住导出器不支持某些节点。游戏引擎不能很好地解释这些节点结构。因此建议您避免使用复杂的节点结构。为此，请您：
 - 在附于定位钉的图层之外创建角色。定位钉之后可能会关联其他定位钉，形成层级。
 - 所有定位钉应关联到节点视图顶部的一个主定位钉
 - 绘图通常应关联合成，确定屏幕上绘图的显示顺序
 - 如果要在动画制作期间更改绘图顺序，可以拖拽 **z** 轴深度更改首张绘图
 - 如果仅使用绘图和定位钉无法使动画正常工作，还可以其他允许导出的节点：
 - **Cutter(切割器)** - 一般只对照一张图纸切割另一张图纸(如按头部边界切出脸部)。不能将一个切割器的输出与另一切割器的输入建立“菊花链”。
 - **Game Bone Deformation(游戏骨骼变形)** - 一般只变形一张绘图。骨骼可以分岔。可以使用运动学输出在骨骼末端附加绘图(如腿部末端的脚)



注意

您还可以在“bake_groups”中使用更复杂的节点结构。参阅[使用烘焙组](#)。

牢记以上建议，再充分利用强大的工具，您就能在Harmony创作紧凑高效的 2D 游戏角色。

应做到以下几点：

- 在“时间轴”视图中创建简单的父级-子级关系层级。
- 用定位钉图层包含关键帧动画数据, 设置为“分隔位置”。
- 使用绘图图层绘图, 需要时创建新绘图。
- 使用“旋转”工具在“定位钉”图层上设置轴心点。
- 合理命名图层, 以便后续需要在游戏引擎中获取特定图层的轴心点时可以轻松识别目标图层。假设有一个顶级组 **A**, 其中包含一个子组(组 **B**), 而绘图图层是组 **B** 的子级, 则导出 **A_B_DrawingLayer** 绘图图层。
- 将锚点设置于希望定位钉在 **Unity** 中显示为变换的位置。

可以随意使用所需的绘图工具: 铅笔和笔刷工具、纹理线条、实心区域和渐变。每个单独绘图渲染后组成一个 **Sprite Sheet**。



注意

由于 **Unity** 游戏引擎不支持 **Unicode** 角色, 建议避免在游戏场景中使用。

游戏变形准则

Harmony 可将变形导出为游戏引擎 XML 格式。只有游戏骨骼组成的层级可以成功导出。另外还会执行运动学输出补充游戏骨骼。

Harmony 中的变形具有各种特性。但其中一些特性与游戏引擎 SDK 不兼容，不可用于游戏制作。因此，游戏骨骼变形器不支持以下标准骨骼变形器功能：

- 影响区域
- 曲线和封套变形器
- 变形层级中有多个姿势
- 动画化变形组下方的绘图定位钉

游戏引擎 SDK 中的变形与 Harmony 中的骨骼变形不尽相同。

为与大多数游戏引擎保持一致并维持快速计算，SDK 执行一种线性基蒙皮算法混合关节处的骨骼。您可能会发现一些取决于所使用关节的曲率的差异。也就是说，相较于常规骨骼变形，Harmony 中 GameBone 与导入到 Unity 后所看到内容更相似。



注意

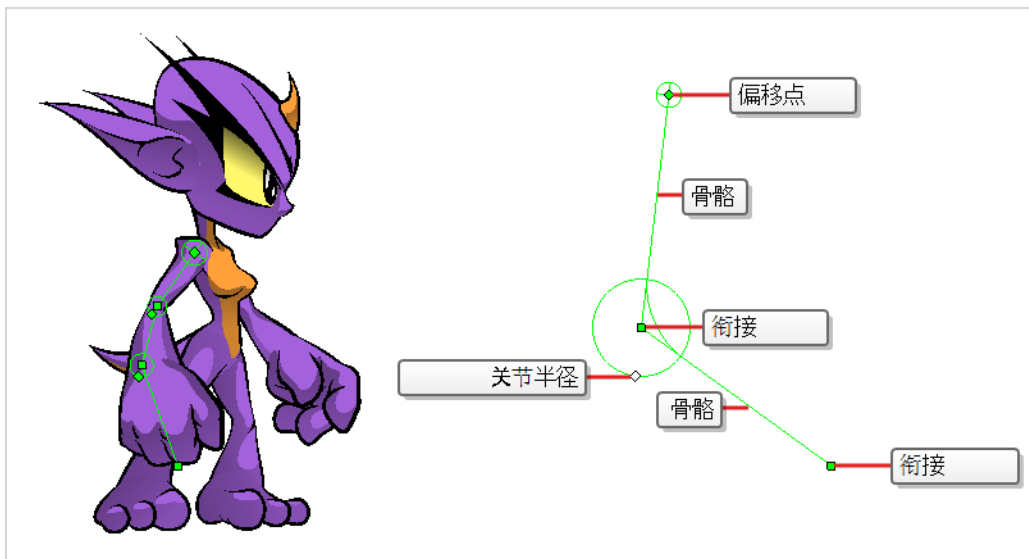
* 目前在游戏引擎 SDK 执行 Cocos2d-x 时不可用。

关于游戏骨骼变形

T-RIG-007-003

游戏骨骼变形与骨骼变形极为相似。它让您可以创建类似骨骼的结构,其中每个部分都是固体,但是具有灵活的关节。这对于为角色的四肢(例如手臂或腿)或者有关节的其他身体部位(例如躯干或手指)生成动画最为有用。例如,使用“游戏骨骼”变形可以使用骨骼连接单一绘图上的手臂,使上臂和前臂可以独立活动,而不必在不同图层上绘制上臂和前臂。**Harmony**将绘图变形,使其看起来有关节。游戏骨骼变形的不同部分可以围绕其关节旋转、伸展和缩短,让您具有与在不同图层上为关节生成动画相同的能力,而不必担心部件脱落、轴心点或剪贴轮廓。

游戏骨骼变形不同于骨骼变形,前者专为 **Unity** 等游戏引擎优化。因此,通常只用于游戏开发,而不用于动画制作。**S**骨骼和游戏骨骼变形的差异在于,游戏骨骼变形没有“偏差”属性“偏差”和“影响区域”属性。关节交叠看起来也更圆一些。



游戏切割器准则

Harmony里的切割器或遮罩使用自定义形状切割绘图。游戏引擎 SDK 使用切割器和方向切割器有以下限制：

- 对于精灵，渲染时只能应用单个切割器绘图。多个遮罩绘图的合成。游戏引擎 SDK 渲染时只使用第一个遮罩绘图，弃用其他绘图。
- 变形的绘图不能切割，但切割的绘图可以变形。



注意

目前在游戏引擎 SDK 执行 Cocos2d-x 时不可用。


创建元数据注释

在Harmony为游戏创建资产时，可能要对场景或角色特定部分或道具为程序员撰写注释。这些随资产导出到 Unity 的嵌入式注释称为元数据。

如何访问“元数据编辑器”视图

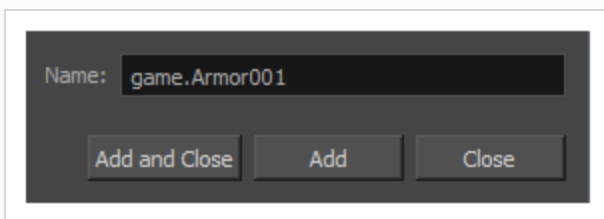
- 在视图的右上角单击 **Add View(添加视图)**  按钮，选择 **Metadata Editor(元数据编辑器)**。
- 在顶部菜单选择 **Windows(窗口) > Metadata Editor(元数据编辑器)**。

如何创建场景元数据

- 在元数据编辑器视图的 **Scene Metadata(场景元数据)** 栏，单击“加号”  按钮新建一个元数据条目。

此时会显示 **Add Metadata(添加元数据)** 对话框。

- 在 **Add Metadata(添加元数据)** 对话框，输入新场景条目的名称。



注意

对于在元数据编辑器内创建的锚点或道具信息，命名惯例是必须有“game.”前缀。Unity 会将“game.”视为元数据信息自动识别。例如 *game.Armor001*。

- 如果只想输入一个条目，请单击 **Add and Close(添加后关闭)**。如果要添加多个条目，请单击 **Add(添加)**。完成一个后，单击 **Add(添加)** 继续添加条目。添加完后单击 **Close(关闭)**。
- 在元数据编辑器视图双击第一个条目的值字段，使之变成可编辑。
- 输入此条目的值信息。
- 继续为所有条目添加值信息。

此信息会随 Harmony 资产导出。到 Unity 后，如果 **Hierarchy(层级)** 视图选定了此资产，场景元数据将在 **Inspector view > Metadata(元数据)** 中显示。

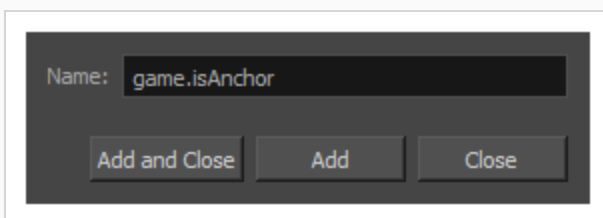
如何创建节点元数据

- 在“时间轴”视图中，选择要添加元数据的图层。

图层名称显示在 **Node Metadata(节点元数据)** 栏顶部附近。



2. 在 **Node Metadata**(节点元数据) 栏, 单击加号 **+** 按钮新建一个元数据条目。
此时会显示 **Add Metadata**(添加元数据) 对话框。
3. 在 **Add Metadata**(添加元数据) 对话框, 输入新节点条目的名称。



注意

对于在元数据编辑器内创建的锚点或道具信息, 命名惯例是必须有“game.”前缀。Unity 会将“game.”视为元数据信息自动识别。例如 *game.Armor001*。

4. 如果只想输入一个条目, 请单击 **Add and Close**(添加后关闭)。如果要添加多个条目, 请单击 **Add**(添加)。完成一个后, 单击 **Add**(添加) 继续添加条目。添加完后单击 **Close**(关闭)。
5. 在元数据编辑器视图双击第一个条目的值字段, 使之变成可编辑。
6. 输入此条目的值信息。
7. 继续为所有条目添加值信息。

此信息会随 **Harmony** 资产导出, 更准确地说, 此信息会与所选图层建立链接。到 **Unity** 后, 如果 **Hierarchy**(层级) 视图选定了此资产, 场景元数据将在 **Inspector view > Metadata**(元数据) 中显示。

游戏动画小贴士

制作游戏动画时,可能要根据游戏类型对动画加以限制。例如,如果是制作手机游戏,希望保持较小的文件,使之能在所有设备上快速播放,那么请将动画限制在简单的关键帧上,并尽可能减少绘图交换。如果制作主机游戏,则可以自由创建更多、更复杂的绘图。您可以与开发人员探讨导出的目标平台的限制程度以及游戏引擎的支持内容。

以下提供一些制作高效、轻量级动画的技巧:

- 着重使用变换,例如移动、旋转、缩放和倾斜。
- 必要时额外创建绘图交换。
- 若使用曲线和封包变形器或变型,需要烘焙绘图再导出。谨慎操作,因为绘图数量要尽可能少。不要烘焙整个镜头,只烘焙精选的绘图。无需烘焙游戏骨骼变形器。
- 绘图视图中的绘图越大,在 **Sprite Sheet** 的纹理尺寸上占据的像素就越多。设置绑定时,不要使用带变换工具的关键帧缩放单个图层。若要放大或缩减事物,请使用 **Select(选择)** 工具。这样事物在 **Sprite Sheet** 上能保持一样的相对尺寸。导出 **Sprite Sheet** 时,还可以在脚本中设置 **Sprite Sheet** 的分辨率,使绘图缩减,适用于小型设备。
- 只有在场景中曝光的绘图才会导出到 **Sprite Sheet**。例如,如果文件库视图中有 10 张绘图,但场景中只显示其中两张,则只导出这两张。这使 **Sprite Sheet** 尽量精简。

制作多个镜头的动画

您可能要为角色制作多个动画。例如, 空闲镜头、奔跑镜头、动作镜头等。需在特定结构中处理, 将所有动画导出到一个 **Sprite Sheet**中。

可以使用的工作流有两种:

- 工作流 1: 分离场景
- 工作流 2: 使用场景标记分离

workflow 1:分离场景

首先创建一个沿用角色名称的场景文件,例如 **Space Duck**。此文件中可以创建或导入游戏绑定。在顶部菜单选择 **File(文件) > Save As New Version(保存为新版本)**,并用动画名称为其命名。例如, **Idle**。

每次需要使用同一角色制作新动画时,都执行一次 **Save As New Version(保存为新版本)**。最后生成:

场景:太空鸭

版本:

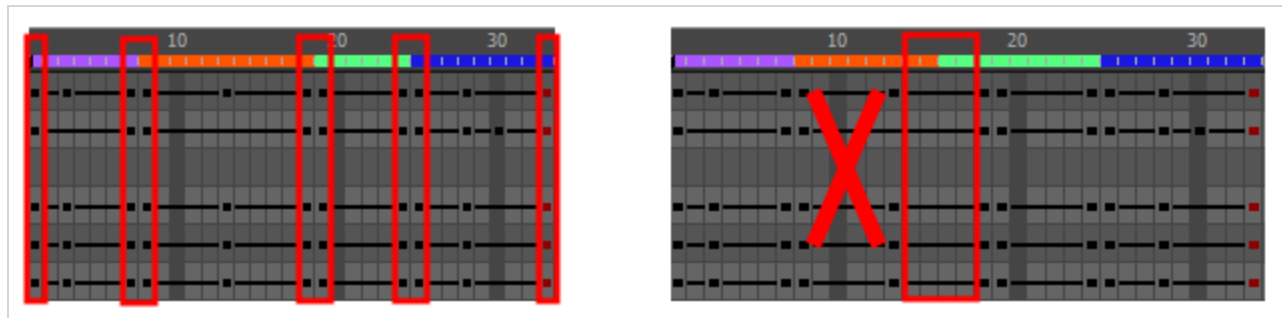
- 空闲
- 奔跑
- 跳跃
- 弹射

运行导出脚本时,会将当前场景的绘图导出至导出文件夹。还会提示其他导出到该文件夹的场景版本。如此,它将重新编译 **Sprite Sheet**,使之包含所有动画的所有绘图。这样可以最大化地重复使用绘图。

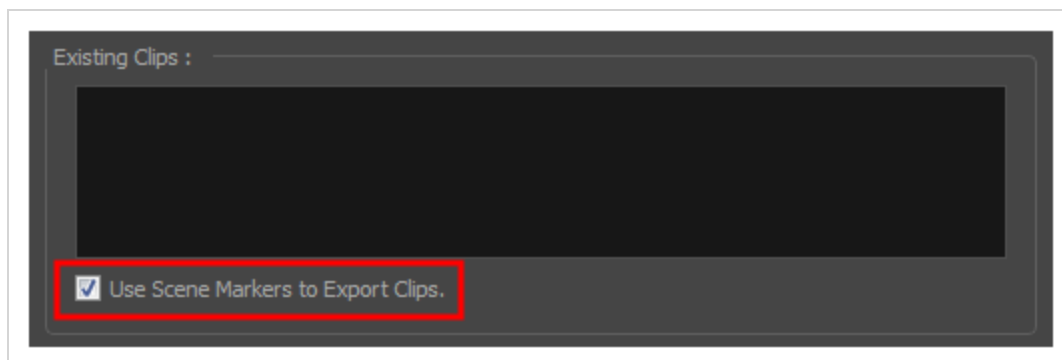
工作流 2: 使用场景标记分离

您还可以在一个场景中相继创建所有角色动画, 例如空闲、奔跑、跳跃和弹射。然后使用场景标记标记并分隔单独动画。

标记单个动画帧范围时, 务必关键帧开始, 关键帧结束。不要对起始或结束于插值运动中间的帧范围创建场景标记。



导出 Sprite Sheet 时, 务必在 Export To Sprite Sheet(导出至 Sprite Sheet) 对话框中勾选 **Use Scene Markers to Export Clips**(对导出剪辑使用场景标记) 选项。默认启用。



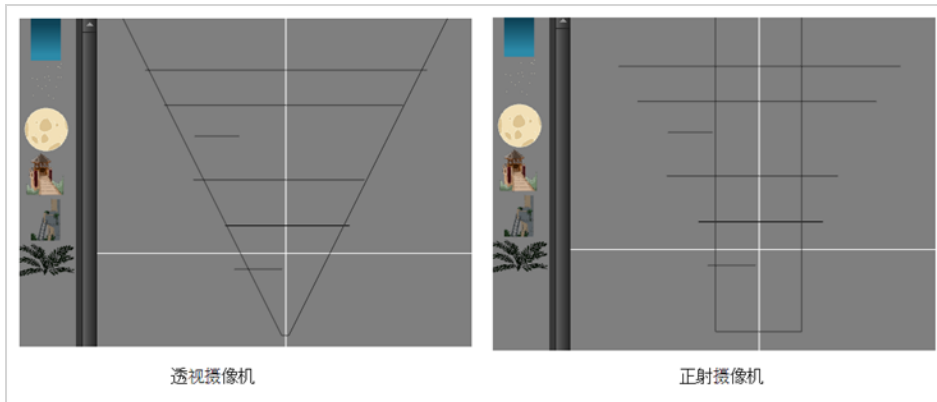
动画化剪辑在 `stage.xml` 中的分割和列出方式, 相同于将每个动画镜头从分隔场景导出到同个文件位置时的显示方式。

关于正射摄像机

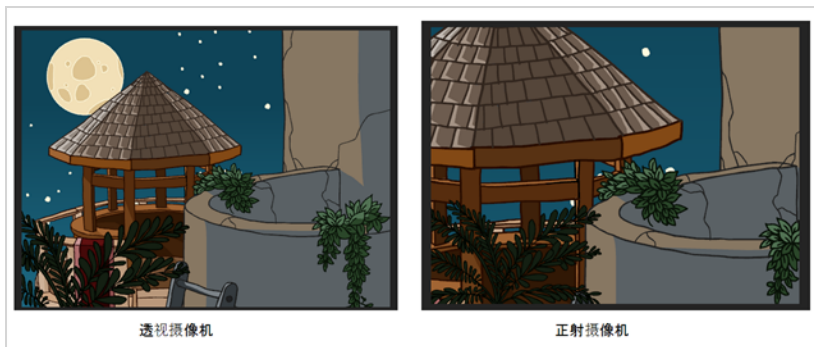
Harmony内有两种摄像机可用：

- 透视
- 正射

正射摄像机专用于游戏工作流程。摄像机类型从透视改为正射。变成没有灭点的摄像机。也就是说“摄像机”视图内没有透视。对象在 Z 轴上移动时大小或缩放不变。

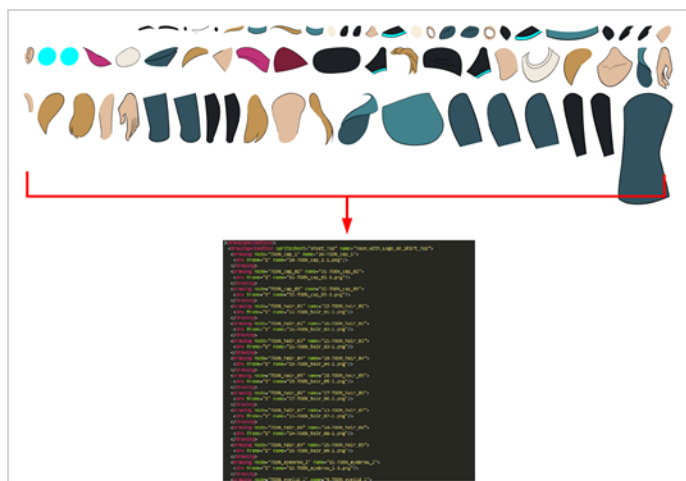


正射摄像机可在 **Scene Settings**(场景设置) 对话框。若要默认使用正射摄像机创建场景, 则必须创建新的自定义场景分辨率。



第 3 章：关于导出到 Unity

如果您有自定义游戏引擎，可以使用导出的 Harmony 数据再继续完成常规流程。也可以修改导出脚本，使之符合惯例。



导出场景前：

- 务必将“显示”设置为 **Display(显示)** 而非 **Display All(全部显示)**。
- 务必保存场景。Harmony 基于 tvg 文件导出，添加的未保存更新都不会导出。
- 将 Harmony 场景设置为正方形分辨率(如 1024 x 1024)。选择 **Scene(场景) > Scene Settings(场景设置)**。

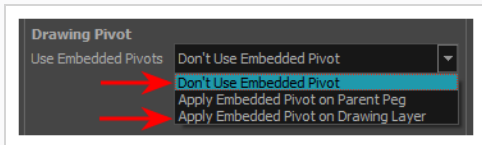
设置锚点

导出到 **Unity** 之前, 应该考虑将什么设置为锚点。

锚点是指角色动画创作期间要在游戏引擎中引用的角色上的一个点。

如何设置锚点

1. 新建一个绘图图层, 按照要创建的锚点对其重命名。
2. 在绘图图层的 **Layer Properties**(图层属性) 内, 确保轴心点选项已设为 **Don't Use Embedded Pivot**(不使用内嵌轴心点) 或 **Apply Embedded Pivot on Drawing Layer**(应用内嵌轴心点于绘图图层)。“应用嵌入轴心点于父定位钉”无法很好地过渡到 **Unity**, 会丢失轴心点信息。



3. 完成此操作后, 可以使用动画工具重新定位图层的轴心点, 例如“高级动画”工具栏中的“旋转”工具。绘图图层的图层属性中应当可以看到轴心点坐标。如果不重新定位轴心点, 则默认位于 **Harmony** 场景中心 (0,0)。
4. 设置好后, 选择 **Drawing layer**(绘图图层) 并单击 **Game**(游戏) 工具栏里的 **Toggle Anchor**(切换锚点) 按钮。还可以多选, 同时在多个绘图图层上设置锚点。

绘图图层将在“时间轴”和“节点”视图以红色高亮显示。现在可以将锚点作为道具使用了。

导出 Sprite Sheet

“导出至 Sprite Sheet”窗口导出多种分辨率，会生成多个 .xml 文件和一个或多个 Sprite Sheet，具体取决于定义了几种精灵分辨率。。

此操作会将同一角色的不同动画以相同名称保存。例如，如果有空闲、奔跑和跳跃动画，它们共享同一“资产名称”。可将其视为动画的整体集合。集合内包含不同的已保存场景版本，其绘图将重复应用于该角色集的所有动画。每个场景版本将显示为列表的一项。

导出动画时，只导出该场景使用的绘图。所有的绘图都是先单独导出，然后再集成为一个 Sprite Sheet。

若将多个动画保存为同一资产名称(如 SpaceDuck: run, idle)，它会重新集合 Sprite Sheet，使之包含该文件夹中所有动画的所有绘图，新建一个动画文件，但重复使用同一骨骼。



注意

令程序员欣喜的是，Harmony 导出的 XML 数据可以在之后转化为更优化的二进制数据结构。要将 XML 转化为二进制格式，请使用 Xml2Bin 实用程序。此实用程序将 Toon Boom Harmony 软件生成的 XML 数据结构转化为压缩的二进制数据结构。此实用程序位于 /HarmonyGameSDK/Plugins/ 下的游戏 SDK。

- Plugins/Mac: Mac OSX 的预编译二进制。
- Plugins/Windows\x86: Windows 的预编译二进制。
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin: Xml2Bin 源。
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin/proj.mac.Xml2Bin.xcodeproj: Mac OSX 的 Xcode 项目。
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin/proj.win32/Xml2Bin.sln: Visual Studio 2010 solution for Windows.

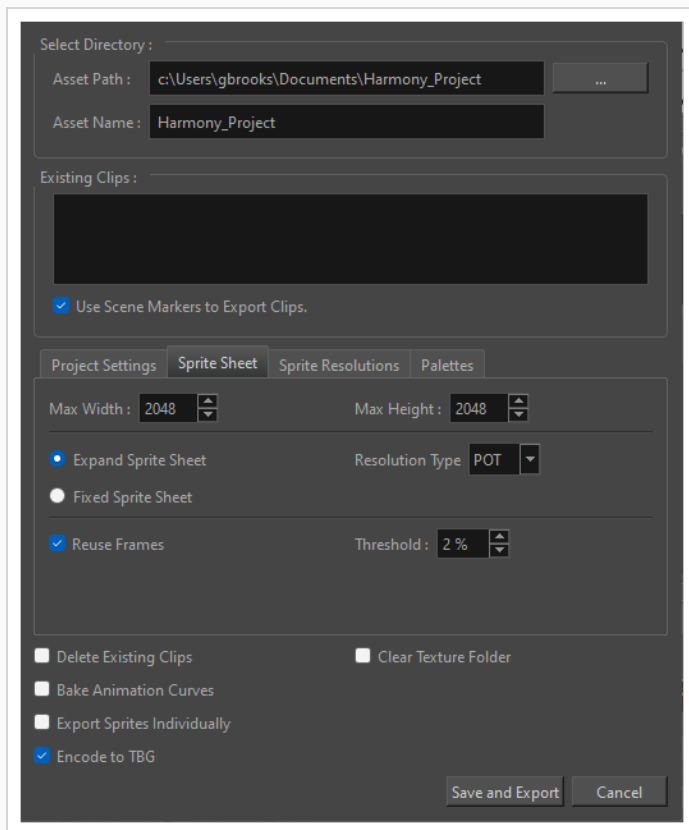
若要集成其他游戏引擎，可在自己的代码中重用和解析处理数据结构的 C++ 代码。

如何导出 Sprite Sheet

1. 选择 **Windows(窗口) > Toolbars(工具栏) > Game(游戏)**，添加“游戏”工具栏。



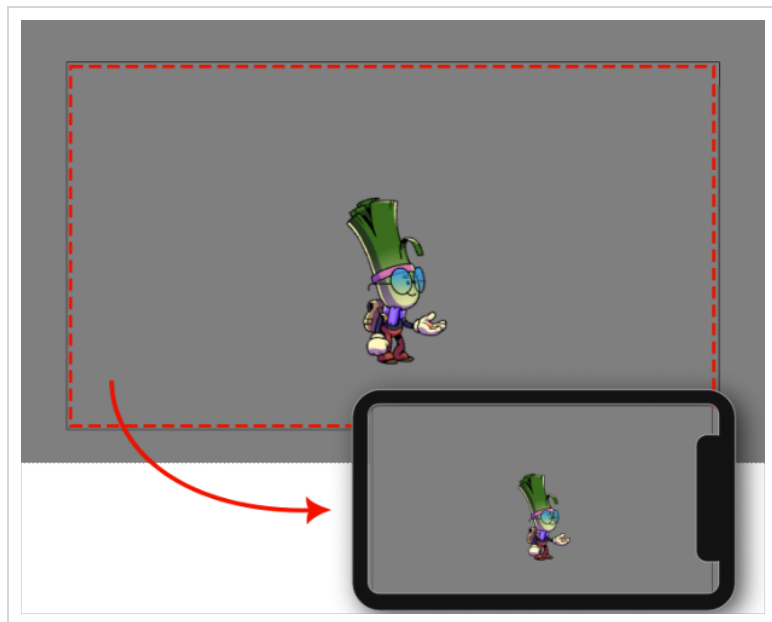
2. 在 **Game(游戏)** 工具栏内单击 **Export to Sprite Sheets(导出至 Sprite Sheet)** 按钮运行脚本。
Export to Sprite Sheets(导出至 Sprite Sheet) 窗口将打开。



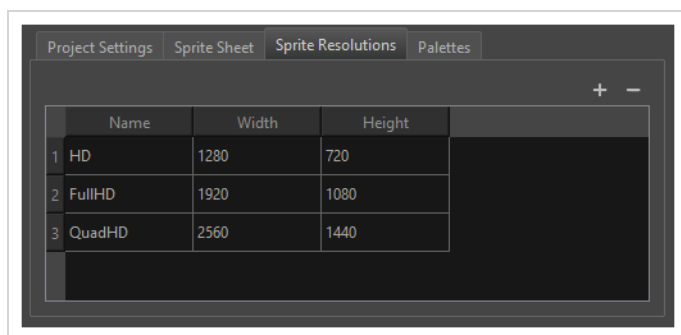
3. 如果要自动更新, 请将 **Save Path**(保存路径) 设置为 **Unity** 项目中正确的文件夹。如果不要自动更新, 可以保存在任意位置, 然后将动画传输到 **Unity** 项目或要使用的其他游戏引擎。
4. 设置 **Sprite Sheet** 首选项。
5. 单击“**Export**”(导出) 。

精灵分辨率

您可以根据游戏的目标分辨率设置精灵的像素数。目标分辨率可基于电视或手机屏幕等目标硬件。



对于“导出至 Sprite Sheet(XML)”窗口的“分辨率”选项卡中指定的每个目标分辨率，都会新建一个 Sprite Sheet。访问此窗口前需启用“游戏”工具栏：选择 **Windows(窗口) > Toolbars(工具栏) > Game(游戏)**。



请注意，精灵的大小就是角色创建于 Harmony 时的大小。

如何设置多种精灵分辨率

1. 启用“游戏”工具栏：选择 **Windows(窗口) > Toolbars(工具栏) > Game(游戏)**。
2. 选择 **Export to Sprite Sheets(导出至 Sprite Sheet)** 。此时会打开“导出至 Sprite Sheet(XML)”窗口。
3. 在“导出至 Sprite Sheet(XML)”对话框，选择 **Sprite Resolutions(精灵分辨率)** 选项卡。此选项卡会列出所有当前目标分辨率。
4. 单击 **Add(添加)** 图标。此操作将为新的目标分辨率创建新行。
5. 输入新目标分辨率的 **Name(名称)**、**Width(宽度)** 和 **Height(高度)**。

默认目标分辨率是 **HD**、**FullHD** 和 **QuadHD**。列出的屏幕分辨率表示可以玩游戏的目标设备。

您完全可以根据目标分辨率和硬件在该列表手动添加或删除分辨率。导入到游戏引擎后, 可以更改渲染的 **Sprite Sheet** 选择要显示的分辨率。



提示

- 精灵分辨率受绘图笔触的缩放影响。
- 缩放绘图或其定位钉不影响输出精灵分辨率。
- 若要增大精灵分辨率, 可缩放绘图的子图层。

使用 Bake_Group

Bake_Group 是一组绘图和特效节点，可以作为 Sprite Sheet 的统一绘图一起导出。

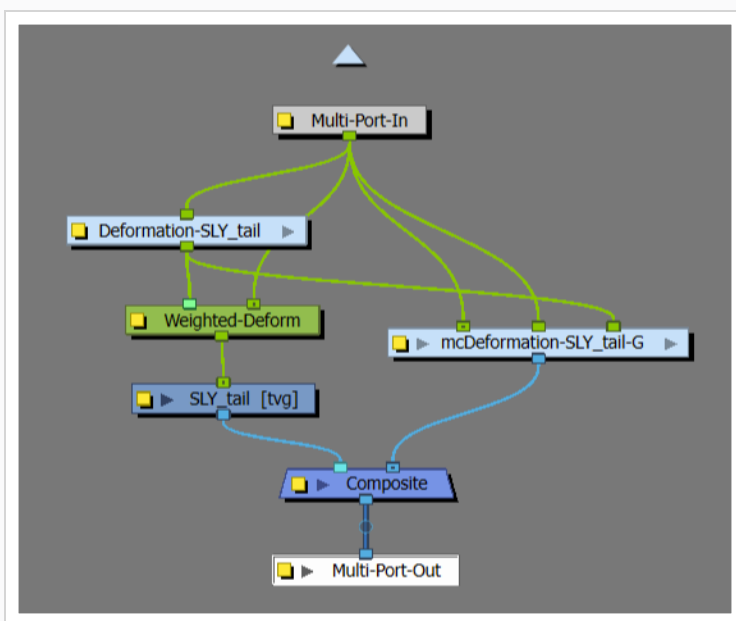
如何创建烘焙组

1. 将要一起烘焙的图层组成一组。建议从节点视图将图层分组。。



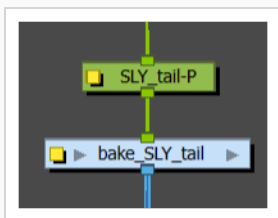
注意

Bake_group 应只有一个输入和一个输出。意味着 **bake_group** 内必须包含一个复合节点，导出才能正常工作。还要考虑需要哪些定位钉指导绘图，因为所有定位钉最终都源自组外的单个定位钉。



2. 在时间轴内，组名前添加“bake_”前缀。

查看下面示例中含“bake_”前缀的组节点样式。标记为“hips”的“group”名称应为“bake_hips”，方能应用烘焙。



若组名包含“bake_”前缀，将分析该组包含特效、移动和绘图替换的帧，每个特殊帧都将导出成为 Sprite Sheet 的精灵。

对于这些生成的精灵，将会创建新的绘图动画，并提供给游戏引擎。游戏引擎无法访问组内的结构，而是将整个组视为包含绘图替换的单个绘图。

导出至 Easel JS

“Export to Easel JS”(导出至 Easel JS) 窗口可以拼合动画的图像镜头。即使有一个完全绑定的木偶或包含动画镜头的单个绘图图层, 仍然会输出各帧的拼合, 在 **Sprite Sheet** 内形成一组。这样, 您可以使用任意工具或特效模块, 灵活性和创作自由度更高。但可能会使文件更大, 具体取决于动画的长度、复杂性和导出大小。

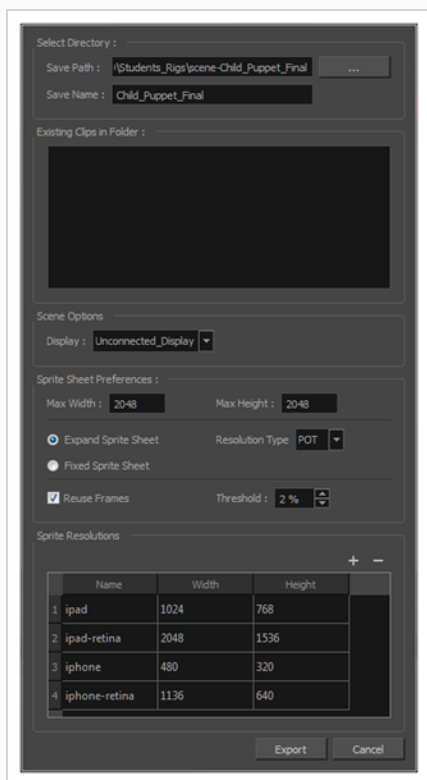
如何导出至 Easel JS

1. 选择 **Windows(窗口) > Toolbars(工具栏) > Game(游戏)**, 添加“游戏”工具栏。



2. 在 **Game(游戏)** 工具栏内单击 **Export to Easel JS(导出至 Easel JS)** 按钮运行脚本。

Export to Easel JS(导出至 Easel JS) 窗口将打开。



3. 如果要自动更新, 请将 **Save Path(保存路径)** 设置为 **Unity** 项目中正确的文件夹。如果不要自动更新, 可以保存在任意位置, 然后将动画传输到 **Unity** 项目或要使用的其他游戏引擎。
4. 设置 **Sprite Sheet** 首选项。
5. 单击 **Export(导出)**。

色板变体

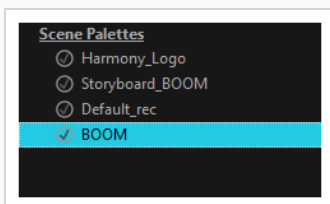
若计划将角色自定义添加为游戏的功能，需创建有色板差异的不同角色版本。在 Harmony 中，可以为一个角色创建多个色板，使其在游戏中呈现不同的外观。

在 Harmony 中创建了角色及其色板后，就可以为每个色板将角色导出到 Sprite Sheet。



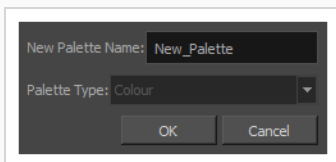
如何为游戏角色创建色板变体

1. 在“颜色”视图中，选择一个调色板进行克隆。

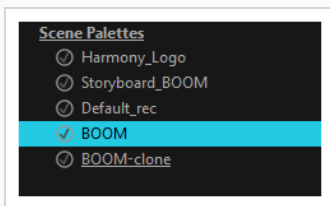


2. 执行以下其中一项操作：从 Colour(颜色) 菜单，选择 **Palettes(色板) > Clone(克隆)** 或右键单击选择 **Clone(克隆)**。
 - 右键单击色板后选择 **Clone(克隆)**。
 - 打开 Colour(颜色) 菜单 , 然后选择 **Palettes(色板) > Clone(克隆)**。

此时将打开 Clone Palette(克隆色板) 对话框



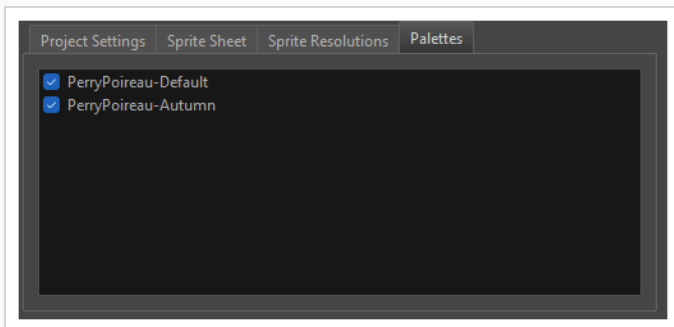
3. 在 **New Palette Name(新色板名称)** 字段中，输入新色板的名称。色板名称默认是原色板名称加上“-clone”。此字段上方的色板列表列出选定位置中已存在的色板。确保输入的名称没有其他色板使用。
4. 单击 **OK(确定)**。
克隆的色板显示在色板列表中。



所有场景色板都会显示在 **Export to Sprite Sheet**(导出至 **Sprite Sheet**) 窗口的 **Palettes**(色板) 选项卡中。

如何为每个色板变体导出一个 **Sprite Sheet**。

1. 执行以下操作之一启用“游戏”视图：
 - 在顶部菜单选择 **Windows**(窗口) > **Toolbars**(工具栏) > **Game**(游戏)。
 - 右键单击任意工具栏后选择 **Game**(游戏)。
2. 在 **Game**(游戏) 工具栏选择 **Export to Sprite Sheets**(导出至 **Sprite Sheet**) .
3. 打开 **Palette**(色板) 选项卡。此时能看到所有项目的色板及其克隆色板。



注意

只显示色板及克隆色板。色板副本不显示。

4. 选择要导出的色板的名称。选中的每个色板及其角色将作为自己的 **Sprite Sheet** 一起导出。
5. 导出 **Sprite Sheet**。

角色导入到 **Unity** 后, 可以更改渲染的 **Sprite Sheet** 选择要显示的色板。

第 4 章：关于 Harmony Unity SDK

Toon Boom 在 Unity Assets store 提供一个 Unity 资源包，包含导入从 Harmony 导出的数据所需的脚本。解压到 Unity 新项目后，Toon Boom Harmony Gaming SDK 文件夹结构内含：

- 程序说明(开发人员为主的 Unity 集成信息)
- 插件(SDK 文件库)
- 样本(演示场景、导出器脚本、SDK 源以及一个预览程序)
- 运行时间(游戏播放时执行的 Harmony 脚本)
- 编辑器(仅在编辑器运行的 Harmony 脚本，不出现在 build 内)
- Shader(提供正确渲染 Harmony 特定功能的 shader)
- 材质(使用 Harmony shader 的预制材质)

这些资源包可通过 Unity 的 Package Manager(包管理器)界面手动管理。如果更新的 Unity 版本与 Harmony 脚本的包不兼容，开发人员可对资源包升级和降级。

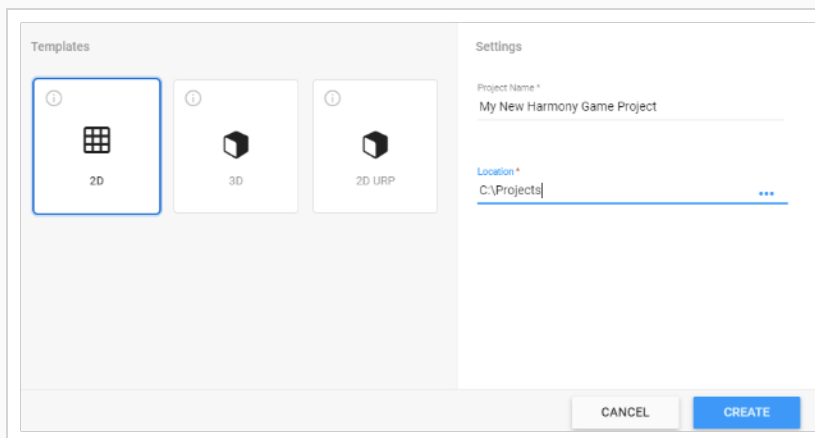
如何创建一个 Unity SDK 项目

1. 创建一个空的 Unity 游戏项目。建议您创建一个空的 2D Unity 游戏项目，这种项目最便于处理 2D 渲染和物理。



注意

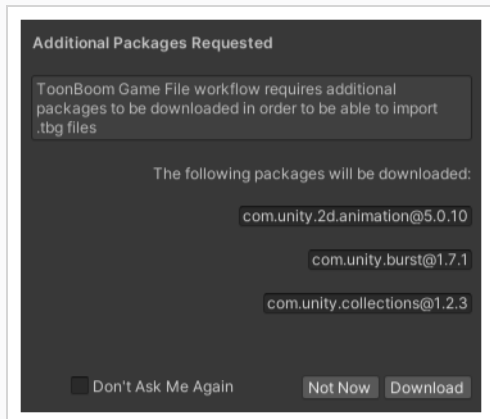
务必使用的 Unity 2020.3.36 或更高版本，此前版本的 Unity 不具备支持导入和渲染 TBG 文件的必要功能。



2. 在 Unity Asset Store 安装 Harmony Game SDK。如果 SDK 是手动下载的，可以在 Unity 项目打开时双击 .unitypackage 文件。

此时会弹出 Additional Packages Requested(已请求附加包)窗口，确保您拥有运行时优化 2D 动画的兼容 Unity 包。

3. 单击 **Download(下载)**。



Unity 下载完所需的包并编译完所有脚本后，项目就可以开始导入 Harmony 角色了。

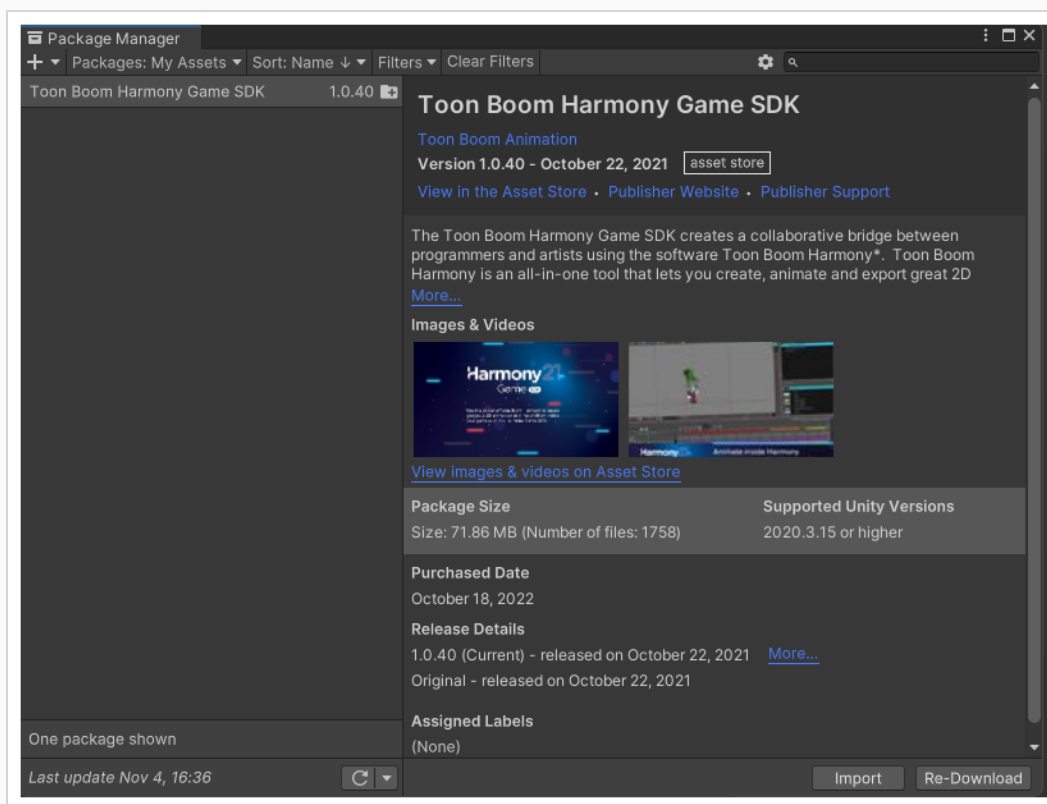


提示

安装 Toon Boom Harmony Game SDK 后，必须将 Harmony SDK 包导入到每个在 Harmony 中新建的项目。

如何将包导入项目

1. 选择 Window(窗口) > Asset Store, 打开 Asset Store。此时会打开 Asset Store 选项卡。
2. 在 Asset Store 中选择 **Open Package Manager(打开包管理器)**。此时会打开 Package Manager (包管理器) 窗口。



3. 包管理器左上角的包下拉菜单有四个选项：

- Packages: Unity Registry(包：Unity 注册表)
- Packages: In Project(包：项目内)
- Packages: My Assets(包：我的资产)
- Packages: Built-in(包：内置)

4. 选择 **Packages: My Assets**(包：我的资产)。

下载的资产将显示在下方列表中。Package Manager(包管理器)选项卡右侧会显示所选资产的信息。

5. 选定 Harmony 包后，选择包管理器底部的 **Import**(导入)。此时会出现显示资产文件的 **Import Unity Package**(导入 Unity 包)窗口。



6. 选择 **Import(导入)**。

Harmony Previewer

本节介绍如何在未安装 Unity 的情况下使用 Harmony Previewer 查看动画。

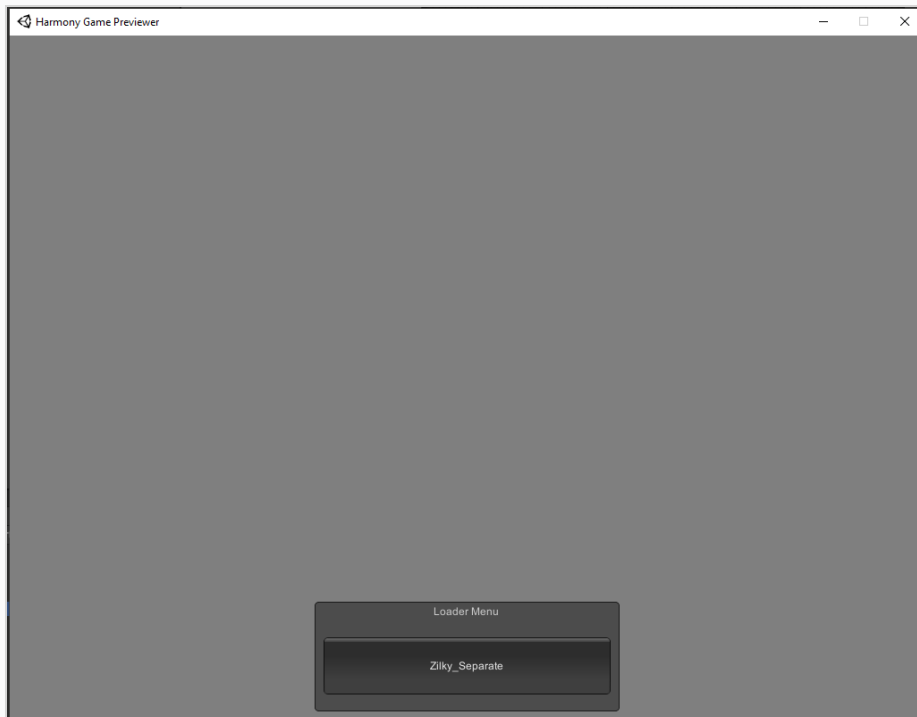
Harmony Game Previewer 内包含 Windows 或 macOS 的 build，可以直接解压缩。

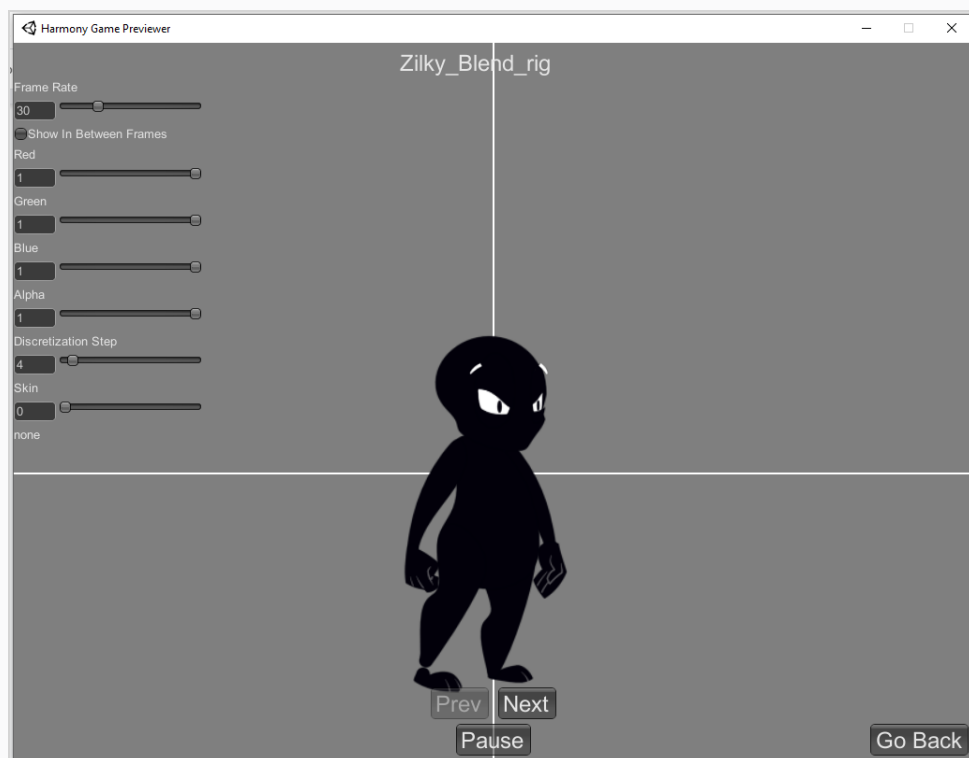


注意

请勿解压 Unity 项目内的 build，建议在其自身文件夹内解压缩。

如何使用 Previewer





1. 运行 *Harmony Game Previewer.exe* 或 *Harmony Game Previewer.app*
2. 从“加载器”菜单选择要加载的项目。这些项目从 **Harmony Resources** 文件夹加载而来, 其目录与 *Harmony Game Previewer.exe* 或 *Harmony Game Previewer.app* 相同。
3. 使用 **Prev**(上一个) / **Next**(下一个) 按钮导航动画。
4. 使用左上角的控件调整“帧率”、“色调”、“Discretization Step”和“蒙皮”设置。
5. 单击 **Go back**(返回) 按钮回到上一菜单, 加载另一动画。

资产操控

本节将介绍 Unity 中的 Harmony Mesh 和 Harmony Renderer 组件。

TBG 工作流

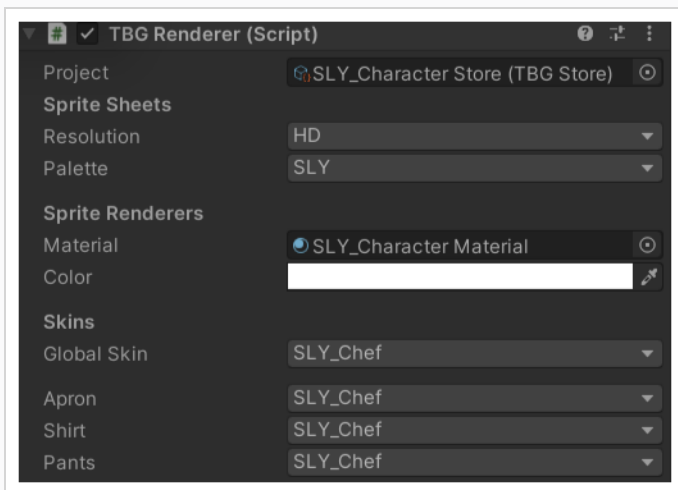
若通过 TBG 工作流将精灵导入 Unity, TBG 文件的资产操控选项将显示在 Inspector View 中。这里显示的资产操控选项与 XML 工作流中显示的不同。

TBG Asset Renderer

TBG 文件工作流的 Harmony Renderer 控制 TBG 文件的 Harmony 资产的动画与渲染, 并显示运行时可能要更改的重要变量。

如何操控 TBG 工作流渲染资产

1. 在场景中选择一个角色。TBG Renderer(脚本) 栏会出现在右侧的 Inspector(检查器) 选项卡中。



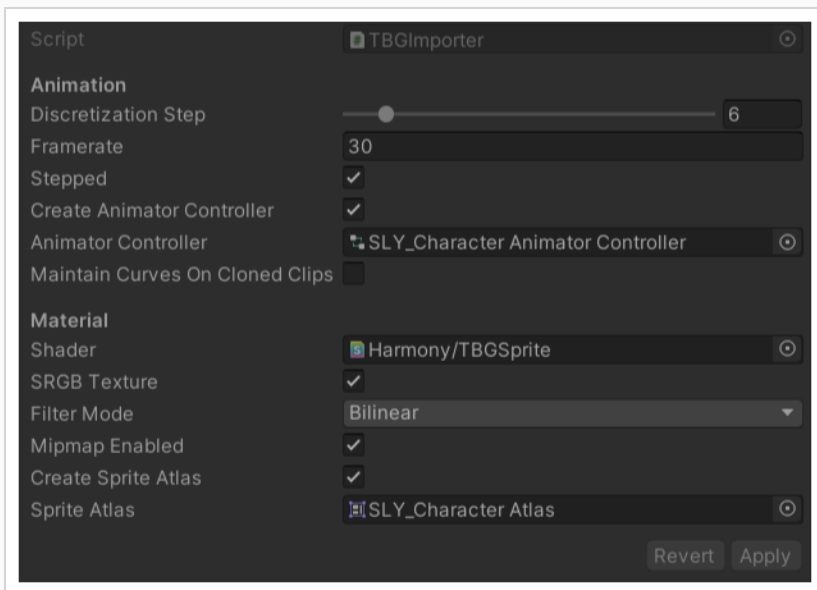
2. 控制角色资产需设置以下项目：
 - **Project(项目)** : TBGStore 资产的引用。此资产由导入器生成, 包含所生成 AnimationClip 未涵盖的 TBG 项目数据。
 - **Resolution(分辨率)** : 展示此角色的分辨率。此项匹配相关的 Sprite Sheet 名称。
 - **Palette(色板)** : 展示此角色的色板。此项匹配相关的 Sprite Sheet 名称。
 - **Material(材质)** : 产生角色阴影的材质。SDK 随附硬件蒙皮, 为不发光材质。
 - **Color(颜色)** : 对资产叠加选定颜色的选项。
 - **Skins(蒙皮)** : 允许您实时更改从 Harmony 导出的蒙皮。每组都有更改蒙皮的下拉列表, 全局下拉列表则可一次性更改所有组的蒙皮。

TBG 导入器

TBG 文件导入到 Unity 时, Harmony 资产将转换为 Unity prefab、AnimationClip 和其他资产。TBG 导入器可指定资产和动画的创建方式。

如何操控 Harmony TBG 资产导入 Unity 的设置

1. 在 Project(项目) 视图选择一个 .tbg 角色。此时会打开 inspector(检查器) 选项卡。
2. 在 inspector(检查器) 视图顶部单击模型的 **Select(选择)**。此时会打开 TBG Importer(TBG 导入器) 视图, 显示 TBG 导入器选项。



3. 设置以下内容控制动画导入的方式:

- **动画设置**

- **Discretization Steps:** 一个表示精灵在进行骨骼变形时分割成的列数/行数的滑块
- **Frame Rate:** Unity 的动画系统播放 Harmony 帧的动画速率。此值应于 Harmony 帧率相同。
- **Stepped:** 若选中, 动画将在整数帧间移动, 不会平滑插值。若未选中, 动画将在整数帧间播放, 以获得流畅的动画效果。
- **Create Animator Controller:** 若勾选, 此导入器将自动新建包含所有已生成 AnimationClip 的 Animator Controller, 并在 Project(项目) 视图将资产放置在 prefab 旁边。
- **Animator Controller:** 更改 .tbg 文件时, 引用的 Animator Controller 将更新的 AnimationClips。
- **Maintain Curves on Cloned Clips:** 若选中, 从 .tbg 文件中的剪辑 sub-asset 复制(并在 AnimatorController 中引用)的曲线将被更新的 .tbg 文件的新数据覆盖。

- **材质设置**

- **Shader:** 创建 prefab 后, 所有 SpriteRenderer 都将引用一种使用此 Shader 生成的材质。
- **SRGB Texture:** 若选中, 所生成纹理的伽玛颜色空间将映射线性颜色空间渲染。
- **Filter Mode:** 纹理的相邻像素如何插值。
- **Mipmap Enabled:** 生成较低分辨率的纹理, 增加渲染距离。
- **Create Sprite Atlas:** 若勾选, 此导入器将自动新建包含所有已生成精灵的 Sprite Atlas, 并在 Project(项目) 视图将资产放置在 prefab 旁边。
- **Sprite Atlas:** 变更 .tbg 文件后, Sprite Atlas 引用将更新精灵。

XML 工作流

若通过 XML 工作流将精灵导入 Unity, TBG 文件的资产操控选项将显示在 **Inspector View** 中。这里显示的资产操控选项与 TBG 工作流中显示的不同。

Harmony Mesh

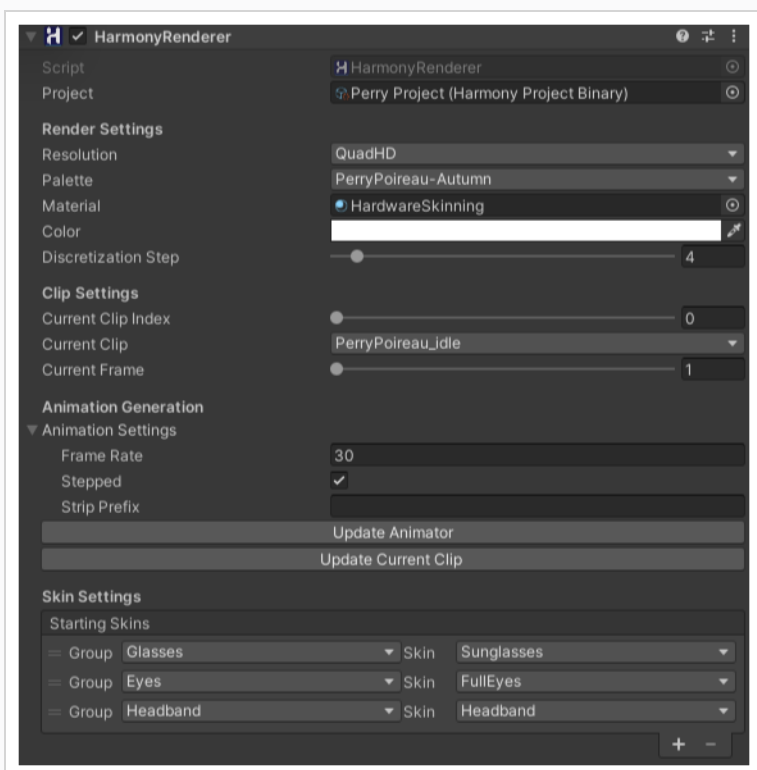
Harmony Renderer 通过 Harmony Mesh 组件, 基于您的动画曝光的绘图创建资产网格。

XML Asset Renderer

XML 工作流的 Harmony Renderer 控制 XML 文件的 Harmony 资产的动画与渲染, 并显示运行时可能要更改的重要变量。

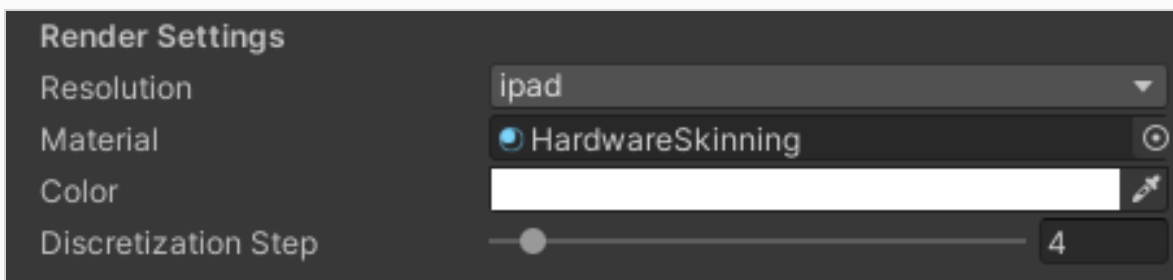
如何操控 XML 工作流渲染资产

1. 在场景选择一个角色
2. 在 **Inspector View** 选择底部的 **Add Component(添加组件)**。此时会显示 **HarmonyRenderer** 栏。



3. 控制角色资产需设置以下项：

- **Project(项目)** : Harmony 项目二进制资产的引用。此资产由导入器生成, 包含有效二进制格式的 Harmony 项目数据。
- **Render Settings(渲染设置)** : 渲染设置用于修改资产, 可通过更改分辨率或用于渲染的 Shader, 或更改变形质量。

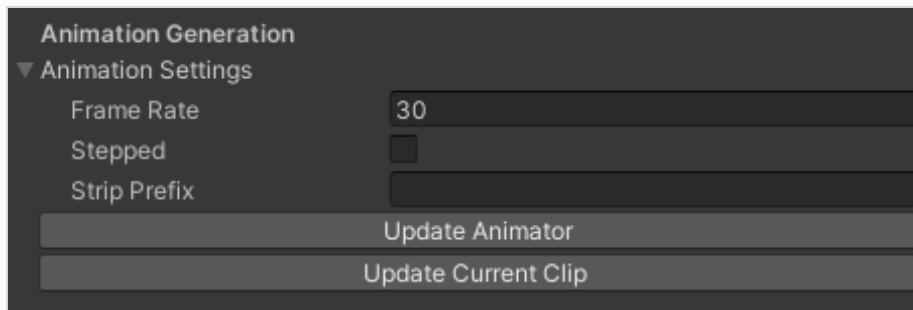


- **Resolution(分辨率)** : 展示此角色的分辨率。此项匹配相关的 Sprite Sheet 名称。
- **Palette(色板)** : 展示此角色的色板。此项匹配相关的 Sprite Sheet 名称。
- **Material(材料)** : 产生角色阴影的材料。SDK 随附硬件蒙皮, 为不发光材质。
- **Color(颜色)** : 对资产叠加选定颜色的选项。
- **Discretization Steps** : 定义所有变形绘图的排列的滑块。

- **Clip Settings(剪辑设置)**：剪辑设置在不打开动画窗口的情况下查看动画。此项仅在 Editor(编辑器) 模式影响项目，播放游戏时，动画器会否决剪辑设置。



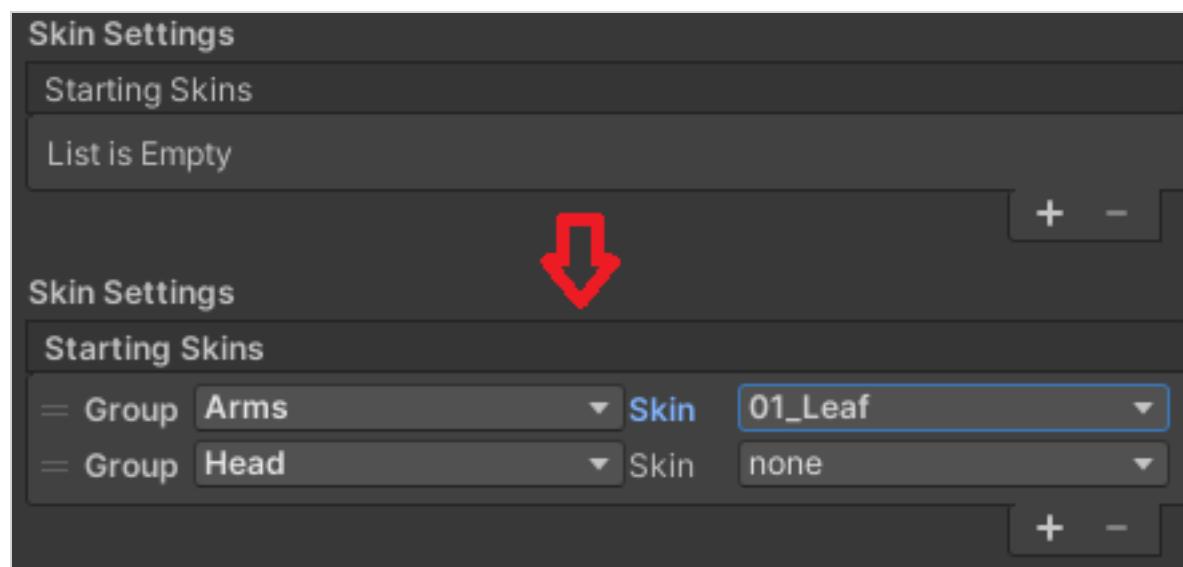
- **Current Clip Index(当前剪辑索引)**：显示的数字剪辑索引。可在动画窗口中动画化。
- **Current Clip(当前剪辑)**：选择要显示哪个动画的下拉列表。与当前剪辑索引对应。
- **Current Frame(当前帧)**：当前显示的帧号。
- **Animation Settings(动画设置)**：与导入资产设置相同。导入或重新导入修改资产到 Harmony 时此项很有用。完成修改后，单击 **Update Animator(更新动画器)** 或 **Update Current Clip(更新当前剪辑)** 保存修改。



- **Frame Rate(帧率)**：Unity 的动画系统播放 Harmony 帧的动画速率。此值应于 Harmony 帧率相同。
- **Stepped**：若选中，动画将在整数帧间移动，不会平滑插值。若未选中，动画将在整数帧间播放，以获得流畅的动画效果。
- **Strip Prefix(去除前缀)**：用于删除动画名称的前缀。例如，想将“Wishling_Idle”更改为“Idle”，将“Wishling_”设置为前缀。游戏的所有角色动画名称相同时，此项很有用。
- **Update Animator(更新动画器)**：单击此按钮更新角色的整套动画，与 Harmony 项目的内容匹配。此项将对现有动画更新您输入的新设置，并补充缺失的设置。
- **Update Current Clip(更新当前剪辑)**：仅更新当前选定的剪辑。

蒙皮设置

蒙皮设置可实时更改从 Harmony 导出的蒙皮。更改蒙皮前要先按下 + 按钮新增一组。需要在 Harmony 导出中添加您拥有的所有组。如果没有分组，只需添加一组并保留。您可以定义要修改哪个组以及将哪个皮肤分配给哪个组。可以选择组后按 - 按钮删除该组。



如何在 XML 工作流中更改蒙皮

1. 单击 + 按钮添加一组/多组。
2. 修改分组, 并分配蒙皮到该组。
3. 选择任意组后单击 - 按钮删除该组。

关于 Unity 示例项目

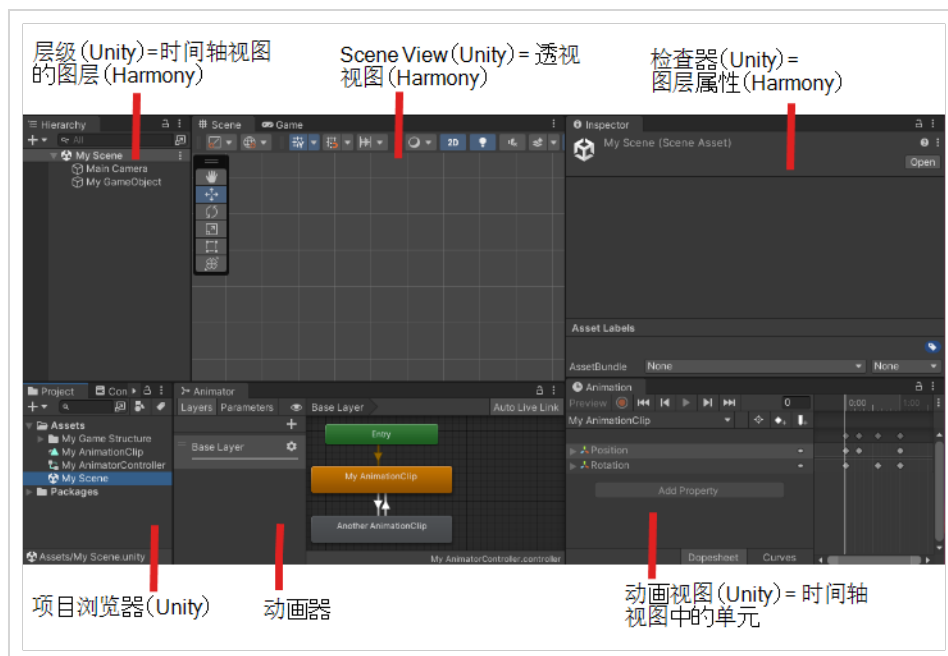
Toon Boom有一个 Unity 示例项目，其中包含将 Harmony导出的数据导入的脚本。项目内的 **Asset**(资产) 文件夹包含以下文件夹：

- **Plugins**
- **Scenes**(演示场景和一个预览程序)
- **Scripts**(所有Harmony脚本)
- **StreamingAssets**(所有Harmony场景文件导入的位置)

配置Harmony脚本后，可自动导出动画到 **StreamingAssets** 文件夹。如此，Unity 将动态加载出现在文件夹的最新资产。您也可以手动将导出的Harmony数据放入此位置。

关于 Unity 界面

下列为 Unity 界面的主要组件及其Harmony对应组件：



Unity	Harmony	描述
Scene View	透视图	设置场景的位置，其中可以选择和定位环境、玩家、摄像机、对手及其他所有 GameObject 。
游戏视图	摄像机视图	游戏中摄像机的渲染视图。展示最终发布的游戏。
检查器	图层属性	显示所选 GameObject 的详细信息，包括所有附加组件及其属性。
层级	时间轴视图的图层	显示场景中元素的层级，可为不同的游戏对象设置父子级关系。
动画	时间轴视图的单元	显示每个动画化变换的动画关键帧。
动画师		显示状态和过渡图，可设置角色如何回应游戏事件和玩家操作
项目浏览器	---	可访问和管理项目的资产。

以下为 Unity 交互的一些数据类型，及其在 Harmony 的对应数据：

Unity	Harmony	描述
精灵	绘图	可在 <code>SpriteRenderer</code> 上显示的单个图像。
<code>SpriteRenderer</code>	图层	一次显示一个精灵。动画期间可交换精灵，解释角色身体部位的运动。
变换	定位钉	构成角色身体的层级。缩放、旋转或位移，解释角色身体部位的运动。倾斜需从缩放和旋转的层级衍生。
<code>GameObject</code>		始终匹配一个变换。 <code>GameObject</code> 托管显示和模拟游戏的组件。SDK 提供运行于 <code>GameObject</code> 的自定义行为，显示 Harmony 特定的视觉特效。
资产		Unity 项目中视效、物理、游戏和音效的核心数据。它们一般是文件，也可以作为较大文件中的 <code>sub-asset</code> 托管。纹理、 <code>AnimationClip</code> 、精灵、 <code>AnimatorController</code> 、 <code>AudioClip</code> 和 <code>Prefab</code> 都属于资产。

Harmony文件导入 Unity

在Harmony完成图稿和可循环动画角色动作后，应将其导入 Unity 进行游戏整合。

如何将Harmony文件直接导入 Unity

1. 创建一个 Unity 2D 项目。
2. SDK Harmony项目需导入项目才能导入资产。请执行以下操作之一：
 - 转到 Unity Asset Store (assetstore.unity3d.com/en/#!/content/31211)，在计算机上保存 Harmony Game SDK 文件。然后导入到您的项目：**Top Menu(顶部菜单) > Asset(资产) > Import Package(导入包) > Custom Package(自定义包)**。
 - 在 Unity Asset Store 搜索 Harmony Game SDK，然后下载并导入包。Asset Store 可从这里访问：**Top Menu(顶部菜单) > Window(窗口) > Asset Store(资产商店)**。



注意

每个新的 Unity 项目需重复步骤 2，因为其中包含使Harmony导入过程正常运行所需的文件和文件夹。

从 Harmony 向 Unity 导出和导入数据，主要通过两种工作流

- **TBG 文件/TBGRenderer**

如有以下情况，适合这种工作流：

- 只使用 Unity，希望获得原生 Unity 体验。
- 希望更快导入，减少步骤。
- 希望生成浅显易懂的变换层级和可微调、使用简便的 **SpriteRenderer**。有少数或没有脚本编辑经验的人也能为角色添加程序动画特效。
- 希望利用 Unity 集合了 **sprite atlas** 和 **animator controller** 与图层和过渡混合的精灵网格。这样将充分通过已有资源包或简单脚本创作反向运动学、抖动物理对象、布娃娃。

- **XML 文件夹/HarmonyRenderer**

如有以下情况，适合这种工作流：

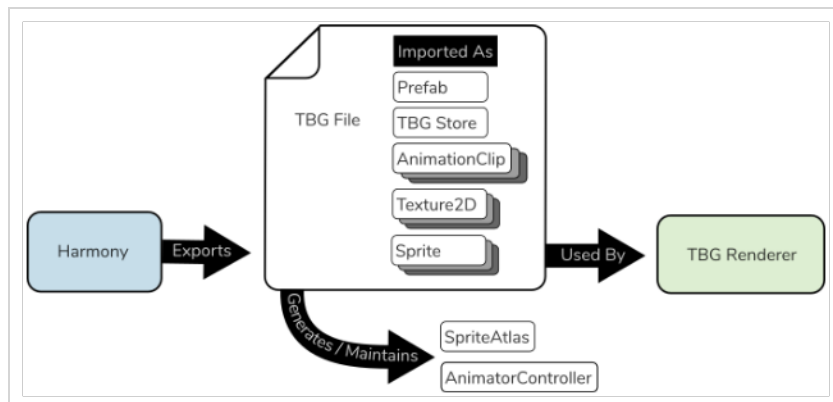
- 使用 Unity 制作原型，然后转移到其他游戏引擎。XML 工作流调用可移植到其他引擎的 C++ 库，因此可转移。
- 想要更多空间来使用自定义资产流解决方案，并希望有一个创作自定义高性能程序动画特效的起点。

**提示**

- 如果有此想法，可以在后面切换工作流。
- 两个系统都默认使用 **Animator Controller** 选取和播放角色动画，仅与角色的 **Animator** 交互的脚本具备类似功能。但更改蒙皮、分辨率或提供程序特效的脚本，在两种工作流上都必须覆盖或自定义。
- 总的来说，从 **HarmonyRenderer** 过渡到 **TBGRenderer** 更好，因为 **HarmonyRenderer** 默认具有 **TBGRenderer** 的功能子集。如果需要不同的设置或工作流重新导出，最好维持原始 **Harmony .xstage** 项目的访问权限。

Unity 中的 TBG 文件工作流

在 TBG(Toon Boom Gaming) 文件工作流中, Unity 会将 Harmony 导出的 TBG 文件立即解释为新资产, 该资产可引用, 从文件衍生出新的纹理、精灵、项目数据和 Prefab。生成的 Prefab 拖到 Unity 场景, 立马就能查看并制作动画。TBG 文件被 Harmony 新导出的内容覆盖时, TBG 导入器可自动执行重新导入流程, 更新现有的 prefab、精灵和纹理。艺术家因此可以快速重做角色, 并在游戏中马上看到成果。



生成的 TBG Renderer prefab 包含完整的变换和 SpriteRenderer 层级, 便于用适合 Unity 的绑定展示角色。其中还包含一个 TBG Renderer 脚本, 使动画过程顺利交换精灵, 对相关 SpriteRenderer 正确应用切割器。



注意

TBG 文件采用的 XML 格式为更新版本, 可能不兼容 XML 文件夹工作流, 无法创作更复杂的角色。最好保留原始 Harmony 文件, 之后使用不同的设置重新导出。

如何将 TBG 导入 Unity

1. 请在 Harmony 的 Game(游戏) 工具栏找到 Export to Sprite Sheet(导出至 Sprite Sheet) 按钮。
2. 在 Export to Sprite Sheet(导出至 Sprite Sheet) 窗口, 勾选 **Encode as TBG(编码为 TBG)**。
3. 在 Asset Path(资产路径) 输入要保存角色的文件路径。
4. 选择 **Save and Export(保存并导出)**。此时会出现一个加载条, 完成导出过程后将生成一个新的 TBG 文件。
5. 打开您的 Unity 项目。Unity 会自动将 TBG 加载为一个 Prefab。



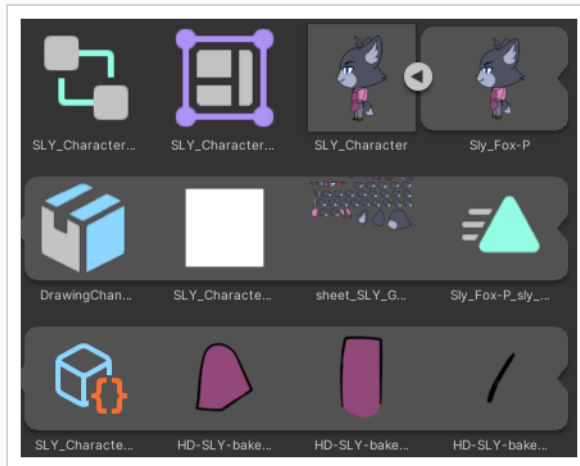
注意

如果角色未作为 Prefab 显示, 请确认:

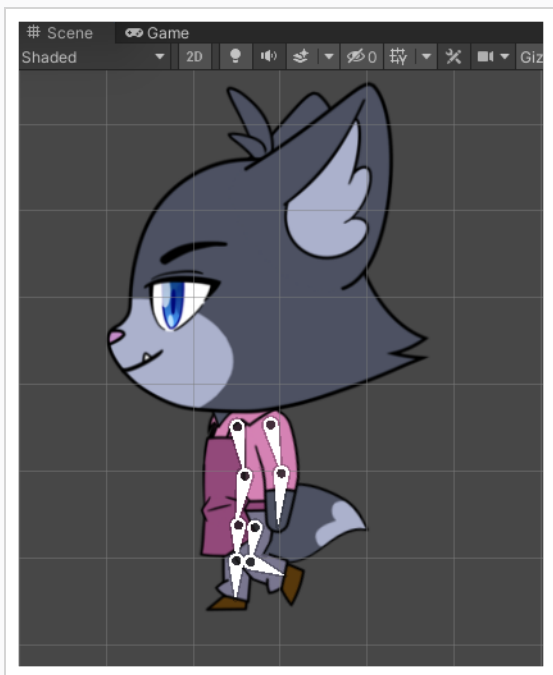
- 已为 Unity 项目安装了 HarmonySDK。每个新建项目都必须安装。
- 场景标记均未超出角色曝光范围。

Unity 项目应有:

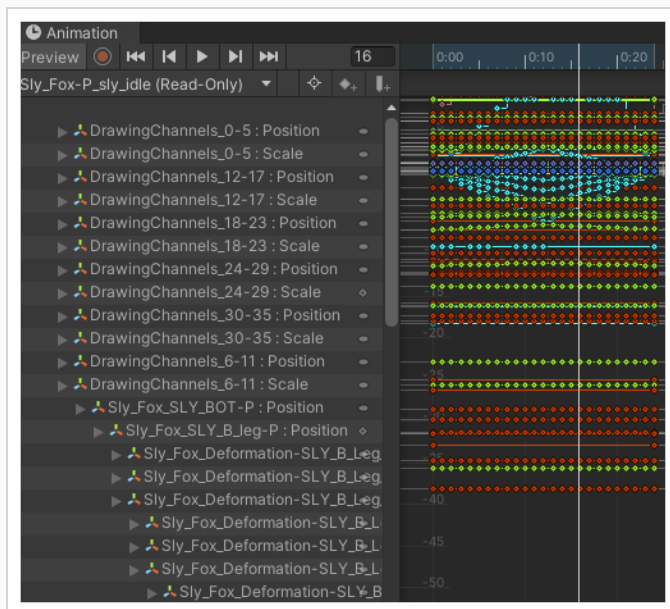
- 纹理、精灵、TBG Store 数据和 AnimationClip 等 sub-asset。
- TBG Prefab 旁边的两个新的文件资产 SpriteAtlas 和 AnimatorController。



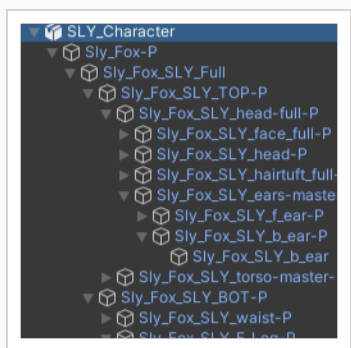
现在 TBG Prefab 可拖入层级, 在其中显示和交互。



选择 Hierarchy(层级)中的 Prefab(预制件), 可以查看动画视图下的角色动画。

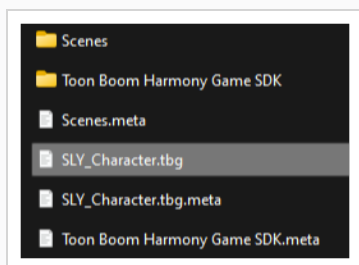


单击小箭头后在层级视图中打开 **Prefab**, 可以探索角色的层级, 您会发现它与 **Harmony** 的时间轴中所显示的非常相似。

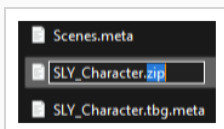


如何手动检查 TBG 文件的内容

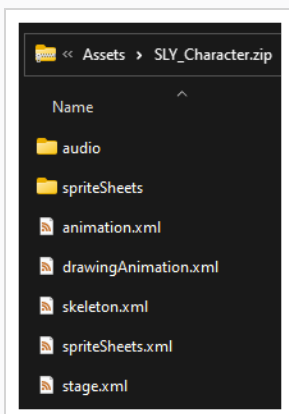
1. 在计算机的文件资源管理器中搜索 **TBG** 文件



2. 若使用 **Windows** 或 **Mac**, 请重命名文件扩展名, 因为 **Windows** 和 **Mac** 计算机无法识别 **.tbgi**。



重命名后就可以在 Windows 和 Mac 浏览 .zip 文件的内容，因此 SpriteSheet 的大小务必要合理，数据文件要包含需要的内容。



注意

重新使用 Unity 前，文件扩展名要改回 .tbgs，避免 Unity 出现问题。通常，只有在未安装 Unity 的计算机上才需要执行此检查过程。

为 TBG Renderer Prefab 添加反向动力学功能

您可以使用 Unity 本身的 2D 反向运动学功能为角色肢体添加程序动画目标。可以在此流程中使用 TBG 工作流示例场景中的 SLY_Character。

如何为 TBG Renderer Prefab 添加反向动力学功能

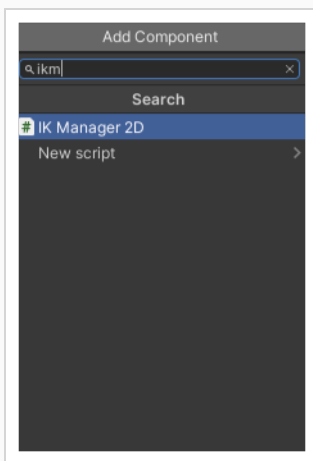
1. 在项目视图找到 **SLY_Character** 并拖拽到 **Scene view**, 然后按下 **F** 使角色固定在视图中。



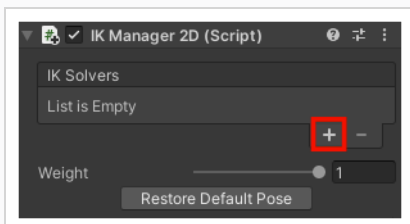
注意

一定要在层级中选定 **SLY_Character**。

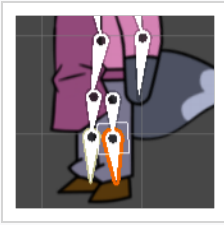
2. 在 **Inspector view** 中添加一个 **IK Manager 2D** 组件。



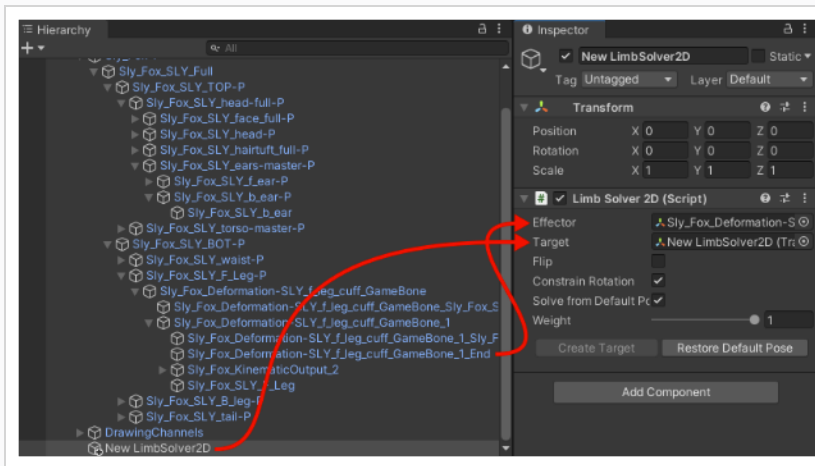
3. 在 **IK Manager 2D** 组件中单击 **+** 按钮新增一个 **IK 解算器**, 然后选取 **Limb(肢体)**。



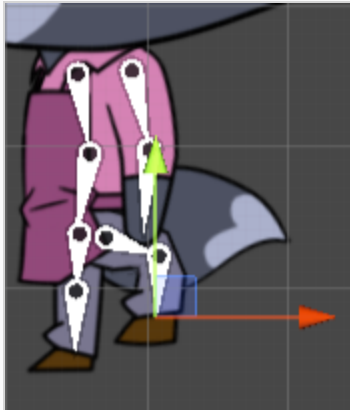
- 单击 Sly 腿部右上方骨骼，显示此部位的层级。我们要使用其同级 GameObject(名称包含“..._GameBone_1_end”)。



- 选定“New LimbSolver2D”游戏对象后，将“New LimbSolver2D”拖入 LimbSolver2D 脚本的 Target (目标) 槽，再将“_leg_cuff_GameBone_1_end”GameObject 拖入 Effector(特效器) 槽。



完成后可以环绕新的 LimbSolver2D 变换进行移动，查看肢体弯曲以匹配目标。



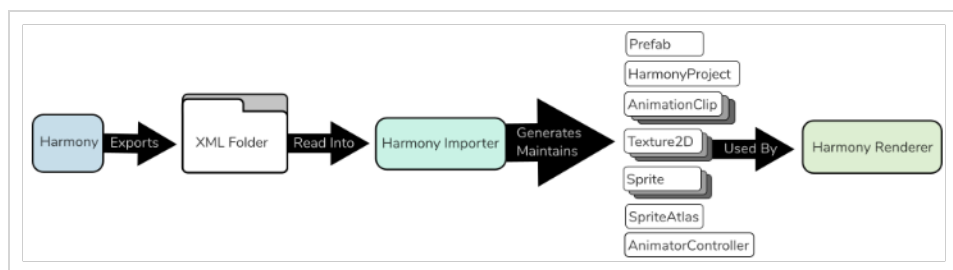
如需切换膝盖弯曲的方向，可以在肢体解算器 2D 组件中单击“Flip(翻转)”复选框。

Unity 中的 XML 文件夹 workflow

XML 文件夹 workflow 中, 开发人员可以创建一个特殊 **Harmony** 导入器资产, 引用从 **Harmony** 导出的 XML 文件夹。然后使用此导入器生成 **Harmony** 项目资产和新的 **Harmony Renderer prefab**。

开发人员需要对每个重新导出的 XML 文件手动设置导入和导出路径, 并从 **Harmony** 导入器资产执行导入过程。

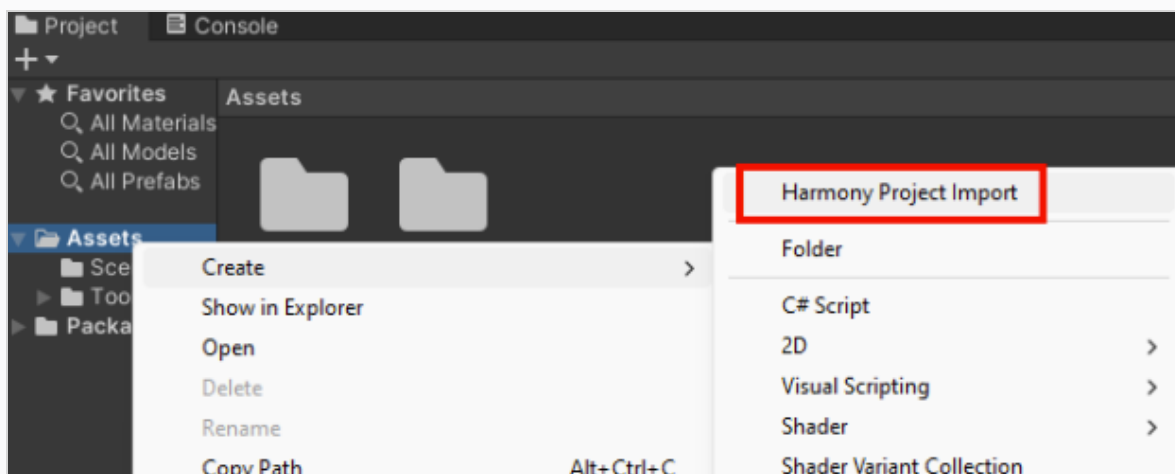
该 workflow 使 XML 数据存放位置灵活。例如, 可存于 **Unity** 项目内, 也可以存于项目外。但此 workflow 涉及更多步骤, 因此错误的几率更大, 这是它的缺点。例如, 移动文件时, **TBG** 导入器的文件路径必须手动更新。



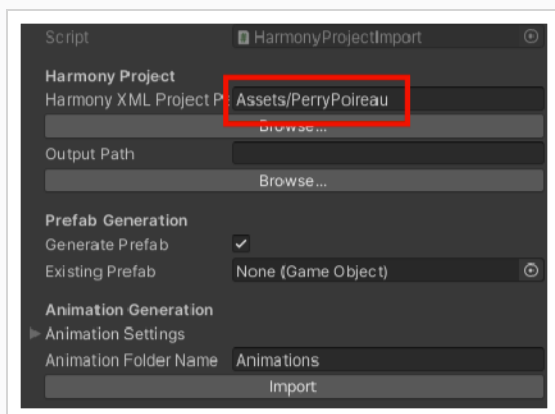
生成的 **Harmony Renderer prefab** 在 **C++** 插件中生成网格, 实现高性能角色渲染。这种方式适合不需要复杂的自定义服装渲染功能的角色群。

如何将 XML 文件夹导入 Unity

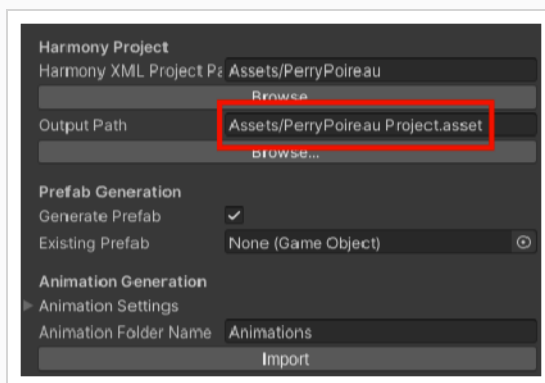
1. 在 **Harmony** 的 **Game(游戏)** 工具栏找到 **Export to Sprite Sheet(导出至 Sprite Sheet)**  按钮。
2. 选择 **Save and Export(保存并导出)**。确认 **Export to Sprite Sheet(导出至 Sprite Sheet)** 对话框中未勾选 **Export as TBG(导出为 TBG)** 选项。
3. 将 XML 保存到 **Unity** 项目 **Assets(资产)** 文件夹内的一个文件夹。
4. 在 **Unity**, 执行下列操作之一新建 **Harmony** 导入器:
 - 在顶部菜单的 **Assets(资产)** 下拉菜单选择 **Create(创建) > Harmony Project Import (Harmony 项目导入)**。
 - 右键单击 **Project(项目)** 视图中的 **Assets(资产)** 文件夹, 选择 **Create(创建) > Harmony Project Import(Harmony 项目导入)**。



5. 命名该 Harmony 导入器，需包含“Importer”后缀，避免以后混淆。
6. 复制导出的文件夹的相对路径，粘贴到新建的 Harmony 导入器中。

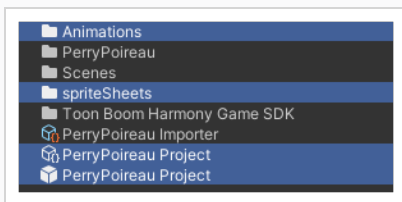


7. 复制 Harmony 导入器资产的相对路径，粘贴到 Harmony 导入器中，将路径中的“Importer”替换为“Project”。

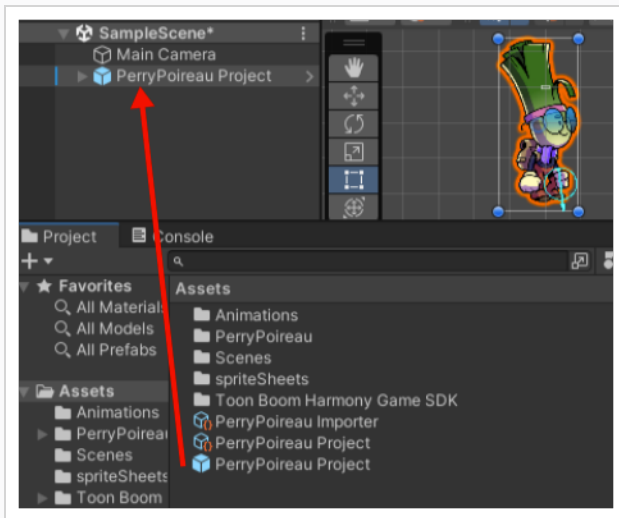


8. 单击 Import(导入)。

这样就创建了一个新的 **HarmonyProject** 资产以及一个带 **Harmony** 渲染器(引用 **HarmonyProject**) 的新 **Prefab**。



该 **Prefab** 资产可从“项目”视图拖入“层级”视图和 **Scene View** 中, 使游戏显示角色

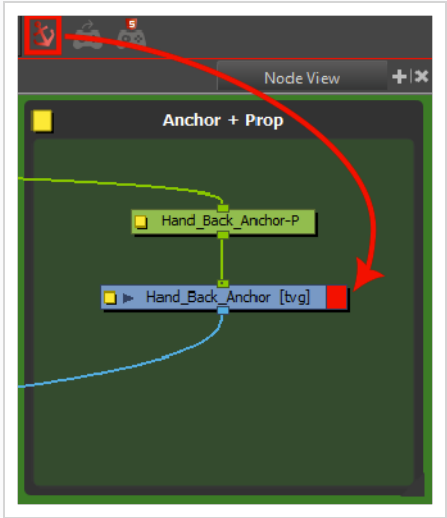


导入锚点

Harmony 中, 锚点指示哪些定位钉和绘图是 **Unity** 的游戏脚本感兴趣的位置。锚点可以分配任意绘图或定位钉。

如何将锚点导入 **Unity**

1. 在“时间轴”视图中选择一个图层或在“节点”视图中选择一个节点, 然后选择 **Game(游戏)** 工具栏中的 **Toggle Anchor(切换锚点)** 按钮。



2. 将角色导入 Unity。

下次执行 XML 导入时，新锚点将位于 Unity 中角色的导入 Prefab 下方，显示为新的 HarmonyAnchor。



此新增 **GameObject** 将跟随角色当前动画播放的姿态、旋转和缩放。这样可以将游戏对象作为子对象动态附加到锚点，进行携带、装备和使用。



注意

移动锚点位置不影响 Harmony 渲染器的视觉效果。若要查看和影响 Unity 中角色的整个结构而不仅仅是锚点，可以考虑使用 TBG 工作流

添加音源

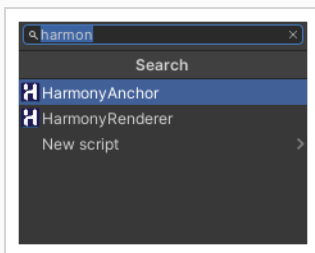
在 Harmony 创建带音频的场景文件时，该音频导出位置是项目文件夹。Harmony 导入器会自动创建一个 **AudioSource** 并添加到所生成的 Prefab。之后，Harmony 渲染器将在动画播放期间某个点触发 **AudioClip** 文件播放，与 Harmony 中的创作相同。

向空游戏对象添加 Harmony Renderer

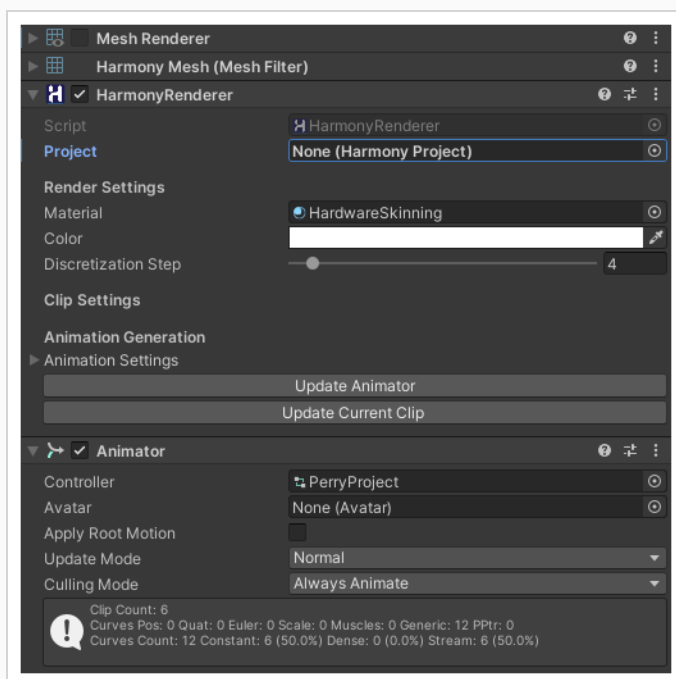
如您没有来自 Harmony 导入器的 prefab, 仍然可以在一个空游戏对象上新建 Harmony Renderer 组件, 获得相同效果。

如何使用空的 GameObject

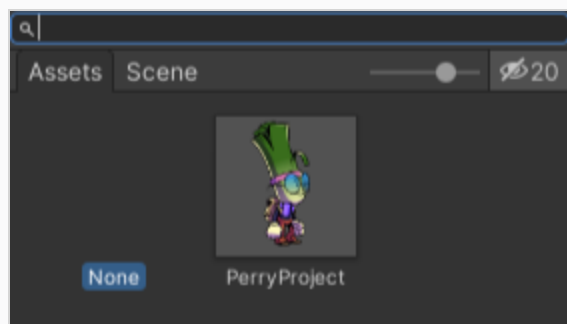
1. 选择 **GameObject > Create Empty(创建空的)**。
2. 重命名这个空的 **GameObject**, 因此清空其层级。鉴于我们使用的是 PerryPoireau 演示文件, 请执行以下一项操作将 **GameObject** 重命名为 **SpaceCat**:
 - 双击名称, 在层级中重命名。
 - 选择 **GameObject**, 在检查器中重命名。
3. 此时, 它是一个空对象。通过脚本访问 Harmony 数据。
 - 选择 PerryPoireau **GameObject**。
 - 在检查器内, 转到 **Add Component(添加组件) > Scripts(脚本) > Harmony Renderer**。



检查器内出现四个区域, 分别是 **Mesh Renderer**、**Harmony Mesh(Mesh Filter)** 和 **Harmony Renderer**。



您需在此为新建的 **Harmony Renderer** 添加一个 **Harmony** 项目。单击空 (**Harmony 项目**) 槽右侧的圆圈, 选择在 **XML 文件夹工作流** 创建 **Harmony** 项目的步骤中已导入的项目。



在 Unity 设置碰撞

如果要制作某些事物碰撞角色, 提取边界框信息很有用。将 2D 角色放入 3D 场景时, 您可能要使用物理对象使 2D 角色碰撞 3D 平面。

碰撞体改进后提供几种可在 Unity 使用的碰撞体。可以选择多边形碰撞体和盒型碰撞体, 获得更准确的边界框。

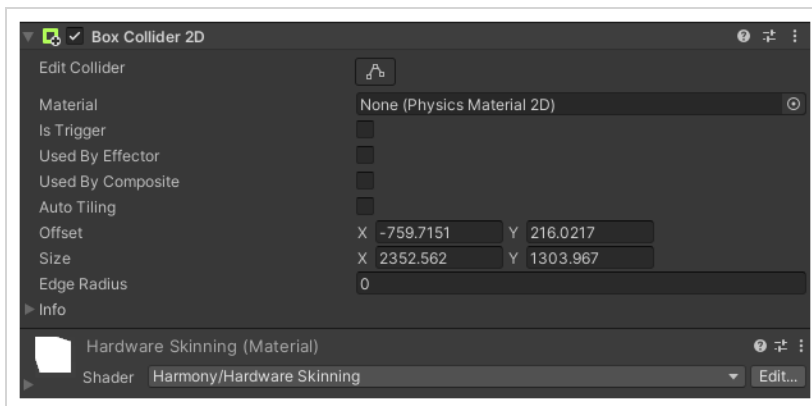
如何设置碰撞

1. 选择导入的角色。
2. 在检查器内, 选择 **Add Component(添加组件) > Physics 2D(2D 物理) > Box Collider 2D(2D 盒型碰撞体)**。
3. 如需启用角色的“物理对象”, 请转到 **Add Component(添加组件) > Physics 2D(2D 物理) > Rigid Body 2D(2D 刚性体)**。



注意

也可以在导入文件上使用 3D 物理对象。



现在可以调整变量, 查看结果了。

游戏术语

3D 立体影像	实际输出两个不同图像的图像或影片。其中每个图像各自对应人的左右眼的视角。通过特殊设备(例如 3D 显示器或虚拟现实目镜)观看或投影时,会使人产生图像中的元素位于眼前的感觉。
Alpha 通道	记录透明度信息的图像通道。图像本身已具有红、绿、蓝 (RGB) 三个通道。Alpha 通道是第四个通道 (A)。蒙版或透明度信息存储在第四个通道中。没有 Alpha 通道的图像总是不透明的。
CMYK	青色、品红色、黄色、黑色英文首字母缩写。这些颜色是胶板印刷过程中用到的标准模型。
Cut-out 动画	Cut-out 动画是通过逐帧移动由多个部件组成的角色以动画的操作。Cut-out 动画可以由计算机生成,也可以使用纸张以传统方式完成。
dpi	即每英寸点数,是数字化打印机分辨率的标准度量单位。有时也应用于屏幕,在这种情况下,称之为每英寸像素数更为准确。无论哪种情况,所谓的点就是构成图像的最小元素。
film-1.33	符合标准 4:3 像素宽高比的宽银幕影片格式的理想分辨率。
film-1.66	符合 16:9 像素宽高比的宽银幕影片格式的理想分辨率。
HDTV	高清电视的英文首字母缩写,是目前电视制作的分辨率标准。HDTV 制作的分辨率为 1280×720(HD就绪)或 1920×1080(全高清),其帧率范围一般为 23.976 至 30。
HSV	色相、饱和度和值。通过色相(色彩)、饱和度(明暗度)和值(色调或亮度)来定义颜色的方法。
NTSC	北美地区采用的传统标准模拟电视广播系统。
PAL	最适合用于欧洲电视和计算机屏幕格式的分辨率,因为矩形像素显示的方向不同。
RGB	即红、绿、蓝。是通过指定三种颜色的数值定义颜色的方法。
安全框	动画和构图艺术家用于定义摄像机区域大小以及摄像机移动的、包含所有框单位的参考线。
安全区域	位于场景框的中心、不会被电视框裁去的区域。电视框会剪裁原始框的边缘部

	分。保留安全区域, 可以确保影片在电视上播放时场景的主要动作保持清晰可见。
绑定	安装 cut-out 木偶各个部位的过程。
贝塞尔曲线	法国数学家皮埃尔·贝塞尔发明的定义曲线的方法。贝塞尔曲线是一条数学或参数曲线。贝塞尔曲线由控制点定义。在 Harmony 中, 贝塞尔曲线有一个起点、一个终点和两个内部控制点。这两个控制点可影响起止点之间的曲线。
背景	填充摄像机框且位于角色和道具后方的画作。通常, 此画作是动作发生的布景或舞台。
变形	在源绘图和目标绘图之间创建计算机生成的绘图的一个功能。使用变形功能创建的动画可以在不同的项目中重复使用。
补间帧	计算机生成的两个关键帧之间的动作。可以选择是否在关键帧之间创建补间帧。
草稿	草稿是动画片中常用的绘图名称, 用作参考但不构成最终图像。构图属于草稿。草稿是动画或设计的结构草图。草稿通常由草图线和形状组成, 但也可包含设计细节。
场景	影片或节目中的一个镜头。一个连续镜头由多个场景组成。通过简单剪切或过渡, 可从一个场景切换到另一个场景。
传统动画	在纸上手绘所有动画连续镜头, 然后扫描和或化学板上进行描线的动画制作过程。
淡入/淡出	淡入或淡出是用于开始或结束某个剪辑的一种过渡特效。第一个场景逐渐出现, 从完全透明变为完全不透明, 即为淡入效果。最后一个场景逐渐消失, 从完全不透明变为完全透明, 即为淡出效果。
低分辨率	非常适合网络视频的格式, 对于网络视频而言, 文件的大小和下载速度优先于质量。低分辨率图像缺少清晰细节。
底层	在动画中, 底层是指主动画背后所放置的部分具体装饰。
顶层	位于主动画的前面的场景环境的组成部分, 例如椅子或灌木丛。
定场镜头	可以让观众看到某个片段所发生区域的全貌的一个场景。例如, 如果孩子在房屋前的空地上玩耍, 那么在定场场景中, 观众可以看到房屋、空地、部分街道以及动作中心点周边的建筑物。这样能够帮助观众了解故事发生的地点和场景方位。

定位钉	在传统动画中, 用于在化学板图层移动时确保动作精确定位的工具。在数字动画中, 当进行更高级的木偶绑定时, 可以使用定位钉图层。定位钉图层是指不包含绘图的轨迹图层。它们是运动路径, 可用于添加路径关节。对于后者, 还可以使用反向运动工具。
动画	通过绘制或编辑单个帧来生成的模拟运动。
动作关键帧	带有计算机生成的补间帧的关键帧。
动作路径	动作所遵循的路径。
动作线	动作姿势中的主要中线。绘制动作姿势时, 注意动作线可以确保姿势有动感并且使轮廓与动作的方向一致。
对话	在一部影片或动画中角色所说的文字。
多平面	移动多层绘图以形成镜头深度感的特效。在多平面场景中, 将图层放置于距摄像机距离不等之处, 使摄像机移动时产生深度错觉。使用多平面时, 系统会自动计算所有视角和缩放。
翻转	在传统动画中, 翻转是指非常迅速地翻动某个动画片段的绘图以看到绘图动起来的操作。翻转有时也指创建某个对象的镜像变形。
反向运动 (IK)	主要用于为 3D 角色和具有层级的 cut-out 木偶创建动画的功能。反向运动从一个极点(例如手部)创建木偶动画, 并让身体的其他部分随之自然移动。
分辨率	场景的大小, 通常以像素为计算单位。例如, NTSC 分辨率为 720 x 480。分辨率应与最终输出匹配: HDTV、film-1.33、film-1.66、NTSC、PAL、低分辨率。
分解	在 cut-out 动画中, 分解是指为创造出具有关节的木偶而将角色分割为不同部分的操作。要分解角色, 艺术家需要从角色模型中剪切手和手臂等部位, 并将其粘贴至单独的图层。接下来便是固定关节和设定轴心点。在传统动画中, “分解”是一种动画姿势, 通常位于两个关键姿势之间。关键姿势是动画中的主要姿势, 而分解姿势则为次要姿势, 可帮助描述动作和旋转曲线(通常也称为弧线)。
分解镜头	在故事板中, 分解镜头是指镜头中的一帧。一个镜头可以由一个或多个分解镜头组成。
跟随	由主要动作引发的次要动作。例如, 某个穿着斗篷的角色正在奔跑。此时主要动作是奔跑。这会导致斗篷跟随该动作而移动, 尽管二者并非同时移动, 但几帧之后, 次要动作便会作出反应并随主要动作曲线运动。

构图	构图过程是连接故事板和动画的中间步骤。是绘制故事板模型的操作，即根据模型包中的设计绘制角色，以便动画师能开始工作。构图设计师绘制背景，创建与场景及摄像机运动一致的摄像机和框参考线。最后，由动画师在模型上绘制主要动作姿势。
构图规划	在规划动画并执行其初始阶段时可用作参考的场景的主要特点的绘图。
构图过程	构图过程是连接故事板和动画的中间步骤。
构图和创建姿势	绘制故事板模型(以正确的缩放比例)以便动画师开始工作的操作。
构图设计师	绘制背景、创建摄像机和框参考线以匹配场景和摄像机动作的艺术家。他们在模型上绘制主要动作姿势。
故事板	动画中所有场景和镜头的视觉规划图。故事板显示情节内容、发生时间以及如何场景中放置对象。
关键帧	在一个动作中定义了起始点和结束点的重要位置。关键帧是在指定轨迹上的特定时刻(即帧)由计算机生成的位置。
关键姿势	动作中的重要位置，定义了平滑过渡的起始点和结束点。关键姿势是动画片段中描述动作的主要绘图。例如，如果手臂正在挥动，则关键姿势是手臂动作的一个极点和另一极点处的姿势。通过翻转这些绘图，动画师可以看到动作的结构，而无需绘制所有绘图。
轨迹	计算机生成的路径或元素可以追随的轨迹。可以通过控制点、关键帧和速度来控制轨迹。
过渡	从一个场景切换到另一个场景时，在两个场景之间放置的特效。常见的过渡特效包括交叉溶解和擦除。
过渡位置	当绘制角色的行走片段时，过渡位置是指一条腿迈过另一条腿的位置。
函数	计算机生成的运动、轨迹或路径，可以在其中添加元素、其他轨迹或特效参数。通过在函数曲线上添加关键帧和控制点可以控制函数。
合成	合成是指在渲染前将场景的所有元素进行合并以达到最终效果的操作。例如，合成艺术家将在场景中导入所有动画连续镜头、背景、顶层和底层，并把它们放置在正确的位置。然后，艺术家将设置摄像机框并视需要创建摄像机动画。最后，动画师将为该项目创建所有计算机生成的特效。

化学板	在传统动画中, 化学板(又称赛璐珞) 是指将动画传送至摄像机前, 在其上对动画进行描线和着色的一张透明板。在化学板的正面绘制图片轮廓, 并在背面上色。
画面停格	在动画中, 角色保持位置不动的帧。可以在两个关键帧之间创建画面停格。
剪切	一个场景和随后场景首尾帧直接相连, 没有任何视觉效果。
渐快	动作的逐步加速, 也称为慢进。
渐慢	动作的逐步减速, 也称为慢出。
键盘快捷键	指定用于在按下时执行应用程序中特定命令的键盘按键或按键组合。
交叉溶解	一个场景通过淡出而过渡到随后场景, 同时随后场景淡入。
角色设计	动画片中的每个角色都会以招贴画风格从多个角度绘制, 这称为模型板, 可供动画师们参考。
镜头	影片或节目中的一个场景。一个连续镜头由多个镜头组成。通过简单剪切或过渡, 即可从一个镜头切换到另一个镜头。
剧本	包含所有影片或节目信息的原始文本。在动画中, 剧本包含所有位置说明、对话、时间等信息。项目始于剧本。
剧情线	剧情极少以直接向前推进的方式发生, 通常是逐渐展开的, 作家将其称为剧情线。剧情线的主要目的是将角色或情形从一个状态或情形移动到下一个状态或情形。
口型同步	将某个角色的口型与对话音轨中的声音进行同步的过程。口型会逐帧调整, 使其与对话的声音同步, 产生角色正在说话的错觉。口型同步可用于任何声音片段而不仅仅是说话, 例如可以将其用于鸟鸣或月夜狼嚎。
口型图	基于 8 个动画音位(A、B、C、D、E、F、G, 以及用于表示静默的 X) 绘制, 用于口型同步的图表。
快出	动作结束时的明显加速。
快入	动作开始时的明显加速。
宽高比	任何场景、帧或影片格式的宽度和高度尺寸的比例。目前, 最主流的宽高比为

	16:9。老式显示器使用 4:3 的宽高比。
框	传统动画中使用的一种度量单位,用于记录和跟踪摄像机的位置以及与舞台的距离。
连续镜头	组成故事或影片的特有部分的一系列场景或镜头,通常以相同位置或时间为单位连接。
慢出	动作的逐步减速,也称为渐慢。
慢进	动作的逐步加速,也称为渐快。
描绘和着色	草图动画经过修形以及最终的线条或铅笔测试后,对每一张绘图进行描绘和着色,以制作最终动画。在当今数字世界,除了传统的化学板或透明醋酸纤维胶板方法外,还可以通过多种其他方式进行描绘和着色。
描线和着色	描线和着色是根据颜色模型在最终动画绘图上描绘空白区域并为线条上色的过程。
模板	存储在文件库中的资产,可以在任何项目中重复使用。模板可以是绘图、一系列关键帧、声音文件、分解镜头、cut-out 角色、特效、轨迹、动画,或动画中用到的任何其他内容。
模型/颜色模型	每位艺术家在制作过程中必须遵循的最终的角色、道具或位置设计。颜色模型是指在为动画着色的过程中必须要使用的正式颜色设计。
平板/笔	与鼠标配合使用或取代鼠标,让鼠标指针(有时也称为光标)在计算机屏幕上移动的设备。
曝光	在动画中,曝光是指场景中出现的绘画使用的化学板数量。要延长绘图的显示时间,则必须在更多化学板上曝光。
曝光表	包含垂直列和水平框、用于表示场景时间的表格。每一列代表一个场景图层。帧号显示于每列中,并且在其需要出现的时间范围内重复显示。动画师、导演或剧组其他成员使用曝光表跟踪图像、对话、音效、音轨和摄像机移动的片段和时间。也称为摄影表。
起止时间说明	说明对话和相关动作的开始和结束时间。
清稿	草图经通过测试和审核后,需要去除图像中的所有杂物(多余的线、注释等),从而创建出可以描线、着色并拍摄的最终绘图。修形既可以指在草图上描摹平整线

	条来完成最终版本的过程，也可以指移除扫描后遗留下来的污渍或多余的线条。
日本动漫	日本动画。
色板/主调色板	色板/主调色板是指专用于某个角色或道具的一组颜色。在整个制作过程中，使用色板来保持外观的一致性，并避免在动画过程中颜色发生改变。也称为主调色板。
色板样式	色板样式是现有色板的第二个版本，其色彩和数值略有变化。色板样式可用于创建色板的夜间版本，也称为克隆色板。
色差	在光学中，色差(CA, 又称消色差或色彩失真)是一种失真类型，是由于镜头无法将所有颜色对焦至同一个会集点所导致的。
色卡	色卡是指包含一种纯色的卡片，其尺寸与摄像机相同。当不包含任何背景图像时，色卡使用纯色来填充背景。
色域	某台特殊设备可以显示的颜色范围。
摄像机抖动	当摄像机沿多个方向轻微地快速移动时，场景中会出现摄像机抖动的现象。这能使观众留下撞击、震动或者道路颠簸等印象。
摄影表	动画师、导演或剧组其他成员用摄影表来跟踪图像、对话、音效、音轨和摄像机移动的顺序和时间，也称为曝光表。
声音滑动	让您在向前或向后移动播放头时能实时听到声音的过程。该功能在微调口型同步时很有用。
时间码	影片片段上打印的时间信息，用于指明屏幕上当前显示的场景以及时、分、秒信息。
时间轴	水平显示场景元素、时间和关键帧的方式。
矢量	基于矢量的图像由点和贝塞尔曲线组成。计算机读取这些点后描绘线段，然后将它们连起来以重现图像形状。矢量图像没有固定的尺寸或分辨率。可以随心所欲地放大并扭曲图形，系统只须重新计算线段并重新构建形状即可生成图像。计算完成后，矢量图将转换并以像素为单位显示。
手动口型同步检测	手动替换口型位置绘图，使之与音轨匹配。在此过程中，会用到声音滑动(逐帧聆听分解的声波)和绘图替换。
双跳跃步	在双跳跃步行的关键帧和过渡位置中，身体低于以直立位置绘制的参考线。在中

行	间帧中, 身体高于此线便能造成跳跃的感觉。
速度	在动画中, 速度(也称为速率)是指某个动作的加速或减速。它可以是由函数曲线创建的动作, 也可以是由一系列动画绘画形成的动作。渐快和渐慢也常常称为慢进和慢出。
速率	在动画中, 速率(也称为速度)是指某个动作的加速和减速。它可以是由函数曲线创建的动作, 也可以是由一系列动画绘画形成的动作。渐快和渐慢也常常称为慢进和慢出。
缩略图	一张用作参考或指示的小图片。
跳接	由前一个场景突然切换至后一个场景。通常情况下, 跳接的视觉效果欠佳。它通常是由于在一个场景结束和另一个场景开始时使用相似的图像而造成的。由于变化不大, 因此造成视觉上有画面唐突跳跃的感觉。
停止动作 关键帧	不含计算机生成的补间帧的关键帧。
透光台	让您在处理某个图层的同时能够清楚地看到其他图层的设备。
图层	在动画中, 图层是指单独的列、级别或角色。场景图层叠加后便形成最终图像。
拖动	单击并按住项目, 然后移动鼠标光标来移动该项目的一种操作。这可以通过鼠标左键或笔尖来完成。
微调	使用键盘上的箭头键对选定元素进行轻推(向左、向右、向上、向下、向前或向后)。微调用于非常轻微、准确地移动选定元素。
位图	由像素网格组成的图像, 其中每个像素的颜色均单独定义。相对于矢量图形而言, 位图图形的优点是画作精细逼真。但是, 位图图形不能缩放、旋转或倾斜, 否则会损失图片质量。
文件库	包含可以在任何项目或场景中重复使用的模板及资产的存储区域。
无纸化动画	无纸化动画过程是以数字方式创建动画的操作。主要的无纸化动画过程是直接 在软件中一帧一帧地绘制动画。
线条	在不表示实际画刷或铅笔笔触时, 线条是指用于定义颜色区域或闭合开放颜色区域的不可见轮廓。

像素	显示器或电视屏幕上所显示画面的最小元素。像素是图像元素的简称，指图像中的单个点。像素是图像的一个小样本，是一个点、一个方块，或者是平滑滤出的非常小的一部分。放大数字图像到足够近的位置，便可以看到像素，它像是不同颜色和亮度的小方块。
旋转台	允许旋转工作台以增加绘图时的舒适度的设备，相当于动画盘/动画台。
渲染	使用计算机创建动画时的最后一个步骤。在渲染过程中，计算机计算屏幕上出现的每个像素、处理所有组件并添加动作模糊效果，最后制作出最终的图像。它是合成后计算最终图像的过程。
循环	可以通过重复使用来模拟连续运动的一系列动画帧。例如，通常制作角色向前走动两步的动画（每只脚走一步），然后循环播放该动画，使角色行走。
颜色轮盘	以圆盘的形式显示的色谱。
颜色模型	在为动画着色过程中必须要使用的正式颜色设计。模型是指艺术家在制作过程中必须遵循的最终的角色设计、道具设计或位置设计。
洋葱皮	可以使您看到连续镜头中前后绘图的功能。
样片	通过对故事板中的面板进行排序，并根据其代表的动作的粗略持续时间对每个面板进行定时，然后与音轨同步而制作的影片。样片用于将故事板转换成反映最终影片的粗略草稿，用来确定分配给每个场景和每个动作的时间，并将动作与音轨同步。
摇摄	在场景中沿任何方向移动摄像机。
音轨拆分	将动画片的音轨拆分为单独的声音，逐帧生成每个声音的精确位置。
音位	语言中的声音单位。
元件	元件将动画、艺术装饰或图层合并到一个单独的对象中，使您能够在一个图层中进行控制。可以对 cut-out 木偶的各个身体部位创建元件。可以在元件中放置任何内容。可以使用元件来创建木偶动画，或创建可重复使用的动画效果，例如眨眼。
着色区	可供上色的区域。
帧	一个动画帧即为影片中的一张摄影图像。在传统动画中，北美标准通常为每秒 24 帧，而欧洲标准则为每秒 25 帧。

帧率	帧率是指帧的播放速度。通常按每秒帧数来计算。例如, 场景可以按每秒 12、24、25、30 或 60 帧或任何其他数字的速率来播放。帧率是成像设备形成单独连续图像(即帧)的频率(速率)的度量单位。该术语同样适用于计算机图形、摄像机、胶片摄像机以及动作捕捉系统。帧率最常表示为帧/秒 (fps), 而在逐行扫描监视器中表示为赫兹 (Hz)。
正向运动	正向运动是用于为 3D 角色和具有层级的 cut-out 木偶创建动画的功能。它可用于从一个父级部件(例如肩膀)创建木偶动画, 并让手臂的其他部位作为单独的一个部件随之移动。
直前动画	从起始位置到结束位置按顺序逐一绘制整个片段的技巧。使用此方法时, 几乎不进行规划。此时, 角色最后的状态以及导致该状态的原因, 可能会让观众和动画师都感到出乎意料。虽然这种方法比较自然且较具创意, 但可能会产生不准确的结果。
制作摄影表	将某张特殊绘图分配给一个范围内的帧。
中间帧	关键姿势之间的绘图。绘制这些绘图是为了实现姿势间的流畅过渡。
轴	对象围绕其旋转的假想线。2D 图形有两个轴: X 轴(水平)和 Y 轴(垂直)。3D 图形有三个轴: X 轴(水平)、Y 轴(垂直)和 Z 轴(深度)。在不断旋转的动画中, 轴元素指定了对象围绕哪个轴进行旋转。如果数值为负, 则动画会沿逆时针方向旋转; 而如果数值为正, 则动画会沿顺时针方向旋转。
轴心点	定位钉或绘图绕其旋转的点。
主调色板	某个角色或道具专用的一组颜色。在整个制作过程中, 使用调色板来保持外观的一致性, 并确保在制作过程中使用相同的颜色。也称为色板。
转描	动画片中用到的一种动画技巧, 其中动画师会一帧一帧地描绘真人动画片的运动。是通过描绘真人动画片镜头以创建动画片段的操作。
姿势到姿势动画	姿势到姿势动画是指创建所有主要动作姿势(称为关键姿势)并在关键帧之间放置次要姿势的过程。次要姿势也称为分解姿势。最后, 动画师用中间帧绘图来填充空白帧以实现平滑的动画效果。
自动进纸	将绘图送入扫描仪的自动化方法, 采用这种方法将多张堆叠的绘图置入送纸器中。扫描仪启动后, 会连续扫描绘图, 无需用户进一步干预。
自动口型同步检测	自动将元素中的绘图映射到为声音生成的口型图。这样, 对音轨进行口型同步时能够节省时间。

字幕

故事板中包含对话、特效、声音或起止时间说明信息的文本字段。

走路循环

一系列“现场”绘图，用于描绘角色的步行动作。利用背景平移可以制造出移动的错觉。为了避免绘制太多绘图，动画师通常会为角色制作步行循环。