

Harmony22

**Toon Boom
Harmony22**

Jeux



TOON BOOM ANIMATION INC.

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

+1 514 278 8666

contact@toonboom.com
toonboom.com

Mentions légales

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

Tél. : +1 514 278 8666

Fax : +1-514-278-2666

toonboom.com

Clause de non-responsabilité

Le contenu de ce document est la propriété de Toon Boom Animation Inc.; il est protégé par le droit d'auteur. Toute reproduction, en tout ou en partie, est strictement interdite.

Le contenu de ce document de est couvert par une garantie limitée spécifique et des exclusions et une limite de responsabilité en vertu du Contrat de licence applicable, complétées par les conditions générales spéciales du format de fichier Adobe® Flash® (SWF). Pour plus de détails, veuillez vous référer au contrat de licence et à ces conditions particulières.

Marques déposées

Toon Boom® est une marque déposée. Harmony™ et le logo Toon Boom sont des marques déposées de Toon Boom Animation Inc. Toutes les autres marques déposées appartiennent à leurs propriétaires respectifs.

Date de publication

02-22-2023

Copyright © Toon Boom Animation Inc., une société de Corus Entertainment2023Inc. Tous droits réservés.

Table des matières

À propos du Jeu	4
Notes de mise à jour du SDK Harmony Gaming 2022	5
Chapitre 1 : Exportation Harmony des données de jeu	10
Chapitre 2 : À propos de Création de ressources de jeu	13
Directives pour le rigging de jeu	14
Directives pour les déformations de jeu	16
About Game Bone Deformations	17
Directives pour les découpes de jeu	18
Création de notes de métadonnées	19
Conseils pour l'animation de jeux	21
À propos de la Caméra orthographique	23
Chapitre 3 : À propos de Exportation vers Unity	24
Ancrages	25
Exportation de feuilles de sprite	26
Résolutions de sprite	28
Utilisation de Bake_Groups	30
Exportation vers Easel JS	32
Variations de palettes	34
Chapitre 4 : À propos du SDK Harmony Unity	37
Harmony Previewer	41
Manipulation des ressources	43
À propos de l'Exemple de projet Unity	49
À propos de l'interface Unity	50
Importation de fichiers Harmony dans Unity	52
Workflow de fichier TBG dans Unity	54
Workflow de dossier XML dans Unity	61
Ajout d'un moteur de rendu Harmony Renderer à des objets de jeu vides	65
Configuration des collisions dans Unity	67
Glossaire des jeux	68

À propos du Jeu

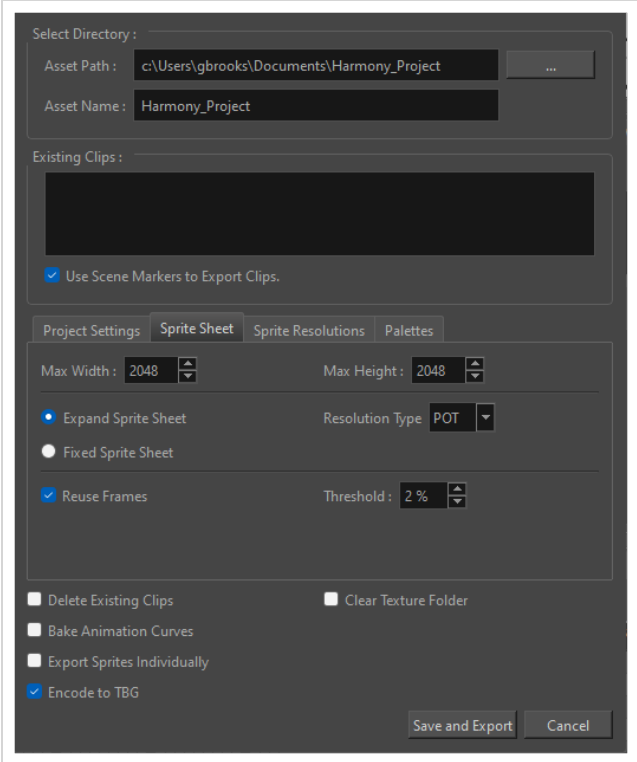
Harmony vous permet d'exporter des animations créées dans Harmony vers des moteurs de jeu externes. Cela vous permet de concevoir des personnages avec des mouvements simples qui peuvent être utilisés dans un moteur de jeu.

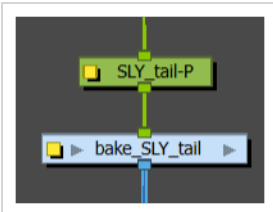
Unity est un exemple de moteur de jeu avec lequel vous pouvez incorporer des données Harmony. Toon Boom dispose d'un SDK Harmony que vous pouvez télécharger à partir de la boutique de ressources Unity Asset pour avoir le contrôle de votre personnage une fois importé dans Unity.

Ce guide montrera comment exporter des animations à partir de Harmony. En outre, le SDK Harmony Game pour l'intégration d'Harmony avec Unity sera également présenté en détail.

Notes de mise à jour du SDK Harmony Gaming 2022

Exportation depuis Harmony

Fonctionnalité	Description
Options pour les nouveaux exportateurs	<p>Pour faciliter une meilleure prise en charge au niveau des éditeurs de jeux, l'interface Exporter vers une feuille de sprite a reçu des fonctionnalités supplémentaires pour regrouper davantage d'informations dans le fichier exporté.</p>  <p>Nouvelles options</p> <ul style="list-style-type: none"> • Onglet de sélection de palette - Exclusion de certaines palettes que vous ne souhaitez pas inclure dans votre fichier exporté, cela couvrira toutes les variations possibles de palettes qui entraîneront l'affichage de différentes couleurs sur le personnage • Encoder en TBG - Comprime les fichiers XML au format de fichier personnalisé TBG afin qu'ils puissent être importés dans Unity avec Unity Custom Importer. Lorsque les fichiers sont importés, ils sont traités comme n'importe quelle ressource dans Unity et contiennent des sous-ressources pour les textures, les sprites et les animations. Cela améliore l'ergonomie de la création et du déplacement des données de personnages et de la vérification des modifications apportées au contrôle de la source. • Performances améliorées sur « Enregistrer et

Fonctionnalité	Description
	exporter » - Le processus d'exportation prend généralement 1/3 du temps par rapport à Harmony 21, pour les dossiers XML et les fichiers .tbg
Groupes « assemblage_ »	<p>Il existe actuellement de nombreux nœuds qu'il n'est pas possible de rendre dans le SDK de jeu. Afin d'apporter de la valeur à ces nœuds pour votre exportation, il existe une nouvelle fonctionnalité pour regrouper les dessins et les nœuds qui les affectent dans un nouveau groupe portant le préfixe « assemblage_ » (par ex. assemblage_hanches).</p>  <p>Tout groupe portant le préfixe « assemblage_ » sera analysé pour des images d'animation uniques pour les effets, les mouvements et les substitutions de dessins. Une fois exportée vers une feuille de sprite, chaque image unique de ce groupe est exportée en tant que sprite. Si aucune modification n'a été apportée à de nombreuses images du montage au sein de ce groupe, elles seront toutes considérées comme le même sprite exporté dans la feuille de sprite</p>

Importation de fichiers TBG dans Unity

Fonctionnalité	Description
Automatic Importing Process	<p>Unity détectera automatiquement les modifications apportées aux fichiers TBG et les réimportera à l'aide d'un importateur personnalisé. Le fichier peut ensuite être glissé dans le plan en tant que prefab, comme vous le feriez avec un fichier FBX.</p> <p>Des textures, des sprites, des animations et des ressources TbgStore seront générés et fournis en tant que sous-ressources sous la principale ressource de prefab TBG. Ces ressources n'auront pas besoin d'être gérées séparément dans le contrôle de code source et n'existeront que dans le cache de la bibliothèque Unity. Des ressources AnimatorController et FeuilleDeSprite sont générées à côté du personnage, car elles ne peuvent pas fonctionner en tant que sous-ressources.</p>
Importer Settings	<p>Le processus d'importation peut être modifié à partir des paramètres Inspector View.</p> <p>Paramètres d'animation</p> <ul style="list-style-type: none"> Discretization Step - Le nombre de colonnes / lignes dans lesquelles les sprites seront divisés s'ils sont déformés par des bones.

Fonctionnalité	Description
	<ul style="list-style-type: none"> • Framerate - Le nombre d'images par seconde que comporte l'animation. • Stepped - Désactive l'interpolation entre les images créées. • Create Animator Controller - Désactive l'interpolation entre les images créées. • AnimatorController - Fournit un espace de travail pour faire la transition des animations entre elles sous certaines conditions. Peut être référencé dans les scripts de jeu pour déclencher des animations pendant le jeu. • Maintain Curves on Cloned Clips (Maintenir les courbes sur les clips clonés) - Les courbes dupliquées à partir de sous-ressources de clip dans le fichier .tbg (et référencées dans AnimatorController) verront leurs courbes remplacées par de nouvelles données du fichier .tbg mis à jour. <p>Paramètres du matériau</p> <ul style="list-style-type: none"> • Shader - Utilisé pour faire le rendu de tous les sprites dans le prefab • SRGBTexture - Mappage de l'espace colorimétrique gamma des textures générées dans le rendu de l'espace colorimétrique linéaire • FilterMode - Comment les pixels voisins de la texture sont interpolés • MipmapActivé - Génère des textures de résolution inférieure pour obtenir un rendu à des distances plus grandes • CréerUnSpriteAtlas - Si aucun SpriteAtlas n'est référencé ci-dessous, une nouvelle ressource SpriteAtlas sera créée dans le projet à côté du fichier .tbg • SpriteAtlas - Tous les sprites du fichier .tbg peuvent être ajoutés à un SpriteAtlas, ce qui permettra de combiner plusieurs SpriteRenderers originaux en un seul appel de dessin, améliorant ainsi les performances de rendu

Rendu de fichiers TBG dans Unity

Fonctionnalité	Description
Prefab Hierarchy	<p>La structure du Prefab généré dans Unity doit correspondre à la structure hiérarchique mise en place dans le montage dans Harmony. La structure de la Node View d'Harmony des composites servant à alimenter le nœud de sortie sont exclus lors de l'exportation.</p> <p>Les nouveaux ObjetsJeu peuvent être attachés à n'importe laquelle de ces transformations de cheville enfant. Cela permet de fixer l'équipement procédural.</p> <p>En outre, n'importe quel ObjetsJeu peut être activé ou désactivé au moment de l'exécution, de sorte que les ajouts visuels facultatifs sur les personnages peuvent être masqués en attendant qu'ils soient nécessaires. La hiérarchie peut également être entièrement décomposée, restructurée ou dupliquée au moment de l'exécution. Cela permet de faire exploser des personnages ou de tirer certains dessins comme projectiles.</p>

Fonctionnalité	Description
Paramètres TBG Renderer	La résolution, les palettes, les matériaux, les couleurs et les habillages peuvent être modifiés à partir de l'inspecteur ainsi que les scripts personnalisés de l'ObjetJeu. Le matériau et la couleur de SpriteRenderers peuvent également être définis individuellement par dessin, de sorte que vous pouvez modifier les couleurs des accessoires d'un personnage ou conférer des effets spéciaux à certains dessins grâce à des Shaders personnalisés.
Intégration du 2D Animation Package	Dans Harmony, les dessins déformés par les Game Bones seront convertis en bones HabillageSprite dans Unity à l'aide du kit d'animation « 2D Animation Package » d'Unity. Ce progiciel prend en charge la déformation de maillage haute performance multithreadée pour les SpriteRenderers ainsi que les poignées d'interface utilisateur pour le déplacement des Bones dans Scene View. La chaîne osseuse peut être animée de manière procédurale à l'aide du « IK Manager 2D » d'Unity pour ajouter des effets de visée, d'inclinaison, de secousses, d'ondulation, etc. après l'étape de mise à jour de l'animation.
Animations	<p>Toutes les données d'animation sont exportées d'Harmony par cheville, de sorte que toutes les rotations et échelles intermédiaires sont représentées dans la fenêtre Animation dans Unity. Les animations peuvent être modifiées une fois que leur ressource AnimationClip associée est dupliquée d'une sous-ressource TBG à une nouvelle ressource dans le projet, ce qui permet d'ajouter des événements et de fournir des courbes supplémentaires dans Unity.</p> <p>Les AnimationClips sont sélectionnés par le biais d'un AnimatorController, ce qui permet de passer d'une animation à l'autre, soit immédiatement, soit par une fusion lisse de transformations de chevilles.</p>
Intégration de ShaderGraph	Des Shaders graphiques compatibles sont fournis comme éléments du SDK. Ils peuvent être dupliqués et modifiés, ce qui permet une interface facile pour introduire des effets visuels personnalisés spécifiques au jeu sur vos personnages.

Améliorations de HarmonyRenderer (dossiers XML)

Fonctionnalité	Description
Aperçu d'HarmonyProject	Lors de la sélection d'un personnage dans le dossier des ressources Unity Asset, un aperçu du personnage apparaîtra dans la fenêtre d'aperçu. La fenêtre d'aperçu montre un aperçu des animations du personnage et vous

Fonctionnalité	Description
	aurez la possibilité de modifier les habillages sans avoir à faire glisser le personnage dans Scene View.

Problèmes résolus

Les problèmes suivants ont été résolus dans cette version du SDK Harmony Game :

HarmonyRenderer

- Auparavant, les ancres perdaient de la mémoire lors de la mise à jour des transformations d'ancre. Une mise à jour du plug-in HarmonyRenderer C++ résout ce problème.
- HarmonyRenderers a amélioré les performances de rendu. Il recevra désormais de nouveaux maillages du plug-in HarmonyRenderer C++ qui peuvent être transmis directement au backend C++ d'Unity à l'aide des nouvelles fonctionnalités de génération de maillage Unity. Cela réduit considérablement le temps nécessaire pour mettre à jour les animations sur les personnages HarmonyRenderer, ce qui permet d'avoir simultanément plus de personnages à l'écran.

Chapitre 1 : Exportation Harmony des données de jeu

Il existe deux pipelines principaux pour l'exportation de données à partir de Harmony à votre moteur de jeu :

- [Exportation Harmony des données de jeu](#) page 10
- [Exportation Harmony des données de jeu](#) page 10



Création d'une animation dans Harmony

Lors de la création de rigs de personnages et d'animations de jeu, il convient de réfléchir à quelques éléments avant de commencer. Les artistes et les programmeurs doivent travailler ensemble pour s'assurer que leurs exigences sont satisfaites.

- Pour quelles plateformes votre jeu sera-t-il créé? Windows, Mac, Mobile, iOS, PS, XBOX, etc.
- Quel moteur de jeu utiliserez-vous?
- Quel style d'animation sera utilisé pour créer l'apparence du jeu? Dessin à la main, pantins, textures, etc.

Ce ne sont là que quelques-unes des questions à se poser avant de commencer. Ces dernières ont toutes un impact sur la façon dont vous concevez, construisez et animez les personnages.

Par exemple, si vous planifiez un jeu mobile pour smartphones, vous voudrez probablement garder votre jeu en dessous de 50 Mo, afin qu'il puisse être téléchargé sans devoir être en Wi-Fi. Dans ce cas, le plus important est de créer des personnages efficaces avec des feuilles de sprites très serrées et de réutiliser une grande partie de l'animation pour que les fichiers soient de petite taille. Au cours du processus, vous devrez :

- Effectuer le rig et l'animation des personnages dans Harmony.
- Extraire les données Harmony.
- Importer les données Harmony dans le moteur de jeu.

N'oubliez pas que, si vous travaillez avec un moteur personnalisé, vous pouvez également traiter les données exportées Harmony et les utiliser dans un moteur de jeu personnalisé. Si vous avez besoin d'aide pour adapter les données de votre moteur, communiquez avec le support technique à l'adresse suivante :

store.toonboom.com/contact/support.

Si vous créez un jeu pour des consoles comme la PS ou la Xbox, vous pouvez créer des textures plus grandes. Il est possible que vous souhaitiez animer image par image, avec un pantin, ou les deux.

Si vous allez simplement exporter sur une séquence image par image, vous pouvez utiliser dans Harmony tous les outils sans limitations. Vous pouvez ensuite traiter une séquence d'images exportée dans une feuille de sprite.

Exportation de données de jeu brutes (exporter vers une feuille de sprite)



La méthode d'exportation de données de jeu brutes est appropriée lorsque vous souhaitez que la taille des fichiers soit la plus petite possible. Toon Boom vous permet de convertir des données directement à partir de votre plan Harmony pour les incorporer dans un moteur de jeu. Vous pouvez extraire les informations du squelette, les informations du dessin et les données d'animation des images clés, ainsi que les déformations (bones et articulations uniquement), la coupure, les nœuds de transparence et les colonnes de synchronisation.

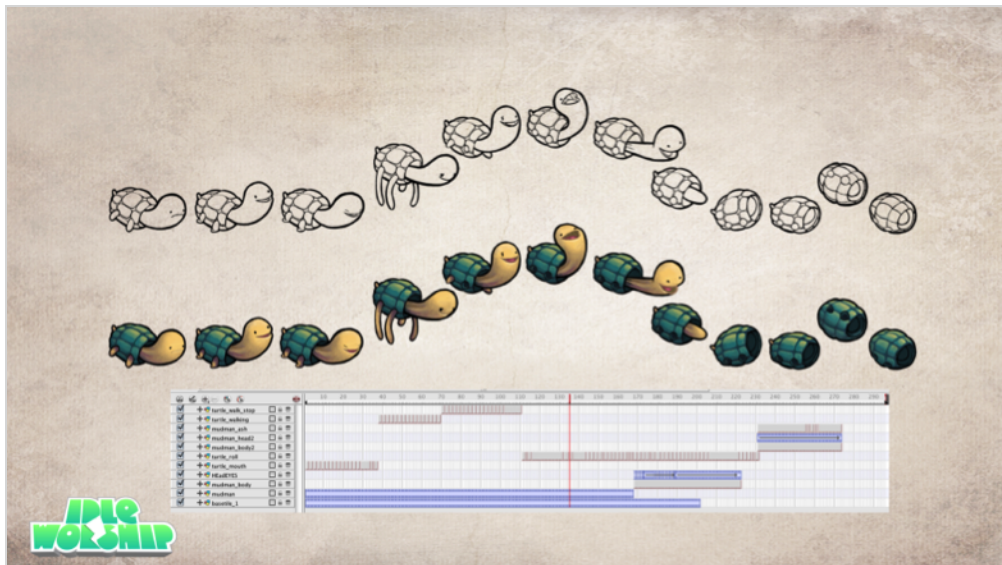
- **Avantage** : Il s'agit de l'exportation la plus légère, qui permet de conserver des fichiers de petite taille, ce qui est idéal pour les applications mobiles.
- **Inconvénient** : Vous êtes quelque peu limité dans les outils que vous pouvez utiliser dans Harmony. Vous pouvez utiliser des outils tels que le morphage ainsi que les déformateurs de courbes et d'enveloppes, mais vous devrez les assembler en dessins pour qu'ils soient interprétés correctement dans le moteur de jeu. Vous pouvez utiliser des effets de découpe (masquage), mais vous ne pouvez pas les mettre en cascade, ce qui signifie que vous ne pouvez pas en avoir plus d'un dans une chaîne hiérarchique. Les déformateurs Game Bone peuvent être utilisés sur votre rig sans avoir à l'assembler sur des dessins.

Cependant, même avec ces limitations, vous pouvez créer une très grande animation de marionnettes dans Harmony et extraire toutes les données compatibles. En déplaçant, en tournant, en mettant à l'échelle et en inclinant les différents calques de dessin, vous pouvez créer une animation d'aspect avancé.

Lorsque vous extrayez les données, vous aurez des feuilles de sprite qui contiennent uniquement les dessins des parties du corps utilisées dans votre fichier de plan Harmony. Vous pouvez également gérer plusieurs types d'animations, comme « inactive », « exécution » et « atteindre » tout en réutilisant le même squelette et les mêmes dessins.

Toon Boom a entièrement intégré cette solution au moteur de rendu de développement de jeux Unity. Si vous créez votre jeu dans Unity, vous disposez d'un pipeline transparent sans avoir à traiter à nouveau les données de votre moteur de jeu.

Exportation image par image (Exportation vers Easel JS)



Vous pouvez exporter à partir de Harmony à l'aide d'une séquence d'images, qui peut être recompilée dans une feuille de sprite.

Avantage : Vous pouvez utiliser n'importe lequel des outils dans Harmony quand vous réalisez une animation!

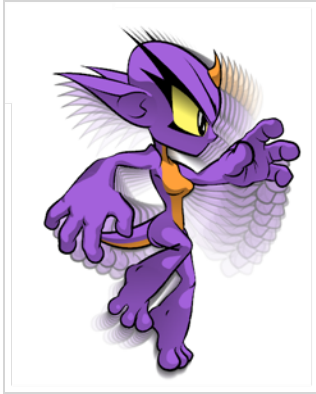
Inconvénient Ces feuilles de sprite peuvent se retrouver sous forme de grandes textures à traiter dans le jeu. Cela peut convenir à de nombreux jeux, comme les jeux sur console, mais lorsque vous développez des jeux pour des appareils mobiles, comme iOS ou Android, vous devez être très économe en espace de texture. En outre, les tailles de fichiers doivent être maintenues en dessous de 50 Mo pour faciliter le téléchargement sans avoir besoin d'être connecté au Wi-Fi.

Un script disponible dans Harmony vous permet d'exporter votre animation image par image vers le moteur de jeu EaselJS. EaselJS est un moteur de jeu compatible avec le web en javascript. Il utilise une structure de données json pour mapper chaque image à une entrée dans la feuille de sprite. La structure de données est facile à comprendre et peut être adaptée ou convertie en un autre moteur de jeu si nécessaire.

Pour exporter vers EaselJS, vous devez ajouter le script dans la barre d'outils Scripting (TB_ExportToEaselJS). Ce script est inclus dans Harmony. Voir le guide de script pour plus d'informations sur la façon d'ajouter le script à votre barre d'outils script ainsi que les paramètres d'interface.

Chapitre 2 : À propos de Création de ressources de jeu

Dans cette section, vous trouverez des directives pour le rigging, la déformation, le découpage et de nombreux conseils d'animation utiles.



Directives pour le rigging de jeu

Ce qui suit est une liste de directives générales à garder à l'esprit lors du rigging de votre personnage. Lorsque vous planifiez le rig de personnage pour votre jeu, gardez à l'esprit le style du personnage et créez votre palette de couleurs personnalisée. Cependant, il existe certaines limitations à prendre en compte si vous prévoyez d'extraire les données du jeu:

- Définissez votre plan Harmony avec une résolution carrée (ex. 1024 x 1024). Cette opération peut être effectuée dans la boîte de dialogue Paramétrage du plan- voir le Guide de référence.
- Dessinez le motif sur les calques Dessin au trait et Motif de couleurs uniquement. Les informations contenues dans la superposition et le sous-calque ne seront pas exportées vers Unity. Il peut toutefois être utilisé en toute sécurité comme calque de référence, mais pas pour tout ce qui sera exporté.
- Repositionnez les calques dans l'espace Z si vous avez besoin de réorganiser les calques. Cependant, les décalages Z significatifs ne sont pas pris en charge dans un rig de personnages.
- Faites de chaque calque de votre moteur de jeu un plan distinct dans Harmony. Si vous avez deux personnages à des profondeurs différentes, placez-les dans des fichiers de plan séparés.
- Le rig de personnages en groupes au niveau racine de votre plan Harmony s'affiche sur un seul plan dans Unity, mais utilise des jeux de données de feuille de sprite et d'animation séparés. Gardez cela à l'esprit pour les plans dans lesquels vous pouvez avoir plus d'un personnage interagissant les uns avec les autres.
- N'utilisez pas d'espace 3D. Si vous souhaitez configurer les éléments dans l'espace 3D, vous pouvez le faire lorsque vous arrivez à votre moteur de jeu.
- Définissez vos points de pivot sur les calques Cheville à l'aide de l'outil Rotation pour définir le pivot sur l'ensemble du calque. Les pivots de cheville sont recommandés par rapport aux pivots de dessin. Vous devez également définir les points de pivot sur vos calques de dessin, même si vous ne les animez pas, car cela vous permettra de récupérer les informations plus tard dans le moteur de jeu si vous devez ajouter une ancre sur un calque de dessin.
- N'utilisez pas le Morphage. Ceci n'est pas encore pris en charge dans les moteurs de jeu.
- Le morphage et les déformations peuvent être utilisés, puis assemblés sur des dessins individuels. Ces dessins apparaîtront ensuite sous forme de nouveaux dessins dans votre feuille de sprite. Gardez-vous de le faire trop souvent, car cela augmentera votre espace de texture!
- Vérifiez l'endroit où vous placez votre personnage avant de l'exporter. Le pivot principal de votre objet de jeu exporté sera le centre de votre plan Harmony (0,0).
- Assurez-vous de disposer d'un Affichage à la fin de votre hiérarchie.
- Lors du rigging, gardez à l'esprit que certains nœuds ne sont pas pris en charge par l'exportateur. En conséquence, la structure des nœuds ne serait pas bien interprétée par les moteurs de jeu. Il est donc recommandé d'éviter d'utiliser des structures de nœuds complexes. Pour ce faire :
 - Construisez votre personnage à partir de dessins attachés à des chevilles. Les Chevilles peuvent ensuite être connectées à d'autres Chevilles pour former des hiérarchies.
 - Toutes les Chevilles doivent être connectées à une seule Cheville principale en haut de votre Node View

- Les dessins doivent généralement être reliés aux Composites pour déterminer l'ordre des dessins affichés à l'écran
- Si l'ordre des dessins doit être modifié pendant l'animation, vous pouvez faire varier la profondeur Z pour modifier le dessin qui s'affiche en premier
- Si vous n'arrivez pas à faire fonctionner vos animations avec seulement les Dessins et les Chevilles, il existe d'autres nœuds que vous pouvez utiliser et qu'il est possible d'exporter :
 - **Couteaux** - En général, ne coupez qu'un seul dessin contre un autre dessin (par ex. couper une face aux limites d'une tête). Vous ne pouvez pas « enchaîner » la sortie d'un Couteau dans l'entrée d'un autre Couteau.
 - **Déformations par Game Bone** - En général, ne déformez qu'un seul dessin. Le squelette de type Bone peut se ramifier. Les sorties cinématiques peuvent être utilisées attacher des dessins aux extrémités des bones (par ex. pied au bout d'une jambe)

**REMARQUE**

vous êtes toujours en mesure d'utiliser des structures de nœud plus compliquées dans « bake_groups ». [Voir Utilisation de groupes d'assemblage.](#)

Garder ces conseils à l'esprit vous permettra de créer un personnage de jeu 2D compact et efficace dans Harmony tout en profitant de tous les outils remarquables.

Voici quelques mesures que vous devriez prendre :

- Créez une hiérarchie de relations parent-enfant simple dans la Timeline View.
- Utilisez des calques de cheville pour contenir des données d'animation d'image clé, définies sur Position séparée.
- Utilisez des calques de dessin pour dessiner, en créant de nouveaux dessins si nécessaire.
- Utilisez l'outil Rotation pour définir les points de pivot sur les calques de chevilles.
- Nommez correctement vos calques de sorte que, si vous devez récupérer le point de pivot d'un calque spécifique plus tard dans le moteur de jeu, vous pourrez facilement reconnaître le calque dont vous avez besoin. Si vous avez un groupe A de niveau supérieur, qui contient un groupe enfant (groupe B) et que le calque de dessin est enfant du groupe B, le calque de dessin est exporté en tant que A_B_CalqueDessin.
- Définissez vos ancrs à l'endroit où vous souhaitez que vos chevilles apparaissent dans Unity en tant que transformations.

Vous pouvez utiliser n'importe lequel des outils de dessin que vous souhaitez : outils Crayon et Pinceau, lignes texturées, zones de couleurs unies et dégradés. Chaque dessin individuel sera rendu et assemblé ultérieurement dans une feuille de sprite.

**REMARQUE**

Le moteur de jeu Unity ne prenant pas en charge les caractères Unicode, il est recommandé d'éviter de l'utiliser dans les plans destinés aux jeux.

Directives pour les déformations de jeu

Harmony peut exporter des déformations au format XML du moteur de jeu. Seules les hiérarchies composées de Game Bones peuvent être exportées correctement. En outre, des sorties cinématiques ont été implémentées pour compléter vos Game Bones.

Dans Harmony, les déformations sont accompagnées d'un large éventail de caractéristiques. Toutefois, certaines de ces fonctionnalités ne sont pas compatibles avec le SDK du moteur de jeu et ne doivent pas être utilisées pour les jeux. Par conséquent, les déformateurs Game Bones ne prennent pas en charge les fonctionnalités standard de déformation par Bone suivantes :

- Zones d'influence
- Déformations de courbes et d'enveloppes
- Plusieurs poses dans la hiérarchie de déformation
- Animation de chevilles de dessin sous les groupes de déformation

La déformation dans le SDK du moteur de jeu ne se comporte pas exactement comme la déformation par Bone dans Harmony.

Pour se conformer à la plupart des moteurs de jeu et maintenir des calculs rapides, le SDK met en œuvre un algorithme d'habillage de base linéaire pour mixer les bones au niveau des articulations. Vous remarquerez peut-être quelques différences en fonction de la courbure des articulations utilisées. Cela dit, par rapport aux déformations par Bones régulières, les Game Bones dans Harmony ressemblent le plus à ce que vous verrez une fois importé dans Unity.



REMARQUE

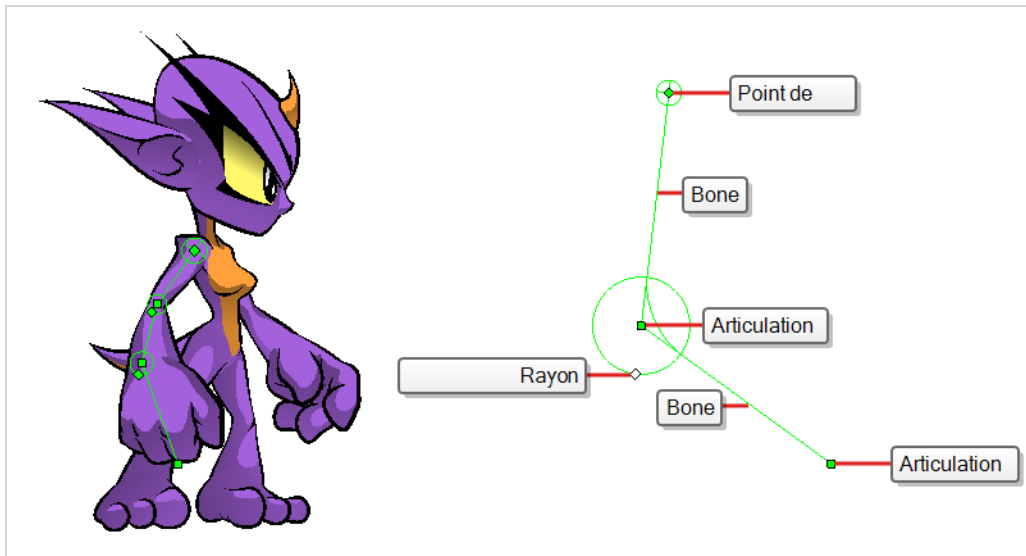
* Non disponible actuellement dans l'implémentation Cocos2d-x du SDK du moteur de jeu.

About Game Bone Deformations

T-RIG-007-003

La déformation Game Bone est très similaire à la déformation Bone. Elle vous permet de créer une structure de type Bone dans laquelle chaque partie est solide, mais avec des articulations flexibles. Ceci est surtout utile pour animer les membres d'un personnage, tels que les bras ou les jambes, ou d'autres parties qui peuvent être articulées telles que le torse ou les doigts. Par exemple, une déformation Game Bone peut être utilisée pour articuler un bras qui est fait d'un seul dessin, de sorte que le haut du bras et l'avant-bras peuvent être déplacés indépendamment, sans avoir à dessiner le haut du bras et l'avant-bras sur différents calques. Harmony déformera le dessin pour le rendre articulé. Les différentes parties d'une déformation par Game Bone peuvent être pivotées autour de leur articulation, étendues et raccourcies, vous donnant les mêmes capacités d'animation des articulations sur différents calques, sans avoir à vous soucier du détachement des pièces, des points de pivot ou des contours d'écèlement.

La déformation de Game Bone diffère de la déformation par Bone dans laquelle elle est optimisée pour les moteurs de jeu tels que Unity. Par conséquent, elle n'est généralement utilisée que pour le développement de jeux et non dans les productions animées. Les différences entre les déformations par Bone et les déformations par Game Bone sont que les déformations par Game Bone n'ont pas de Région d'Influence. Les plis d'articulation ont également l'air légèrement plus arrondis.



Directives pour les découpes de jeu

La fonction de découpe, ou masque, est utilisée dans Harmony pour découper des dessins avec des formes personnalisées. Le SDK du moteur de jeu implémente aussi bien des découpes que des découpes inversées avec les limitations suivantes :

- Pour un sprite, une seule découpe de dessin peut être appliquée lors du rendu. Cela s'applique également à un composite de plusieurs dessins en cache. Le SDK du moteur de jeu n'utilisera que le premier dessin en cache pendant le rendu et rejettera les autres.
- Un dessin déformé ne peut pas être découpé, mais un dessin découpé peut être déformé.



REMARQUE

Non disponible actuellement dans l'implémentation Cocos2d-x du SDK du moteur de jeu.

Création de notes de métadonnées

Lorsque vous créez des ressources pour votre jeu dans Harmony, vous pouvez prendre des notes sur le plan ou des parties spécifiques du personnage ou des accessoires pour le programmeur. Ces notes incorporées qui seront exportées avec vos ressources vers Unity sont appelées métadonnées.

Comment accéder à la vue Éditeur de métadonnées

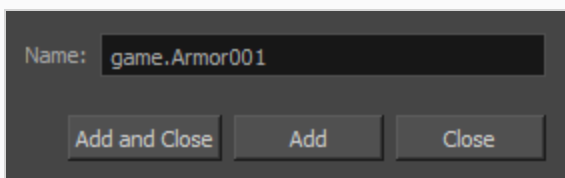
- Dans le coin supérieur droit d'une vue, cliquez sur Ajouter une vue **+** et sélectionnez **Éditeur de métadonnées**.
- Dans le menu supérieur, sélectionnez **Fenêtres > Éditeur de métadonnées**.

Comment créer des métadonnées de plan

1. Dans la vue Éditeur de métadonnées, dans la section Métadonnées de plan, cliquez sur le bouton plus **+** pour créer une entrée de métadonnées.

La boîte de dialogue Ajouter des métadonnées s'affiche.

2. Dans la boîte de dialogue Ajouter des métadonnées, entrez le nom de votre nouvelle entrée de plan.



REMARQUE

Pour toute information d'ancrage ou d'accessoire créée dans Éditeur de métadonnées, la convention de nommage doit toujours utiliser le préfixe « game ». Le terme « game » est reconnu automatiquement par Unity en tant qu'information de métadonnées.
Exemple : *game.Armor001*.

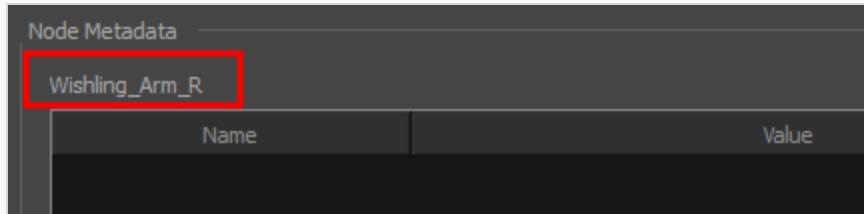
3. Cliquez sur **Ajouter et fermer** si vous avez l'intention d'ajouter une seule entrée. Cliquez sur **Ajouter** si vous avez l'intention d'ajouter plusieurs entrées. Continuez à ajouter plusieurs entrées, en cliquant sur **Ajouter** après chacune d'elles. Cliquez sur **Fermer** lorsque vous avez terminé.
4. Dans la vue Éditeur de métadonnées, double-cliquez sur le champ de valeur de la première entrée pour pouvoir la modifier.
5. Entrez les informations de valeur pour cette entrée.
6. Continuez à ajouter des informations de valeur pour toutes vos entrées.

Ces informations seront exportées avec vos ressources Harmony. Une fois dans Unity, les métadonnées de plan s'affichent dans Inspector View > Métadonnées lorsque la ressource est sélectionnée dans la vue Hiérarchie.

Comment créer des métadonnées de nœud

1. Dans la Timeline View, cliquez sur le calque auquel vous souhaitez attacher des métadonnées.

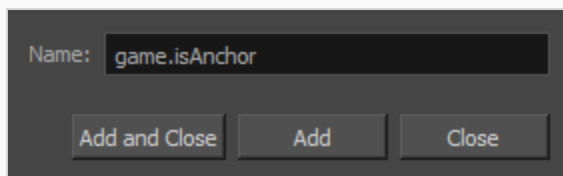
Le nom du calque s'affiche en haut de la section Métadonnées du nœud.



2. Dans la section Métadonnées du nœud, cliquez sur le bouton plus **+** pour créer une entrée de métadonnées.

La boîte de dialogue Ajouter des métadonnées s'affiche.

3. Dans la boîte de dialogue Ajouter des métadonnées, entrez le nom de votre nouvelle entrée de nœud.



REMARQUE

Pour toute information d'ancrage ou d'accessoire créée dans Éditeur de métadonnées, la convention de nommage doit toujours utiliser le préfixe « game ». Le terme « game » est reconnu automatiquement par Unity en tant qu'information de métadonnées.

Exemple : *game.Armor001*.

4. Cliquez sur **Ajouter et fermer** si vous avez l'intention d'ajouter une seule entrée. Cliquez sur **Ajouter** si vous avez l'intention d'ajouter plusieurs entrées. Continuez à ajouter plusieurs entrées, en cliquant sur **Ajouter** après chacune d'elles. Cliquez sur **Fermer** lorsque vous avez terminé.
5. Dans la vue Éditeur de métadonnées, double-cliquez sur le champ de valeur de la première entrée pour pouvoir la modifier.
6. Entrez les informations de valeur pour cette entrée.
7. Continuez à ajouter des informations de valeur pour toutes vos entrées.

Ces informations seront exportées avec vos ressources Harmony, plus précisément, ces informations seront reliées au calque sélectionné. Une fois dans Unity, les métadonnées de plan s'affichent dans Inspector View > Métadonnées lorsque la ressource est sélectionnée dans la vue Hiérarchie.

Conseils pour l'animation de jeux

Lors de l'animation de jeux, selon le type de jeu que vous créez, vous devrez peut-être limiter votre animation. Par exemple, si vous créez des jeux mobiles et que vous souhaitez que les fichiers soient de petite taille et que la lecture soit rapide sur tous les appareils, limitez-vous à une simple animation par images clés avec le moins de changements de dessin possible. Si vous créez des jeux sur console, vous avez la liberté de créer plus de dessins et d'avoir une plus grande complexité. Vous pouvez explorer avec vos développeurs les limites des plates-formes vers lesquelles vous exportez et ce que votre moteur de jeu prend en charge.

Voici quelques conseils pour une animation efficace et allégée :

- Utilisez principalement des transformations, telles que le déplacement, la rotation, la mise à l'échelle et l'inclinaison.
- Créez des échanges de dessin supplémentaires si nécessaire.
- Si vous utilisez des déformateurs de courbes et d'enveloppes ou le morphage, vous devrez assembler les dessins pour l'exportation. Faites preuve de prudence lorsque vous faites cela, car vous voudrez peut-être garder un petit nombre de dessins. N'assemblez pas une séquence entière, juste des dessins sélectionnés. Vous n'avez pas besoin d'assembler les déformateurs Game Bone.
- Plus les dessins sont grands dans la Drawing View, plus ils occuperont de pixels dans la taille de texture sur la feuille de sprite. Lors de la configuration de votre rig, assurez-vous de ne pas mettre à l'échelle les calques individuels à l'aide d'une image clé avec l'outil Transformer. Si vous souhaitez augmenter ou diminuer l'échelle des choses, utilisez l'outil Sélectionner. Cela permettra de conserver la même taille relative sur la feuille de sprite. Lorsque vous exportez les feuilles de sprite, dans le script, vous pouvez également définir la résolution de la feuille de sprite afin que les dessins puissent être réduits pour les appareils plus petits.
- Seuls les dessins exposés dans le plan seront exportés vers la feuille de sprite. Par exemple, si vous avez 10 dessins dans la vue Library, mais que seuls deux d'entre eux s'affichent dans votre plan, seuls ces deux dessins sont exportés. Cela permet de garder la feuille de sprite aussi compacte que possible.

Animation de séquences multiples

Vous aurez probablement plusieurs animations pour vos personnages. Par exemple, une séquence inactive, une séquence d'exécution, une séquence d'action, etc. Vous devez travailler dans une structure spécifique afin de pouvoir exporter toutes ces animations vers une seule feuille de sprite.

Il existe deux workflows différents que vous pouvez utiliser :

- Workflow 1 : Plans séparés
- Workflow 2 : Séparation à l'aide de marqueurs de plan

Workflow 1 : Plans séparés

Tout d'abord, créez un fichier de plan avec le nom du personnage, tel que **Space Duck**. C'est le fichier où vous pouvez créer ou importer votre rig de jeu. Dans le menu supérieur, sélectionnez **Fichier > Enregistrer en tant que nouvelle version**, et donnez à cette nouvelle version le nom de l'animation. Par exemple **Idle**.

Chaque fois que vous devez effectuer une nouvelle animation en utilisant le même personnage, effectuez une sauvegarde sous une nouvelle version. Au final, vous obtiendrez peut-être un résultat comme celui-ci :

Plan : Space Duck

Versions :

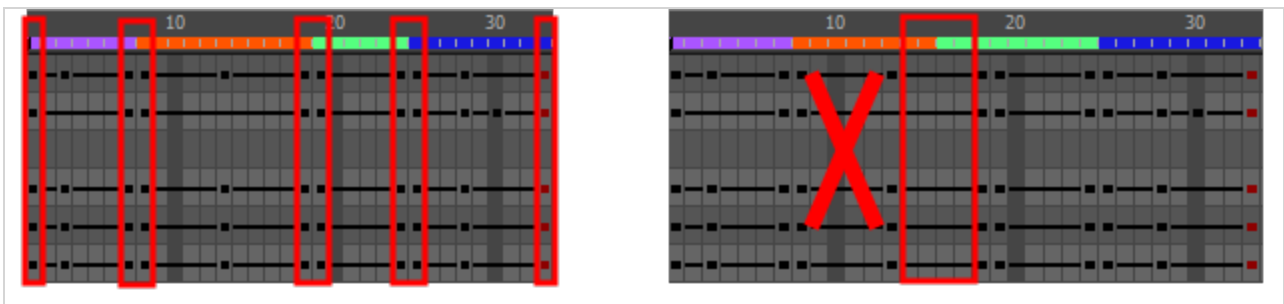
- Inactif
- Exécuter
- Atteindre
- Tirer

Lorsque vous exécutez le script d'exportation, les dessins du plan courant sont exportés dans le dossier d'exportation. Cela vous permettra également de savoir s'il existe d'autres versions du plan qui ont déjà été exportées vers ce dossier. Si c'est le cas, il recompilera la feuille de sprite pour inclure tous les dessins de toutes les animations. Cela permet la réutilisation maximale possible des dessins.

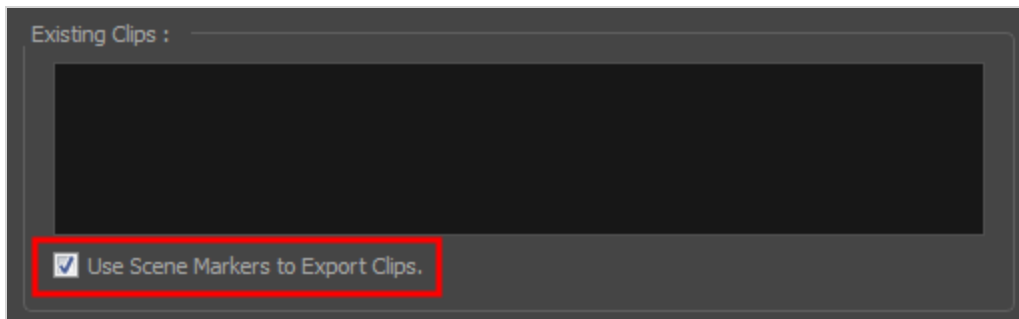
Workflow 2 : Séparation à l'aide de marqueurs de plan

Vous pouvez également créer toutes vos animations de personnages dans un seul plan, l'un après l'autre, comme inactif, exécuter, atteindre et tirer. Ensuite, utilisez des marqueurs de plan pour marquer et séparer les animations individuelles.

Lorsque vous marquez des plages d'images d'animation individuelles, assurez-vous qu'elles commencent et se terminent par une image clé. Ne créez pas de marqueurs de plan pour une série d'images qui commence ou se termine au milieu d'un mouvement interpolé.



Lorsque vous exportez votre feuille de sprite, dans la boîte de dialogue Exporter vers une feuille de sprite, veuillez à cocher l'option **Utiliser des marqueurs de plan pour exporter des clips**. Ceci est activé par défaut.



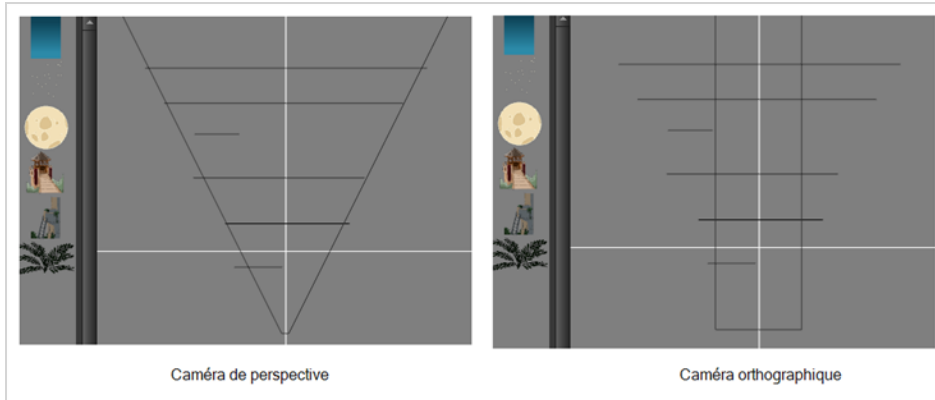
Les clips animés sont divisés et répertoriés dans le fichier stage.xml de la même manière qu'ils apparaîtraient si vous aviez exporté chaque séquence animée de plans distincts vers le même emplacement de fichier.

À propos de la Caméra orthographique

Dans Harmony, il existe deux types de caméras disponibles :

- Perspective
- Orthographie

La caméra orthographique est spécifique au pipeline de jeu. Elle permet de changer le type de caméra de perspective à orthographique. Elle devient une caméra sans points de fuite. Cela signifie qu'il n'y a plus de perspective dans la Camera View. Les objets, lorsqu'ils sont déplacés sur l'axe Z, ne changent pas de taille ou d'échelle.



La caméra orthographique peut être définie dans la boîte de dialogue Paramètres du plan. Afin de créer des plans avec la caméra orthographique par défaut, vous devez créer une nouvelle résolution de plan personnalisée.



Chapitre 3 : À propos de Exportation vers Unity

Si vous avez un moteur de jeu personnalisé, vous pouvez récupérer les données Harmony exportées et appliquer votre processus habituel. Vous pouvez également modifier le script d'exportation pour l'adapter à votre convention.



Avant d'exporter un plan :

- Veillez à définir votre affichage sur **Affichage** et non sur Afficher tout.
- Veillez à sauvegarder votre plan. Harmony effectue l'exportation en fonction des fichiers tvg, toutes les mises à jour non enregistrées que vous avez ajoutées ne seront pas exportées.
- Définissez votre plan Harmony sur une résolution carrée (ex. 1024 x 1024). Sélectionnez Plan > Paramètres du plan.

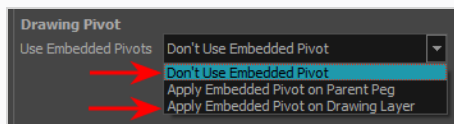
Ancrages


Avant d'exporter vers Unity, vous devez considérer ce qu'il faut définir comme ancrs.

Une ancre est un point sur le personnage que vous souhaitez référencer dans le moteur de jeu pendant l'animation du personnage.

Comment définir une ancre

1. Créez un nouveau calque de dessin et renommez-le en fonction de l'ancre que vous créez.
2. Dans Propriétés du calque de votre calque de dessin, assurez-vous que l'option de pivot est définie sur **Ne pas utiliser le pivot incorporé** ou **Appliquer le pivot incorporé sur le calque de dessin**. L'application du pivot intégré sur la cheville parent ne fonctionne pas bien sur Unity et vous perdrez vos informations de pivot.



3. Une fois cela fait, vous pouvez utiliser n'importe lequel des outils d'animation pour repositionner le point de pivot de votre calque, tel que l'outil Rotation dans la barre d'outils Animation avancée. Vous devriez être en mesure de voir les coordonnées de pivot à partir de Propriétés du calque de votre Calque de dessin. Si le pivot n'est pas repositionné, son emplacement par défaut sera au centre de votre plan Harmony (0,0).
4. Une fois configuré, sélectionnez le calque Dessin et cliquez sur Basculer l'ancre  dans la barre d'outils Jeux. Vous pouvez également effectuer une sélection multiple pour définir une ancre sur plusieurs calques de dessin simultanément.

Le calque Dessin est surligné en rouge dans les vues Timeline et Node. Votre ancre peut désormais être utilisée pour fixer un accessoire.

Exportation de feuilles de sprite

La fenêtre Exporter vers des feuilles de sprite est exportée sous plusieurs résolutions, générant plusieurs fichiers .xml et une ou plusieurs feuilles de sprites en fonction du nombre de résolutions de sprite que vous avez définies.

Cela enregistre différentes animations du même personnage sous le même nom. Par exemple, s'il existe une animation inactive, exécution et atteindre, celles-ci doivent toutes partager le même Nom de ressource. Vous pouvez la considérer comme la collection globale d'animations. À l'intérieur se trouvent les différentes versions de plans enregistrés dont vous pouvez réutiliser les dessins pour toutes les animations de ce jeu de personnages. Chaque version du plan sera affichée en tant qu'élément dans la liste.

Lorsque vous exportez une animation, seuls les dessins utilisés dans ce plan sont exportés. Tous les dessins sont d'abord exportés individuellement, puis regroupés en atlas dans une feuille de sprite.

Si vous avez enregistré plusieurs animations sous le même nom de ressource (c'est-à-dire CanardDeL'Espace : exécuter, inactif), puis il regroupera à nouveau en atlas la feuille de sprite pour inclure tous les dessins de toutes les animations dans ce dossier, en créant un nouveau fichier d'animation, mais en réutilisant le même squelette.



REMARQUE

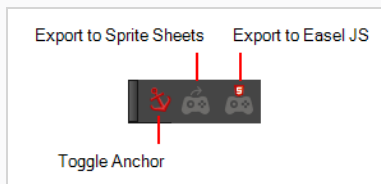
Les programmeurs seront peut-être intéressés de noter que les données XML exportées par Harmony peuvent être converties ensuite en une structure de données binaires plus optimisée. Pour convertir un fichier XML au format binaire, utilisez l'utilitaire Xml2Bin. Cet utilitaire convertit la structure de données XML générée par le logiciel Toon Boom Harmony en une structure de données binaires compressées. Cet utilitaire est disponible dans le SDK de jeu sous /HarmonyGameSDK/Plugins/.


- Plugins/Mac : Binaire précompilé pour Mac OSX.
- Plugins/Windows\x86 : Binaire précompilé pour Windows.
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin : sources Xml2Bin.
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin/proj.mac.Xml2Bin.xcodeproj : Projet XCode pour Mac OSX.
- Samples/HarmonyGameSDKSource/Utils/Xml2Bin/proj.win32/Xml2Bin.sln : Solution Visual Studio 2010 pour Windows.

Le code C++ qui gère la structure de données peut être réutilisé et analysé dans votre propre code si vous souhaitez l'intégrer à d'autres moteurs de jeu.

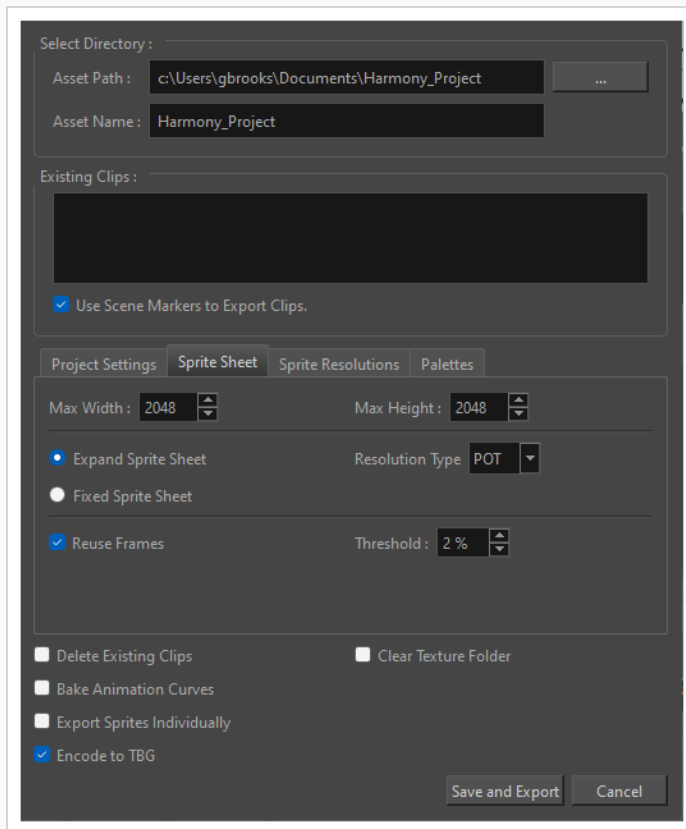
Comment exporter des feuilles de sprite

1. Ajoutez la barre d'outils Jeux en sélectionnant **Fenêtres > Barres d'outils > Jeux**.



2. Exécutez le script en cliquant sur le bouton Exporter vers les feuilles de sprite  dans la barre d'outils Jeux.

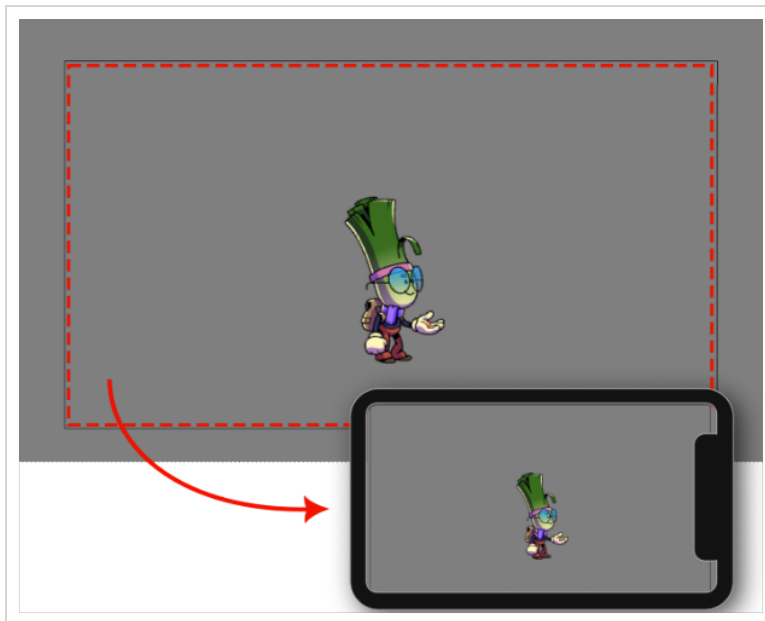
La fenêtre Exporter vers des feuilles de sprite s'ouvre.



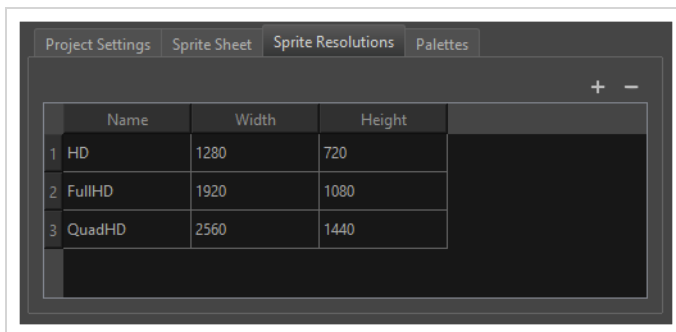
3. Définissez le chemin d'enregistrement sur le dossier correct de votre projet Unity si vous souhaitez qu'il soit mis à jour automatiquement. Sinon, vous pouvez enregistrer n'importe où, puis transférer l'animation dans votre projet Unity ou dans tout autre moteur de jeu que vous souhaitez utiliser.
4. Définissez les préférences de votre feuille de sprite.
5. Cliquez sur **Exporter**.

Résolutions de sprite

Vous pouvez définir le nombre de pixels sur un sprite en fonction de la résolution choisie pour votre jeu. Votre résolution cible peut être basée sur le matériel que vous ciblez, tel qu'un écran de télévision ou de téléphone.




Une nouvelle feuille de sprite sera créée pour chaque résolution cible spécifiée dans l'onglet Résolutions de la fenêtre Exporter vers une feuille de sprite (XML). Pour accéder à cette fenêtre, vous devrez activer la barre d'outils Jeux en sélectionnant **Fenêtres > Barres d'outils > Jeux**.



Veuillez noter que la taille du sprite sera la taille du personnage lorsqu'il a été créé dans Harmony.

Comment configurer plusieurs résolutions de sprite

1. Activez la barre d'outils Jeux en sélectionnant **Fenêtres > Barres d'outils > Jeux**.
2. Sélectionnez Exporter vers des feuilles de sprite 🗑️. La fenêtre Exporter vers une feuille de sprite (XML) s'ouvre.
3. Dans la boîte de dialogue Exporter vers une feuille de sprite (XML), sélectionnez l'onglet Résolutions de sprite. Cet onglet répertorie toutes les résolutions cibles courantes.

4. Cliquez sur l'icône  Ajouter. Cela créera une nouvelle ligne pour une nouvelle résolution cible.
5. Saisissez le nom, la largeur et la hauteur de votre nouvelle résolution cible.

Les résolutions cibles par défaut sont HD, FullHD et QuadHD. Les résolutions d'écran répertoriées représentent un appareil cible sur auquel le jeu pourrait être joué.

Vous êtes libre d'ajouter ou de supprimer manuellement des résolutions de cette liste en fonction de votre résolution cible et de votre matériel. Une fois importée dans le moteur de jeu, il sera possible de sélectionner la résolution affichée en modifiant la feuille de sprite rendue.



REMARQUES

- Les résolutions de sprite sont affectées par l'échelle des traits de dessin.
- La mise à l'échelle des dessins ou de leurs chevilles n'affectera pas la résolution du sprite de sortie.
- Si vous souhaitez augmenter la résolution du sprite, vous pouvez mettre à l'échelle les calques enfants d'un dessin.

Utilisation de Bake_Groups

Un Bake_Group consiste un regroupement de nœuds de dessins et d'effets qui peuvent être exportés comme un seul dessin unifié dans une feuille de sprite.

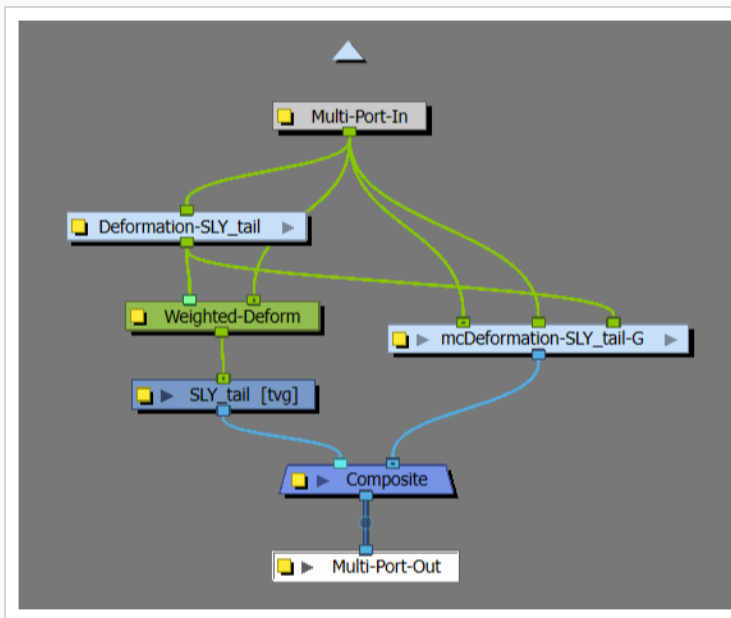
Comment créer un groupe assemblé

1. Regroupez les calques que vous allez assembler. Il est recommandé de regrouper les calques à partir de la Node View. .



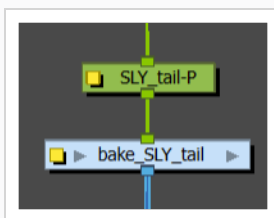
REMARQUE

Un bake_group ne doit avoir qu'une seule entrée et qu'une seule sortie. Cela signifie généralement qu'un nœud Composite doit être contenu dans le bake_group pour que l'exportation fonctionne correctement. Considérez également quelles chevilles sont nécessaires pour guider les dessins, car toutes les chevilles doivent finalement provenir d'une seule cheville extérieure au groupe.



2. Dans le montage, ajoutez le préfixe « bake_ » au début du nom du groupe.

Consultez l'exemple ci-dessous pour savoir à quoi doit ressembler un nœud de groupe avec le préfixe « bake_ ». Le « groupe » intitulé « hips » (Hanches) doit devenir « bake_hips » pour que l'assemblage s'applique.



Si le préfixe « bake_ » est inclus dans le nom du groupe, le groupe sera analysé pour les images contenant des effets, des mouvements et des substitutions de dessin uniques, et chaque image unique sera exportée en tant que sprite dans la feuille de sprite.

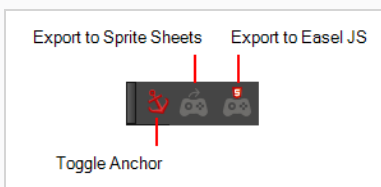
De nouvelles animations de dessin pour ces sprites générés seront créées et fournies au moteur de jeu. Le moteur de jeu n'aura accès à aucune des structures à l'intérieur de ce groupe, mais traitera plutôt l'ensemble du groupe comme s'il s'agissait d'un seul dessin comprenant des substitutions de dessin.

Exportation vers Easel JS

La fenêtre Exporter vers Easel JS vous permet d'aplatir une séquence d'images de votre animation. Même si vous avez une marionnette entièrement configurée ou un seul calque de dessin avec votre séquence d'animation, le résultat sera toujours une sortie aplatie de chaque image, regroupée dans votre feuille de sprite. Cela permet plus de flexibilité et de liberté de travail car vous avez accès à tous les outils ou modules d'effets que vous souhaitez utiliser. Cependant, cela peut produire des fichiers plus lourds en fonction de la longueur, de la complexité et de la taille d'exportation de votre animation.

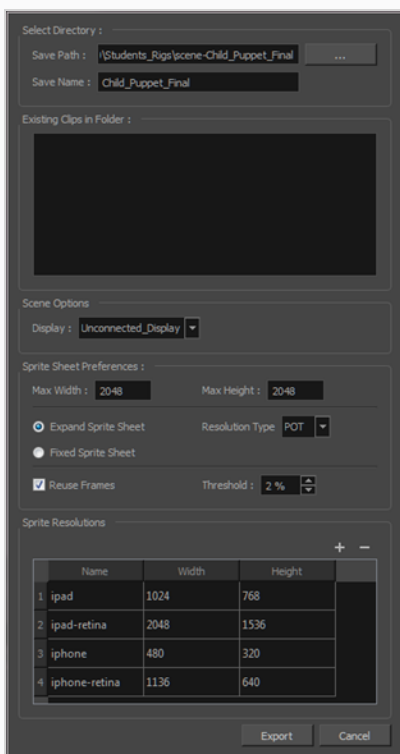
Comment exporter vers Easel JS

1. Ajoutez la barre d'outils Jeux en sélectionnant **Fenêtres > Barres d'outils > Jeux**.



2. Exécutez le script en cliquant sur le Exporter vers Easel JS 🎮 dans la barre d'outils Jeu.

La fenêtre Exporter vers le Easel JS s'ouvre.



3. Définissez le chemin d'enregistrement sur le dossier correct de votre projet Unity si vous souhaitez qu'il soit mis à jour automatiquement. Sinon, vous pouvez enregistrer n'importe où, puis transférer l'animation dans votre projet Unity ou dans tout autre moteur de jeu que vous souhaitez utiliser.
4. Définissez les préférences de votre feuille de sprite.

5. Cliquez sur **Exporter**.

Variations de palettes

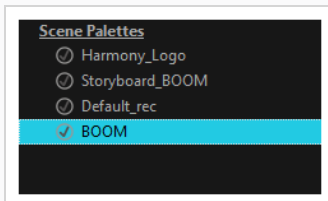
Il est important de créer des versions de personnages avec des différences de palette lorsque vous prévoyez d'ajouter la personnalisation des personnages comme une fonctionnalité de votre jeu. Dans Harmony, vous pouvez créer plusieurs palettes pour un personnage afin qu'il puisse avoir des apparences différentes dans votre jeu.


Dans Harmony, une fois que vous avez créé votre personnage et ses palettes, vous pouvez exporter le personnage vers des feuilles de sprite pour chaque palette.



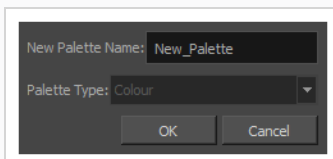
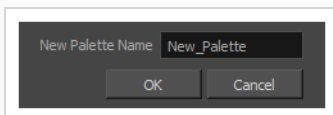
Comment créer des variations de palette pour un personnage de jeu

1. Dans la Colour View, sélectionnez une palette à cloner.



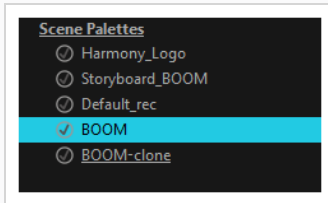
2. Effectuez l'une des opérations suivantes : Dans le menu Couleur, sélectionnez **Palettes > Cloner** ou cliquez avec le bouton droit de la souris et sélectionnez **Cloner**.
 - Cliquez avec le bouton droit sur la palette et sélectionnez **Cloner**.
 - Ouvrez le menu Couleur , puis sélectionnez **Palettes > Cloner**.

La boîte de dialogue Cloner la palette s'ouvre



3. Dans le champ **Nom de la nouvelle palette**, saisissez le nom de la nouvelle palette. Par défaut, le nom de la palette est basé sur le nom de la palette d'origine suivi de « -clone ». La liste des palettes au-dessus de ce champ répertorie les palettes qui existent déjà à l'emplacement sélectionné. Assurez-vous d'entrer un nom qui n'est pas déjà utilisé par une autre palette.
4. Cliquez sur **OK**.

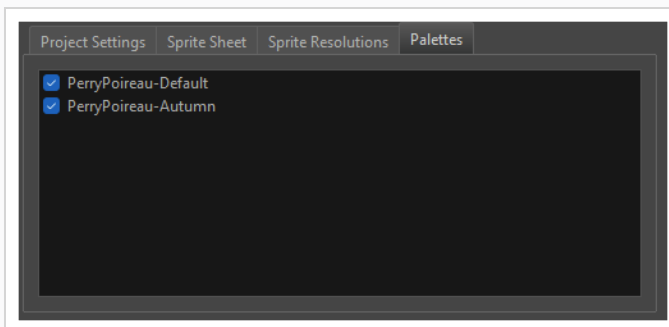
La palette clonée apparaît dans la liste des palettes.



Toutes les palettes de plan apparaîtront dans l'onglet Palettes de la fenêtre Exporter vers une feuille de sprite.

Comment exporter une feuille de sprite pour chaque variation de palette

1. Activez la barre d'outils jeu en effectuant l'une des opérations suivantes :
 - Dans le menu supérieur, sélectionnez Fenêtres > Barres d'outils > Jeu.
 - Cliquez sur le bouton droit de la souris sur n'importe quelle barre d'outils existante et sélectionnez **Jeu**.
2. Dans la barre d'outils Jeu, sélectionnez Exporter vers des feuilles de sprite 🎮.
3. Ouvrez l'onglet Palette. Vous verrez toutes les palettes de votre projet et leurs clones.



REMARQUE

Seules les palettes et leurs clones s'afficheront. Les doublons de palette ne s'afficheront pas.

4. Sélectionnez les noms des palettes que vous souhaitez exporter. Chaque palette cochée sera exportée avec le personnage comme sa propre feuille de sprite.

5. Exportez votre feuille de sprite.

Une fois le personnage importé dans Unity, il sera possible de sélectionner la palette affichée en modifiant le rendu de la feuille de sprite.

Chapitre 4 : À propos du SDK Harmony Unity

Toon Boom fournit un progiciel Unity dans la boutique Unity Assets qui contient tous les scripts nécessaires pour importer les données exportées depuis Harmony. Une fois décompressée dans un nouveau projet Unity, la structure suivante est disponible dans le dossier SDK Toon Boom Harmony Gaming :

- Documentation (informations destinées aux développeurs sur l'intégration Unity)
- Plugins (bibliothèques SDK)
- Exemples (plans de démonstration, scripts d'exportation, source SDK et un générateur d'aperçu)
- Exécution (scripts d'Harmony qui s'exécutent pendant le jeu)
- Éditeur (scripts Harmony qui ne s'exécutent que dans l'éditeur et n'apparaissent pas dans les versions)
- Shaders (Fournit des shaders qui rendent correctement les fonctionnalités spécifiques à Harmony)
- Matériaux (matériaux préconstruits qui utilisent des shaders Harmony)

Ces progiciels peuvent être gérés manuellement via l'interface Package Manager dans Unity. Les développeurs peuvent mettre à niveau et rétrograder les progiciels dans le cas où les nouvelles versions d'Unity présentent certaines incompatibilités avec les scripts Harmony.

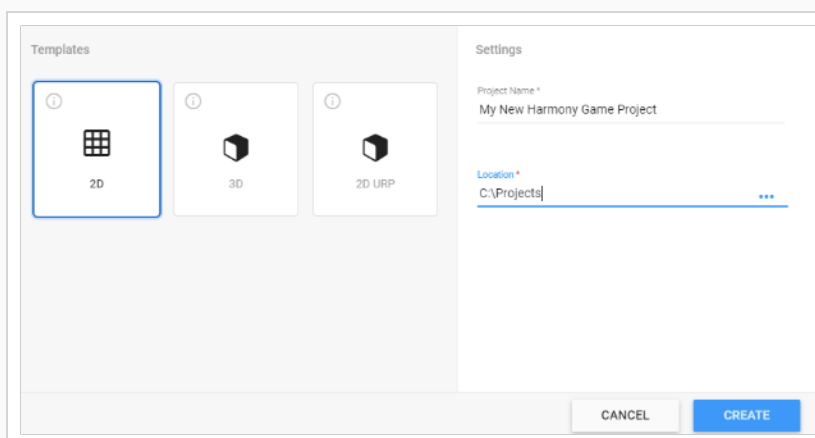
Comment créer un projet SDK Unity

1. Créez un projet de jeu Unity vide. Il est recommandé de créer un projet de jeu Unity 2D vide car il est particulièrement pratique lorsque vous travaillez sur le rendu et la physique 2D.



REMARQUE

Assurez-vous d'utiliser Unity 2020.3.36 ou une version ultérieure, les versions précédentes d'Unity ne disposent pas de toutes les fonctionnalités nécessaires pour prendre en charge l'importation et le rendu de fichiers TBG.

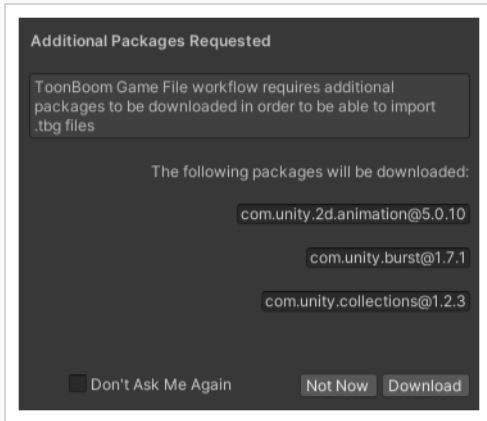


2. Installez le SDK Harmony Game disponible dans la boutique Unity Asset. Si votre SDK a été téléchargé manuellement, vous pouvez double-cliquer sur le fichier .unitypackage pendant que votre

projet Unity est ouvert.

La fenêtre contextuelle Progiciels supplémentaires demandés s'affichera pour garantir que vous disposez de progiciels Unity compatibles qui facilitent une animation 2D optimisée au moment de l'exécution.

3. Cliquez sur **Télécharger**.



Une fois qu'Unity a fini de télécharger les progiciels requis et de compiler tous les scripts, votre projet est prêt pour commencer à importer des personnages Harmony.

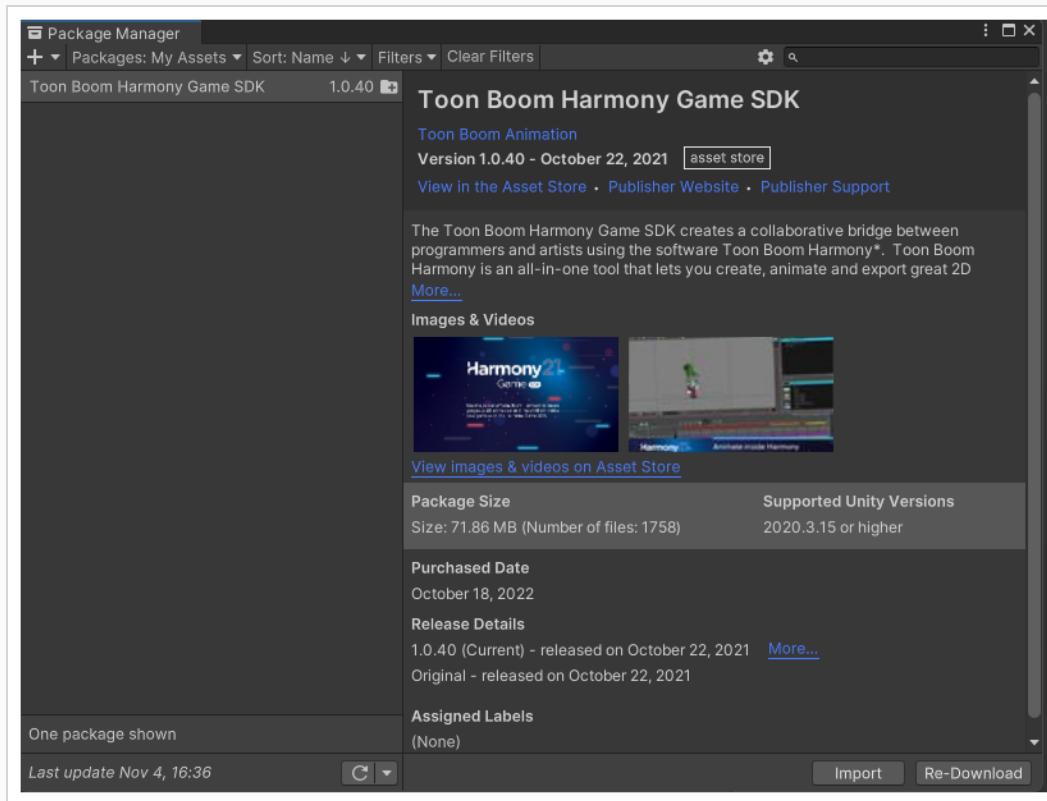


REMARQUES

Lorsque vous avez installé le SDK Toon Boom Harmony Game, le progiciel SDK Harmony devra être importé dans chaque nouveau projet créé dans Harmony.

Comment importer un progiciel dans un projet

1. Ouvrez la boutique de ressources en sélectionnant Fenêtre > Boutique de ressources. L'onglet Boutique de ressources s'ouvre.
2. Dans la boutique de ressources, sélectionnez **Ouvrir le Gestionnaire de progiciels**. La fenêtre Gestionnaire de progiciels s'ouvrira.



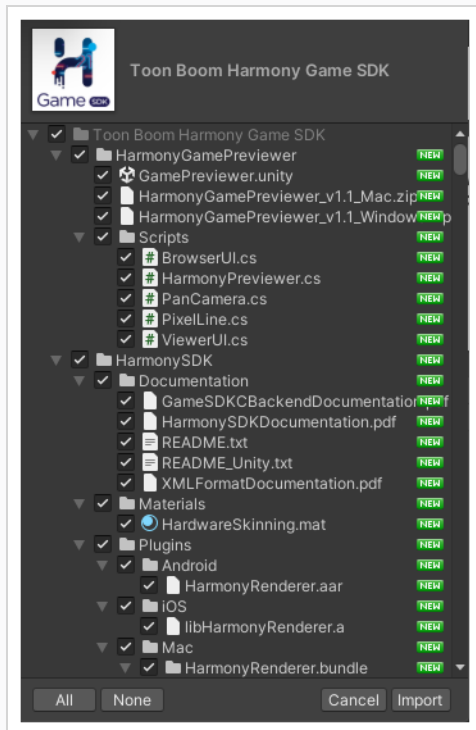
3. Dans la liste déroulante Progiciels en haut à gauche du Gestionnaire de progiciels, vous trouverez quatre options :

- Progiciels : Registre Unity
- Progiciels : Dans le projet
- Progiciels : Mes ressources
- Progiciels : Intégrés

4. Sélectionner **Progiciels : Mes ressources**.

Une liste de vos ressources téléchargées s'affichera dans la liste ci-dessous. À droite de l'onglet Gestionnaire de progiciels : Mes ressources, les informations sur la ressource sélectionnée s'afficheront.

5. Lorsque vous avez sélectionné le progiciel : Mes ressources Harmony, sélectionnez Importer au bas du Gestionnaire de progiciels. La fenêtre Importer un progiciel Unity affichant les fichiers de ressources s'affiche.



6. Sélectionnez **Importer**.

Harmony Previewer

Cette section explique comment utiliser Harmony Previewer pour afficher des animations sans installer Unity.

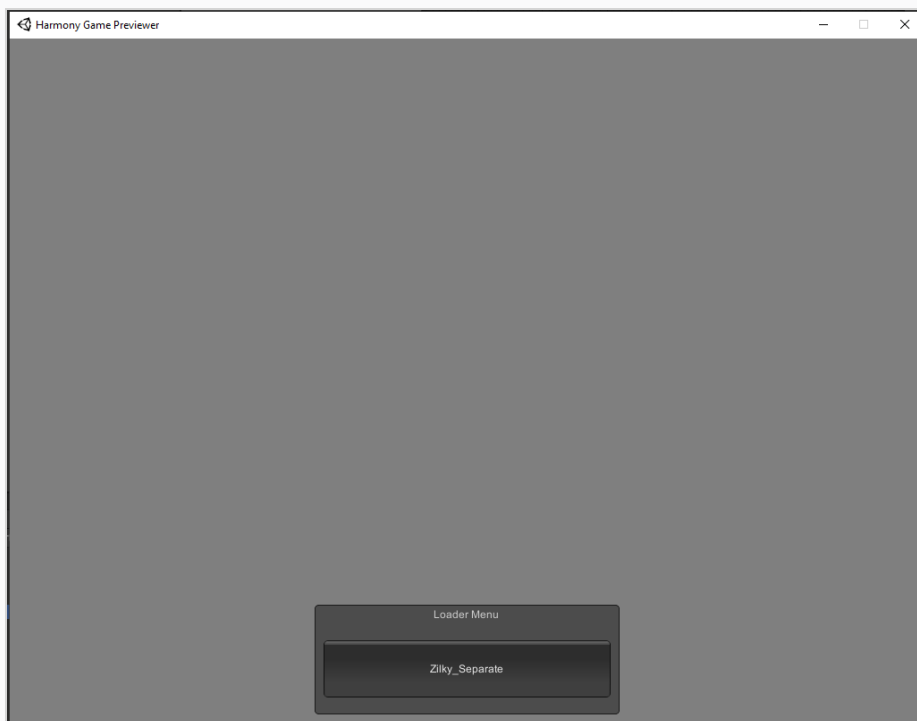
Vous pouvez trouver une version pour Windows ou macOS dans le dossier Harmony Game Previewer que vous pouvez simplement décompresser.

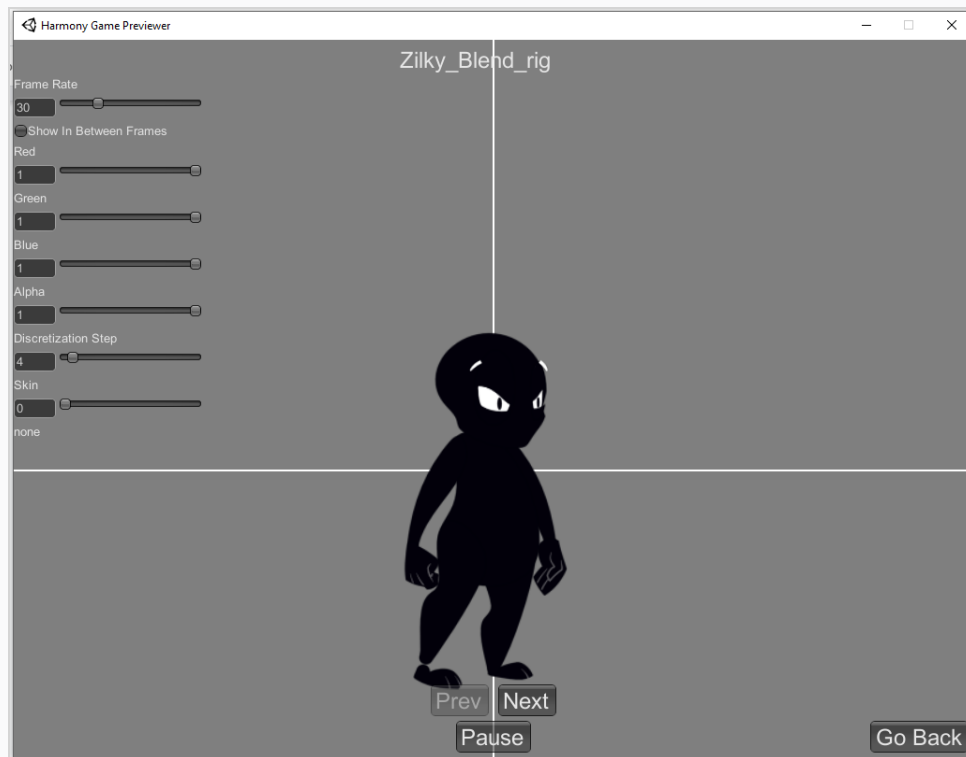


REMARQUE

Ne décompressez pas la version à l'intérieur de votre projet Unity, il est conseillé de la décompresser dans son propre dossier.

Comment utiliser le générateur d'aperçu





1. Exécutez *Harmony Game Previewer.exe* ou *Harmony Game Previewer.app*
2. Sélectionnez le projet à charger dans le menu de chargement. Ces projets sont chargés à partir du dossier Ressources d'Harmony dans le même répertoire que *Harmony Game Previewer.exe* ou *Harmony Game Previewer.app*
3. Naviguez dans les animations à l'aide du bouton Précédent / Suivant.
4. Ajustez les paramètres Fréquence d'affichage, Teinte, Discretization Step et Habillage à l'aide les commandes en haut à gauche.
5. Cliquez sur le bouton Revenir en arrière pour revenir au menu précédent et charger une autre animation.

Manipulation des ressources

Dans cette section, seront décrits les composants Harmony Mesh et Harmony Renderer dans Unity.

Workflow TBG

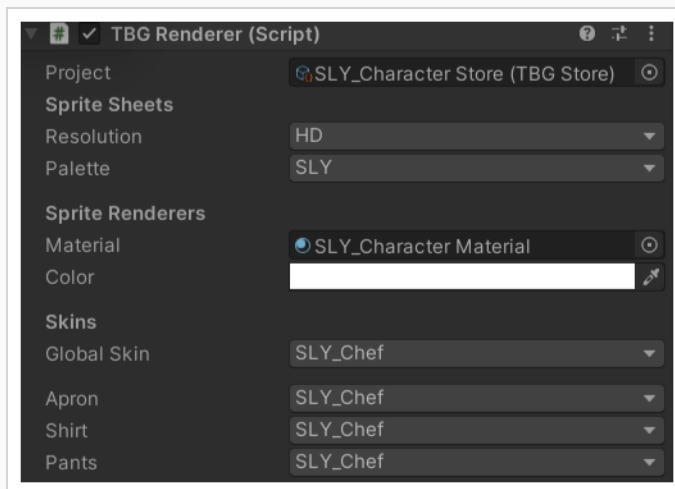
Si vous importez des sprites dans Unity via le workflow TBG, les options de manipulation des ressources s'afficheront pour les fichiers TBG dans Inspector View. Les options de manipulation des ressources affichées sont différentes des options de manipulation des ressources affichées dans le workflow XML.

Moteur de rendu des ressources TBG

Harmony Renderer pour le workflow des fichiers TBG permet de contrôler l'animation et le rendu des ressources Harmony pour les fichiers TBG ainsi que d'exposer les variables importantes que vous pourriez souhaiter modifier pendant l'exécution.

Comment manipuler les ressources de rendu du workflow TBG

1. Sélectionnez un personnage dans le plan. La section TBG Renderer (Script) apparaîtra dans l'onglet Inspecteur à droite.



2. Contrôlez les ressources de votre personnage en définissant les éléments suivants :
 - **Projet** : Une référence à la ressource TBGStore. Cette ressource générée par l'importateur contient les données du projet TBG qui ne sont pas couvertes par les AnimationClips générés.
 - **Résolution** : Résolution utilisée pour afficher ce personnage. Cela correspond aux noms de feuilles de sprite associés.
 - **Palette** : Palette utilisée pour afficher ce personnage. Cela correspond aux noms de feuille de sprite associés.
 - **Matériau** : Le matériau utilisé pour ombrer ce personnage. L'habillage matériel est le matériau non éclairé par défaut fourni avec le SDK.
 - **Couleur** : Option qui multiplie la couleur sélectionnée sur la ressource.

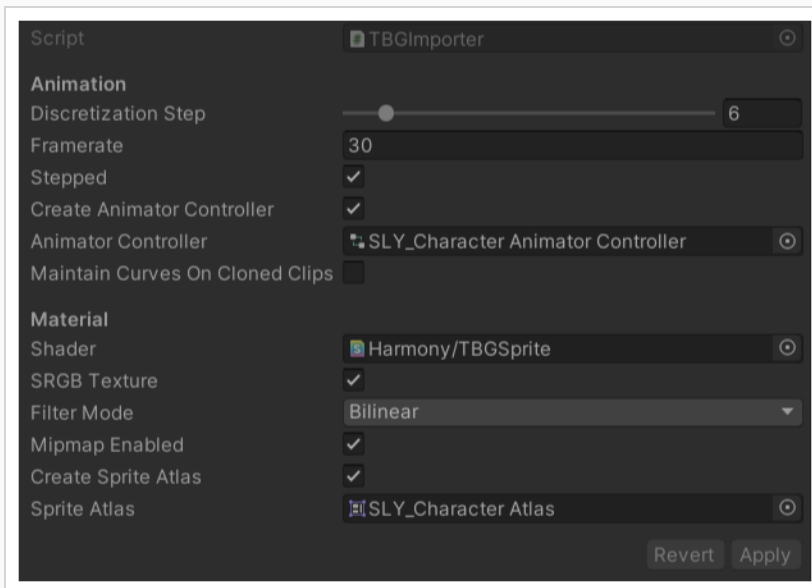
- **Habillage** : Vous permet de modifier en temps réel l'habillage que vous avez exporté depuis Harmony. Chaque groupe possède sa propre liste déroulante pour modifier l'habillage, ainsi qu'une liste déroulante globale pour modifier l'habillage de tous les groupes simultanément.

Importateur TBG

Lorsque des fichiers TBG sont importés dans Unity, les ressources Harmony sont converties en Prefab Unity, AnimationClips et autres ressources. L'importateur TBG vous permet de spécifier comment les ressources et les animations sont créés.

Comment contrôler les paramètres d'importation des ressources Harmony TBG dans Unity

1. Sélectionnez un fichier de personnage .tbg dans votre vue Projet. L'onglet inspecteur s'ouvrira.
2. En haut de Inspector View, cliquez sur **Sélectionner** pour le modèle. La vue Importateur TBG s'ouvrira et affichera les options d'importation du TBG.



3. Contrôlez la façon dont les animations seront importées en définissant les paramètres suivants :

- **Paramètres d'animation**

- **Discretization Steps** : Un curseur représentant le nombre de colonnes / lignes dans lesquelles les sprites seront divisés s'ils sont déformés par des bones
- **Frame Rate** : la fréquence d'affichage d'animation utilisée par le système d'animation d'Unity pour lire les images Harmony. Elle devrait correspondre à votre fréquence d'affichage dans Harmony.
- **Stepped** : Si cette case est cochée, les animations se déplaceront entre les images entières et ne seront donc pas interpolées de manière fluide. Si cette case n'est pas cochée,

l'animation sera lue entre des images entières pour obtenir une animation fluide.

- **Create Animator Controller** : Si cette case est cochée, cet importateur créera automatiquement un nouveau Animator Controller contenant tous les AnimationClips générés et placera la ressource à côté du Prefab dans la vue Projet.
- **Animator Controller** : Une fois les modifications apportées au fichier .tbg, l'Animator Controller référencé est mis à jour avec les nouveaux AnimationClips.
- **Maintain Curves on Cloned Clips** : Si cette case est cochée, les courbes dupliquées à partir de sous-ressources de clip dans le fichier .tbg (et référencées dans AnimatorController) verront leurs courbes remplacées par de nouvelles données du fichier .tbg mis à jour
- **Paramètres du matériau**
 - **Shader** : Une fois le Prefab créé, tous les SpriteRenderers référenceront un matériau généré à l'aide de ce shader.
 - **SRGB Texture** : Si cette case est cochée, l'espace colorimétrique gamma des textures générées sera mappé dans le rendu de l'espace colorimétrique linéaire.
 - **Filter Mode** : Comment les pixels voisins de la texture sont interpolés.
 - **Mipmap activé** : Générez des textures de résolution inférieure pour obtenir un rendu à des distances plus grandes.
 - **Créer un Sprite Atlas** : Si cette case est cochée, cet importateur créera automatiquement un nouvel Sprite Atlas contenant tous les Sprites générés et placera la ressource à côté du Prefab dans la vue Projet.
 - **Sprite Atlas** : Une fois les modifications apportées au fichier .tbg, le Sprite Atlas référencé est mis à jour avec les nouveaux Sprites.

Workflow XML

Si vous importez des sprites dans Unity via le workflow XML, les options de manipulation des ressources s'afficheront pour les fichiers XML dans Inspector View. Les options de manipulation des ressources affichées sont différentes des options de manipulation des ressources affichées dans le workflow TGB.

Harmony Mesh

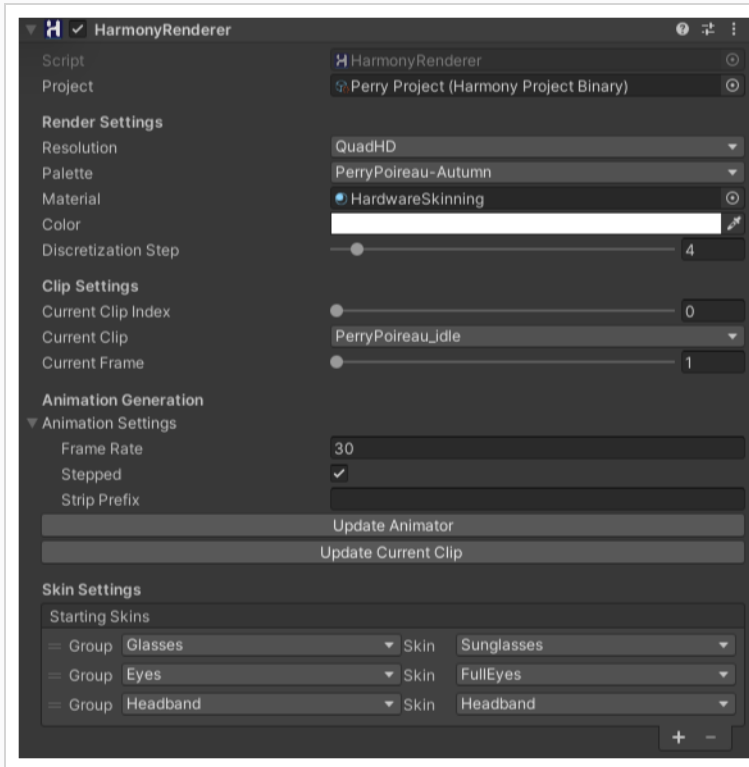
Le composant Harmony Mesh est utilisé par Harmony Renderer pour créer des maillages de ressources basés sur les dessins exposés dans votre animation.

Moteur de rendu des ressources XML

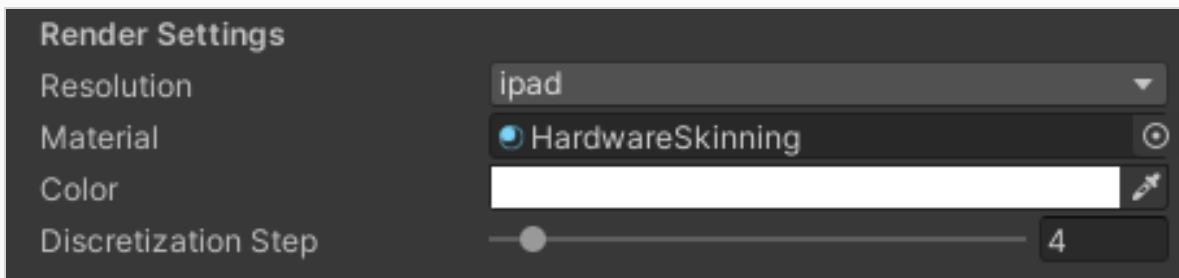
Harmony Renderer pour le workflow des fichiers XML permet de contrôler l'animation et le rendu des ressources Harmony pour les fichiers XML ainsi que d'exposer les variables importantes que vous pourriez souhaiter modifier pendant l'exécution.

Comment manipuler les ressources de rendu de workflow XML

1. Sélectionnez un personnage dans le plan
2. Dans Inspector View, sélectionnez Ajouter un composant au bas de la vue. La section HarmonyRenderer s'affichera.



3. Contrôlez les ressources de votre personnage en définissant les éléments suivants :
 - **Projet** : Une référence à la ressource binaire du projet Harmony. Cette ressource est générée par l'importateur et contient les données du projet Harmony dans un format binaire efficace.
 - **Paramètres de rendu** : Les paramètres de rendu permettent de modifier votre ressource, en modifiant la résolution, le Shader utilisé pour le rendu ou la qualité de la déformation.

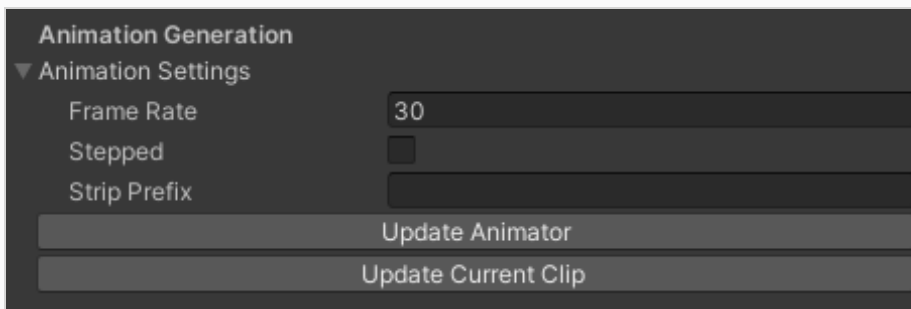


- **Résolution** : Résolution utilisée pour afficher ce personnage. Cela correspond aux noms de feuilles de sprite associés.

- **Palette** : Palette utilisée pour afficher ce personnage. Cela correspond aux noms de feuilles de sprite associés.
- **Matériau** : Le matériau utilisé pour ombrer ce personnage. L'habillage matériel est le matériau non éclairé par défaut fourni avec le SDK.
- **Couleur** : Option qui multiplie la couleur sélectionnée sur la ressource.
- **Discretization Steps** : Curseur qui définit la disposition de tous les dessins déformés.
- **Paramètres du clip** : Les paramètres des clips permettent de visualiser les animations sans ouvrir la fenêtre Animation. Cela affecte uniquement le projet en mode Éditeur, lorsque vous jouez le jeu, l'Animateur annule les paramètres dans les Paramètres du clip.



- **Index de clip courant** : Un index de clip numérique à afficher. Il peut être animé dans la fenêtre Animation.
- **Clip courant** : Une liste déroulante pour sélectionner l'animation à afficher. Il correspond à l'index de clip courant.
- **Image courante** : Numéro d'image actuellement affiché.
- **Paramètres d'animation** : Ces paramètres sont les mêmes que ceux de l'importation des paramètres Ressource. Ils sont utiles lors de l'importation ou de la réimportation de modifications de ressources dans Harmony. Une fois les modifications effectuées, cliquez sur Mettre à jour l'animateur ou Mettre à jour le clip courant afin d'enregistrer vos modifications.

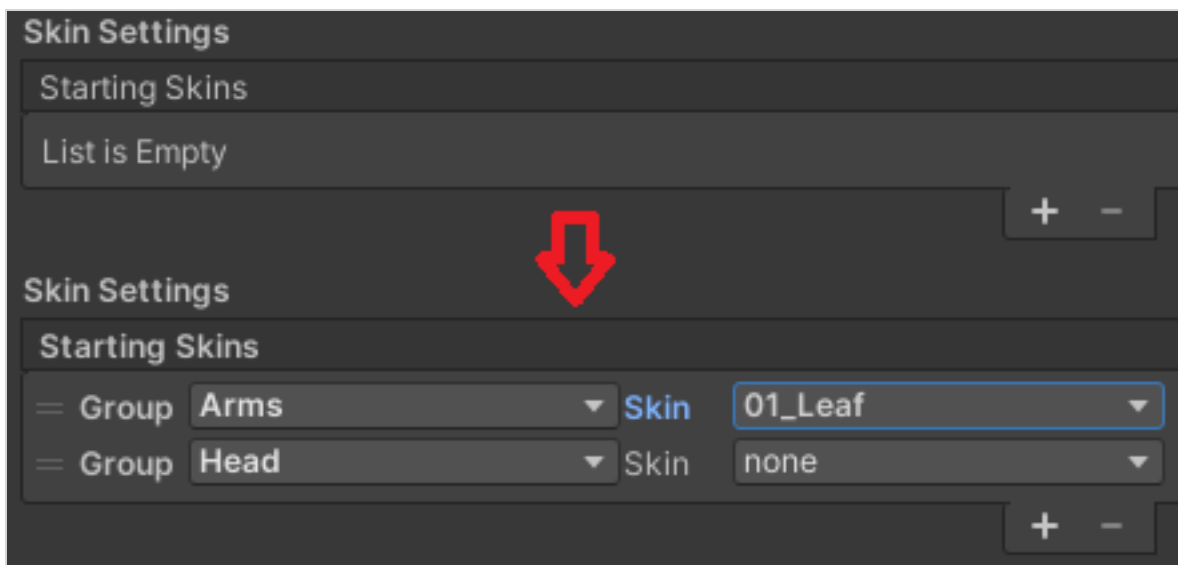


- **Fréquence d'affichage** : la fréquence d'affichage d'animation utilisée par le système d'animation d'Unity pour lire les images Harmony. Elle devrait correspondre à votre fréquence d'affichage dans Harmony.
- **Stepped** : Si cette case est cochée, les animations se déplaceront entre les images entières et ne seront donc pas interpolées de manière lisse. Si cette case n'est pas cochée, l'animation sera lue entre des images entières pour obtenir une animation lisse.

- **Supprimer le préfixe** : Ceci est utilisé pour supprimer un préfixe sur le nom de l'animation. Par exemple, si nous voulons que « Wishling_Idle » devienne « Inactif », nous définissons le préfixe à « Wishling_ ». Ceci est utile lorsque le nom de l'animation pour les jeux est le même pour tous les personnages.
- **Mettre à jour l'animateur** : Cliquez sur ce bouton pour mettre à jour l'ensemble des animations sur ce personnage pour qu'il corresponde à ce qui se trouve dans le projet Harmony. Cela mettra à jour les animations existantes avec les nouveaux paramètres que vous avez entrés et ajoutera ceux qui manquent.
- **Mettre à jour le clip courant** : Cela mettra à jour uniquement le clip actuellement sélectionné.

Paramètres d'habillage

Les paramètres d'habillage vous permettent de modifier en temps réel l'habillage que vous avez exporté depuis Harmony. Pour modifier un habillage, vous devez d'abord appuyer sur le bouton + pour ajouter un groupe. Vous devez ajouter autant de groupes que ceux que vous aviez dans votre exportation Harmony. Si vous n'avez pas de groupes, il suffit d'en ajouter un et de le laisser sur tous. Vous pouvez définir le groupe que vous souhaitez modifier, puis l'habillage à assigner à ce groupe. Vous pouvez ensuite supprimer un groupe en sélectionnant n'importe quel groupe et en appuyant sur le bouton -.



Comment modifier un habillage dans le workflow XML

1. Cliquez sur le bouton + pour ajouter un ou plusieurs groupes.
2. Modifiez le groupe et assignez l'habillage au groupe.
3. Pour supprimer un groupe, sélectionnez n'importe quel groupe puis cliquez sur le bouton -.

À propos de l'Exemple de projet Unity

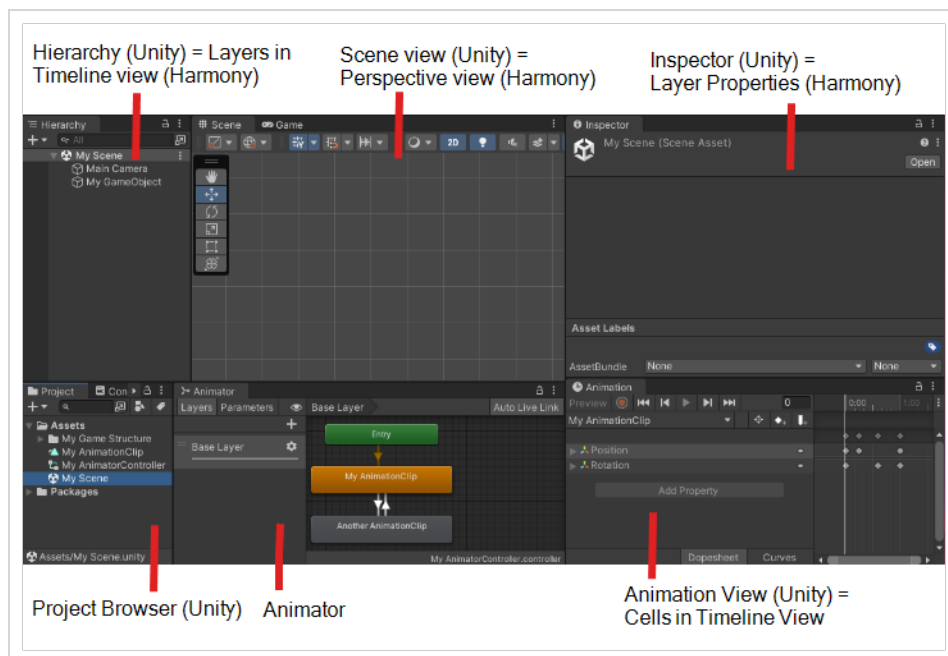
Toon Boom inclut un exemple de projet Unity qui contient tous les scripts nécessaires pour importer les données exportées à partir de Harmony. À l'intérieur de ce projet se trouve un dossier Ressources qui contient les dossiers suivants :

- Plugins
- Plans (plans de démonstration et un générateur d'aperçu)
- Scripts (tous les scripts Harmony)
- RessourcesenDiffusioncontinue (c'est l'endroit où tous les fichiers de plan Harmony doivent être exportés)

En configurant le script Harmony, vous pouvez exporter automatiquement des animations dans le dossier de ressources en diffusion continue. De cette façon, Unity chargera dynamiquement les ressources les plus à jour dès qu'elles apparaîtront dans le dossier. Vous pouvez également placer manuellement les données Harmony exportées dans cet emplacement.

À propos de l'interface Unity

Principaux composants de l'interface Unity et leurs équivalents dans Harmony :



Unity	Harmony	Description
Scene View	Perspective View	C'est ici que vous définissez le plan, en sélectionnant et en positionnant les environnements, le joueur, la caméra, les ennemis et tous les autres ObjetsJeu.
Game View	Camera View	La vue rendue à partir de la ou des caméras de votre jeu. Elle est représentative du jeu final et publié.
Inspector	Propriétés du calque	Affiche des informations détaillées sur l'ObjetJeu sélectionné, y compris tous les composants connectés et leurs propriétés.
Hiérarchie	Layers in Timeline view	Affiche la hiérarchie des éléments du plan et vous permet de configurer des relations parent-enfant pour différents objets de jeu.
Animation	Cells in Timeline View	Affiche les images clés d'une animation par transformation en cours d'animation.
Animateur		Affiche un graphique des états et des transitions qui vous permet de configurer la façon dont vos personnages réagiront aux événements de jeu et aux actions des joueurs

Navigateur de projets	---	Vous permet d'accéder et de gérer les ressources d'un projet.
-----------------------	-----	---

Voici quelques types de données avec lesquels Unity interagit et leurs équivalents dans Harmony :

Unity	Harmony	Description
Sprite	Dessin	Une image simple qui peut être affichée sur un Moteur de rendu de sprite.
SpriteRenderer	Calque	Affiche un sprite à la fois. Les sprites peuvent être échangés pendant une animation pour interpréter le mouvement de la partie du corps d'un personnage.
Transformer	Cheville	Utilisé pour former la hiérarchie du corps d'un personnage. Il peut être mis à l'échelle, pivoté ou traduit pour interpréter le mouvement de la partie du corps d'un personnage. L'inclinaison doit être dérivée d'une hiérarchie d'échelles et de rotations.
ObjetJeu		Toujours associé à une transformation. ObjetsJeu héberge des composants qui peuvent afficher et simuler le jeu. Le SDK fournit des comportements personnalisés qui s'exécutent sur ObjetsJeu pour afficher des visuels spécifiques à Harmony.
Ressource		Les données de base dans un projet Unity qui est utilisé pour les visuels, la physique, le jeu et les effets sonores. Il s'agit généralement de fichiers, mais ils peuvent être hébergés en tant que sous-ressources à l'intérieur de fichiers plus volumineux. Textures, AnimationClips, Sprites, AnimatorControllers, ClipsAudio et Prefabs sont tous des ressources.

Importation de fichiers Harmony dans Unity

Une fois que vous avez terminé la création de votre illustration et des mouvements animés des personnages dans Harmony, vous pouvez les importer dans Unity pour les intégrer au jeu.

Comment importer des fichiers Harmony directement dans Unity

1. Créez un projet Unity 2D.
2. Le SDK du projet Harmony doit être importé dans votre projet pour que les ressources puissent être intégrées. Effectuez l'une des opérations suivantes :
 - Accédez à la boutique Unity Asset (assetstore.unity3d.com/en/#!/content/31211) et enregistrez le fichier SDK Harmony Game sur votre ordinateur. Importez-le ensuite dans votre projet comme suit : **Menu supérieur > Ressource > Importer le progiciel > Progiciel personnalisé.**
 - Recherchez le SDK Harmony Game dans la boutique Unity Asset, puis téléchargez et importez le progiciel. Vous pouvez accéder à la boutique de ressources comme suit : **Menu supérieur > Fenêtre > Boutique de ressources.**



REMARQUE

Pour chaque nouveau projet Unity, vous devez répéter l'étape 2 car elle contient les fichiers et dossiers nécessaires au bon déroulement du processus d'importation Harmony.

Il existe deux workflows principaux pour l'exportation et l'importation de données de Harmony vers Unity

- **Fichiers TBG / TBGRenderer**

C'est le workflow à privilégier si :

- Vous utilisez exclusivement Unity et souhaitez bénéficier de l'expérience la plus native possible de la méthode de travail d'Unity.
- Vous souhaitez importer plus rapidement en moins d'étapes.
- Vous souhaitez produire une hiérarchie significative de transformations et de SpriteRenderers qui peuvent être modifiés et avec laquelle vous pouvez interagir. Cela permettra à ceux qui ont une expérience de script occasionnelle ou sans expérience de script d'ajouter des effets procéduraux uniques sur les personnages.
- Vous souhaitez tirer parti du maillage de sprite d'Unity combinant l'atlassage de sprites et les contrôleurs d'animateur avec des calques et la fusion de transition. Cela ouvre la voie à l'utilisation de progiciels existants ou de scripts simples pour la cinématique inverse, la physique des secousses, les poupées de chiffon, etc.

- **Dossiers XML / HarmonyRenderer**

C'est le workflow à privilégier si :

- Vous utilisez Unity pour faire des prototypes et souhaitez une mobilité vers d'autres moteurs de jeu. Le workflow XML est transférable lorsqu'il est appelé dans une bibliothèque C++ portable sur d'autres moteurs.
- Vous souhaitez disposer de plus d'espace pour expérimenter des solutions de diffusion en continu de ressources personnalisées et voulez un point de départ pour créer des effets procéduraux personnalisés de haute performance.

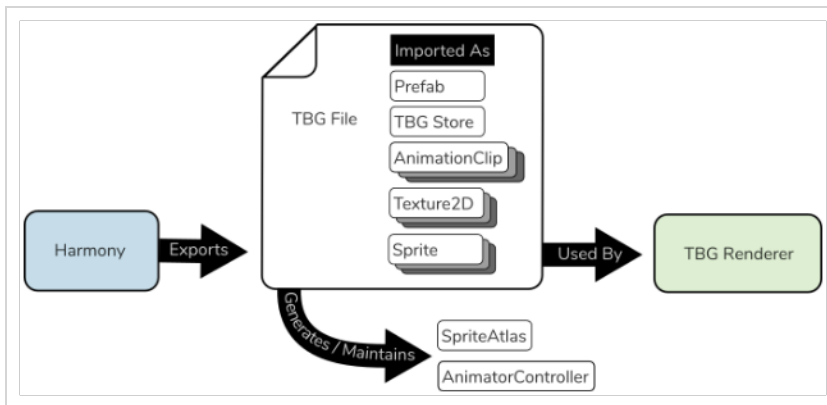


REMARQUES

- Vous pouvez alterner les workflows le cas échéant si vous le souhaitez.
- Les deux systèmes utilisent par défaut des contrôleurs d'animation pour choisir et lire des animations sur le personnage, de sorte que les scripts interagissant uniquement avec l'animateur du personnage fonctionneront de la même manière. Toutefois, les scripts qui modifient les habillages, les résolutions ou fournissent des effets procéduraux devront être réécrits ou créés et personnalisés pour chaque workflow.
- En règle générale, la transition de HarmonyRenderers à TBGRenderers fonctionnera mieux, car HarmonyRenderers dispose d'un sous-ensemble de fonctionnalités disponibles dans TBGRenderers par défaut. Nous vous conseillons de maintenir l'accès à vos projets Harmony .xstage d'origine si vous avez besoin de les réexporter avec des paramètres différents ou un workflow différent.

Workflow de fichier TBG dans Unity

Dans le workflow du fichier TBG (Toon Boom Gaming), Unity interprétera immédiatement un fichier TBG exporté depuis Harmony comme une nouvelle ressource qu'il peut référencer pour dériver de nouvelles Textures, Sprites, données de projet et un Prefab du fichier. Le Prefab obtenu peut être glissé dans le plan d'Unity pour être immédiatement visualisé et animé. Lorsque le fichier TBG est écrasé à partir d'une nouvelle exportation dans Harmony, l'Importateur TBG peut automatiquement effectuer le processus de réimportation pour mettre à jour le Prefab, les Sprites et les Textures existants. Cela permet aux artistes d'itérer rapidement sur leurs personnages et de voir leurs résultats immédiatement dans le jeu.



Le prefab TBG Renderer résultant contient une hiérarchie complète de Transformations et SpriteRenderers pour afficher le personnage dans un rig convivial dans Unity. En outre, un script de rendu TBG Renderer inclus garantit que les Sprites sont échangés pendant l'avancement d'une animation et que les coupures s'appliquent correctement aux SpriteRenderers pertinents.



REMARQUE

Le format XML utilisé dans les fichiers TBG est une version plus récente et peut ne pas être compatible avec le workflow du dossier XML pour les caractères plus compliqués. Nous vous conseillons de conserver vos fichiers Harmony originaux et de les réexporter ultérieurement avec des paramètres différents.

Comment importer un fichier TBG dans Unity

1. Dans Harmony, recherchez le bouton Exporter vers la feuille de sprite 🎮 dans la barre d'outils Jeux.
2. Dans la fenêtre Exporter vers une feuille de sprite, cochez la case **Encoder en TBG**.
3. Pour le chemin d'accès à la ressource, saisissez le chemin du fichier où vous souhaitez enregistrer votre personnage.
4. Sélectionnez **Enregistrer et exporter**. Une barre de chargement s'affichera qui terminera le processus d'exportation et générera un nouveau fichier TBG.
5. Ouvrez votre projet Unity. Unity devrait automatiquement charger le TBG en tant que Prefab.

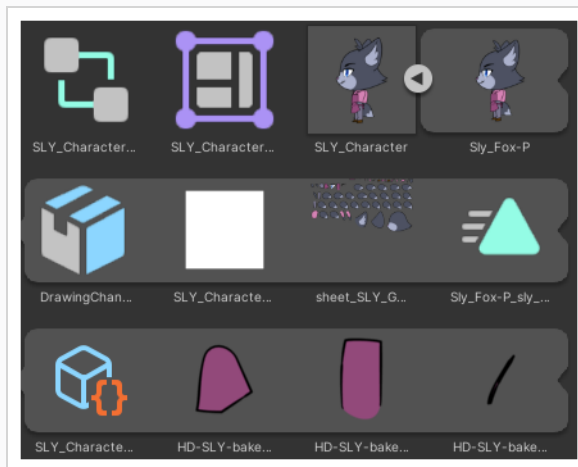
**REMARQUE**

Si votre personnage n'apparaît pas en tant que Prefab, veuillez vous assurer que :

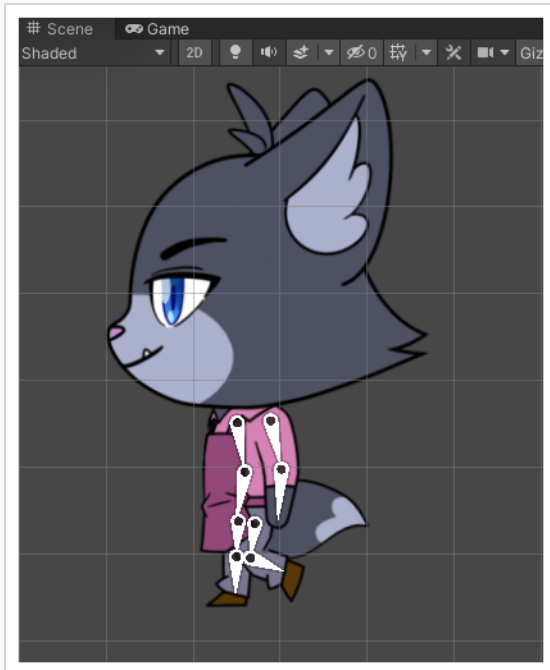
- Vous avez installé le SDKHarmony pour votre projet Unity. Celui-ci doit être installé pour chaque nouveau projet créé.
- Vous n'avez pas de marqueur de plan s'étendant au-delà de l'exposition d'un personnage.

Votre projet Unity doit comprendre :

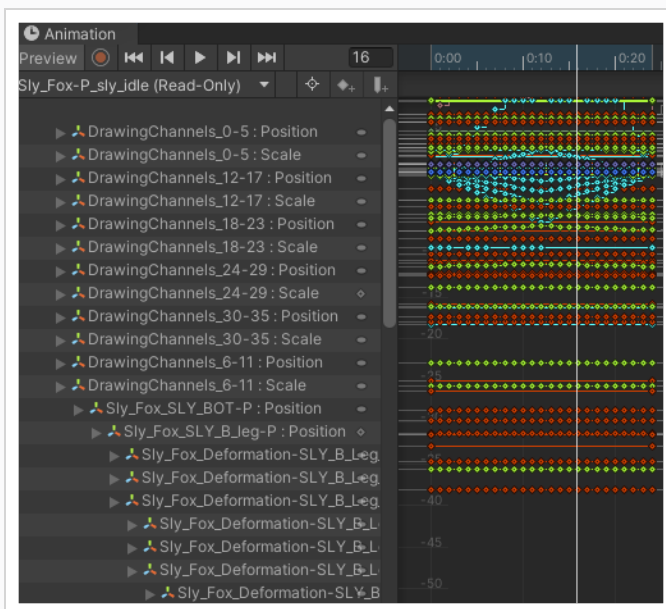
- Des sous-ressources pour les textures, des sprites, des données du TBG Store et les ExtraitsvidéoAnimation.
- Deux nouvelles ressources pour SpriteAtlas et AnimatorController du fichier à côté du TBG Prefab.



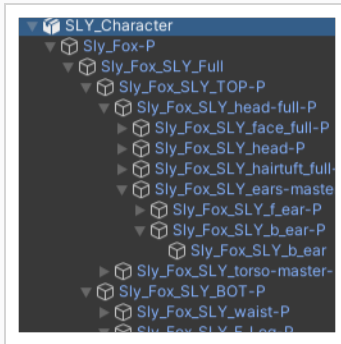
Le Prefab TBG peut maintenant être glissé dans la Hiérarchie, où il sera affiché et interactif.



En sélectionnant le Prefab dans la hiérarchie, vous pourrez afficher les animations du personnage à partir de la vue Animation.

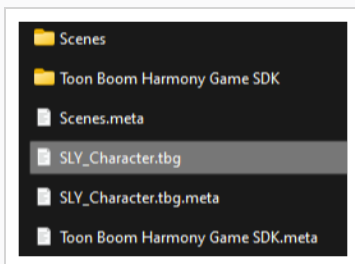


Ouvrez le Prefab dans la vue Hiérarchie en cliquant sur la petite flèche, pour explorer la hiérarchie du personnage de manière très similaire à ce qui est affiché dans un montage Harmony.

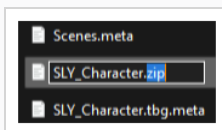


Comment inspecter manuellement le contenu d'un fichier TBG

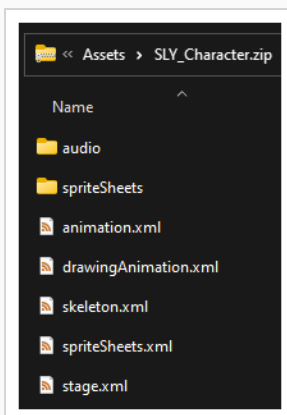
1. Recherchez le fichier TBG dans l'explorateur de fichiers de votre ordinateur



2. Si vous utilisez Windows ou Mac, renommez l'extension de fichier car l'extension de fichier .tbgi n'est pas reconnue par les ordinateurs Windows et Mac.



Une fois renommés, Windows et Mac vous permettront d'explorer le contenu du fichier .zip, afin de vous assurer que vos FeuillesdeSprite sont de taille raisonnable ou que vos fichiers de données correspondent au contenu que vous attendez.



**REMARQUE**

Pour vous assurer qu'Unity ne rencontre aucun problème, vous pouvez modifier l'extension de fichier en .tbg avant de reprendre le travail avec Unity. En règle générale, ce processus d'inspection ne doit être effectué que lorsque vous utilisez un ordinateur sur lequel Unity n'est pas installé.

Ajout de cinématique inverse à une Prefab TBG Renderer

Vous pouvez utiliser la fonction de cinématique inverse 2D d'Unity pour ajouter des cibles procédurales aux membres de vos personnages. Vous pouvez utiliser SLY_Character de l'exemple de workflow TBG pour ce processus.

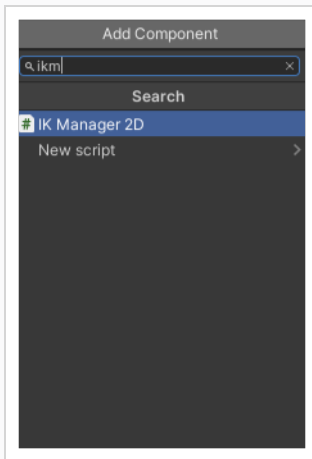
Comment ajouter la cinématique inverse à un Prefab TBG Renderer

1. Recherchez et faites glisser SLY_Character de la vue Projet vers votre Scene View, puis appuyez sur F pour cadrer le personnage dans votre vue.

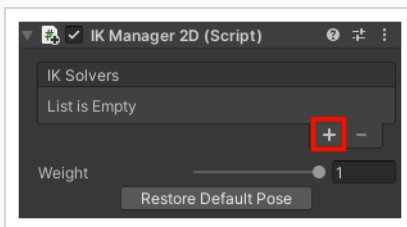
**REMARQUE**

Assurez-vous que le SLY_Character est sélectionné dans la hiérarchie.

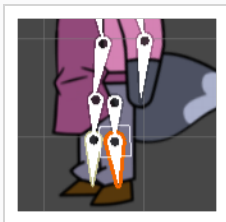
2. Dans l'Inspector View, ajoutez un composant IK Manager 2D.



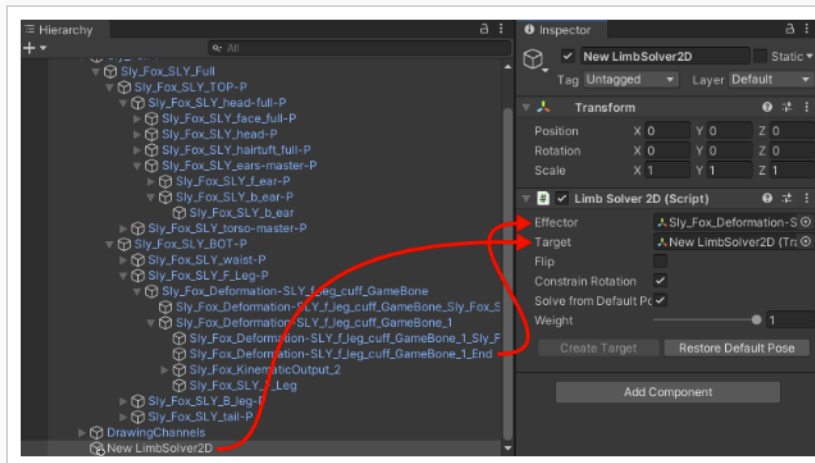
3. Dans le composant IK Manager 2D, cliquez sur le bouton + pour ajouter un nouveau solveur IK, puis sélectionnez **Membre**.



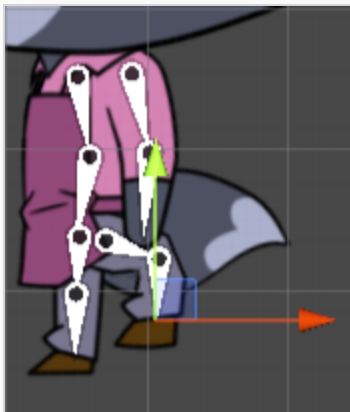
4. Cliquez sur le bone en bas à droite des jambes de Sly pour exposer sa position dans la hiérarchie. Nous utiliserons son frère ObjetdeJeu qui comprend « ..._GameBone_1_end » dans le nom.



5. Une fois l'objet de jeu « New LimbSolver2D » sélectionné, faites glisser « New LimbSolver2D » dans l'emplacement cible du script LimbSolver2D et faites glisser le « _leg_cuff_GameBone_1_end » GameObject dans l'emplacement Effecteur.



Une fois cela fait, vous pouvez vous déplacer la nouvelle transformation LimbSolver2D et voir le membre se plier pour se conformer à la cible.



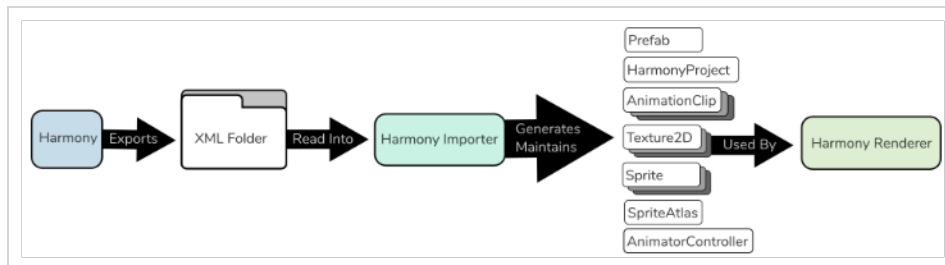
Pour changer la direction de la flexion du genou, nous pouvons cliquer sur la case à cocher « Retourner » dans le composant Limb Solver 2D.

Workflow de dossier XML dans Unity

Dans le workflow des dossiers XML, une ressource Harmony Importer unique peut être créée par un développeur pour référencer un dossier XML exporté depuis Harmony. Cet importateur peut ensuite être utilisé pour générer une ressource Harmony Project et un nouveau Prefab du moteur de rendu Harmony Render.

Pour chaque fichier XML réexporté, les développeurs doivent définir manuellement les chemins d'importation et d'exportation et effectuer le processus d'importation à partir de la ressource Harmony Importer.

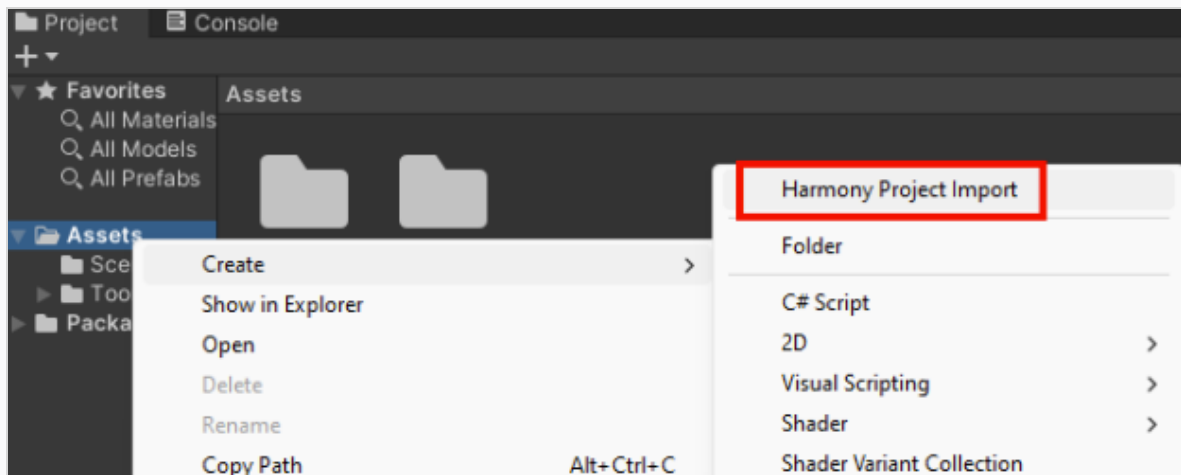
Ce workflow offre une certaine souplesse quant à l'emplacement des données XML. Par exemple, il peut résider à l'intérieur ou à l'extérieur du projet Unity. Toutefois, l'inconvénient de ce workflow est que davantage d'erreurs peuvent être commises, car les étapes sont plus nombreuses. Par exemple, le chemin d'accès aux fichiers de l'importateur TBG devra être mis à jour manuellement lors du déplacement des fichiers.



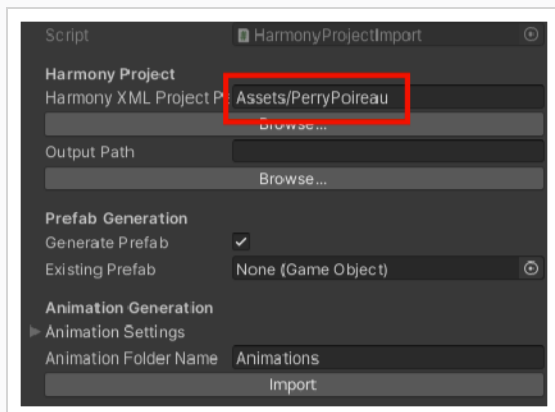
Le Prefab Harmony Render qui en résulte permet un rendu performant des personnages en générant un maillage au sein d'un plugin C++. Ceci est préférable pour les foules de personnages qui ne nécessitent pas de fonctionnalités de rendu personnalisées compliquées.

Comment importer un dossier XML dans Unity

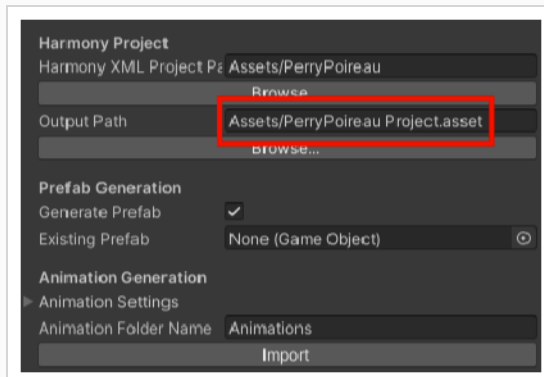
1. Dans Harmony, recherchez le bouton Exporter vers la feuille de sprite 🎮 dans la barre d'outils Jeux.
2. Sélectionnez **Enregistrer et exporter**. Assurez-vous que l'option Exporter en tant que TBG n'est pas cochée dans la boîte de dialogue Exporter vers une feuille de sprite.
3. Enregistrez le code XML dans un dossier à l'intérieur du dossier Ressources de votre projet Unity.
4. Dans Unity, créez une nouvelle ressource Harmony Importer en effectuant l'une des fonctions suivantes :
 - Dans la liste déroulante Ressources dans le menu supérieur, sélectionnez **Créer > Importer un projet Harmony**.
 - Cliquez avec le bouton droit sur le dossier Ressources dans la vue Projet et sélectionnez **Créer > Importer un projet Harmony**.



5. Nommez l'Harmony Importer avec un suffixe « Importer » pour éviter toute confusion future.
6. Copiez le chemin d'accès relatif du dossier exporté, collez-le dans le nouvel Harmony Importer.

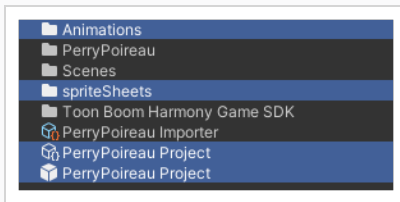


7. Copiez le chemin d'accès relatif de la ressource Harmony Importer, collez-le dans Harmony Importer, remplacez « Importer » par « Project » dans le chemin d'accès.

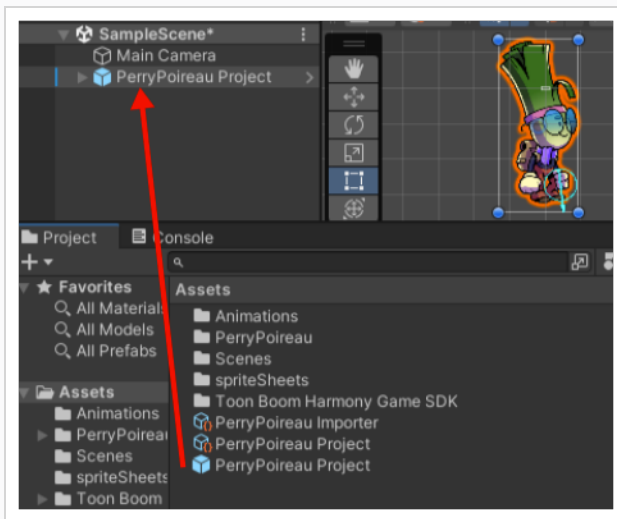


8. Cliquez sur **Import**.

Une nouvelle ressource HarmonyProject sera créée ainsi qu'un nouveau Prefab avec Harmony Renderer qui fait référence à HarmonyProject.



La ressource Prefab peut être glissée à partir de la vue Projet vers la Hierarch View et la Scene View pour afficher votre personnage dans votre jeu

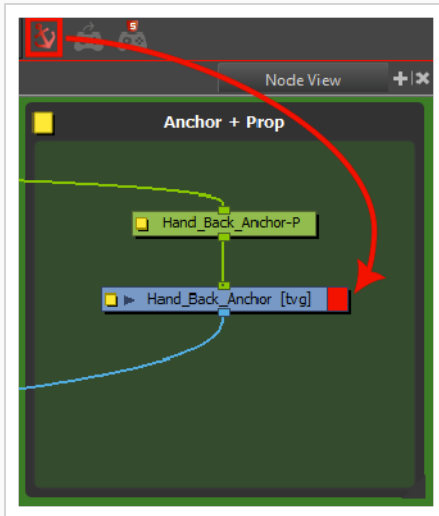


Importation d'ancres

Dans Harmony, les ancres fournissent un moyen d'indiquer quelles chevilles et dessins sont des positions d'intérêt pour les scripts de jeu dans Unity. Un ancrage peut être assigné à tout dessin ou cheville.

Comment importer une ancre dans Unity

1. Sélectionnez un calque dans la Timeline View ou un nœud dans la Node View et sélectionnez le bouton Basculer l'ancre dans la barre d'outils Jeux.



2. Importez le personnage dans Unity.

La prochaine fois que le processus d'importation XML est effectué, la nouvelle ancre apparaîtra dans Unity sous le Prefab importé du personnage en tant que nouvelle HarmonyAnchor.



Ce nouveau GameObject suivra la position, la rotation et l'échelle de l'animation courante sur le personnage. Cela vous permet d'attacher dynamiquement des objets de jeu en tant qu'enfants aux ancrs pour les transporter, les équiper et les utiliser.



REMARQUE

Le déplacement des positions d'ancrage n'affectera pas les visuels du moteur de rendu Harmony Renderer. Pour afficher et affecter l'ensemble de la structure du personnage dans Unity et pas seulement les ancrs, vous pouvez envisager d'utiliser le workflow TBG

Ajout de sources audio

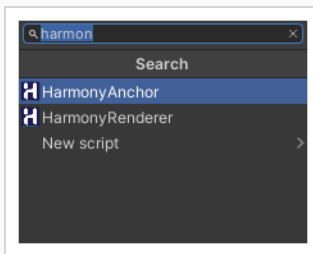
Lorsque vous créez des fichiers de plan dans Harmony avec audio, l'audio est exporté dans le dossier du projet. Harmony Importer créera automatiquement une AudioSource et l'ajoutera au Prefab généré. Harmony Renderer peut ensuite déclencher la lecture des fichiers AudioClip à certains moments de la lecture de l'animation, comme cela a été fait dans Harmony.

Ajout d'un moteur de rendu Harmony Renderer à des objets de jeu vides

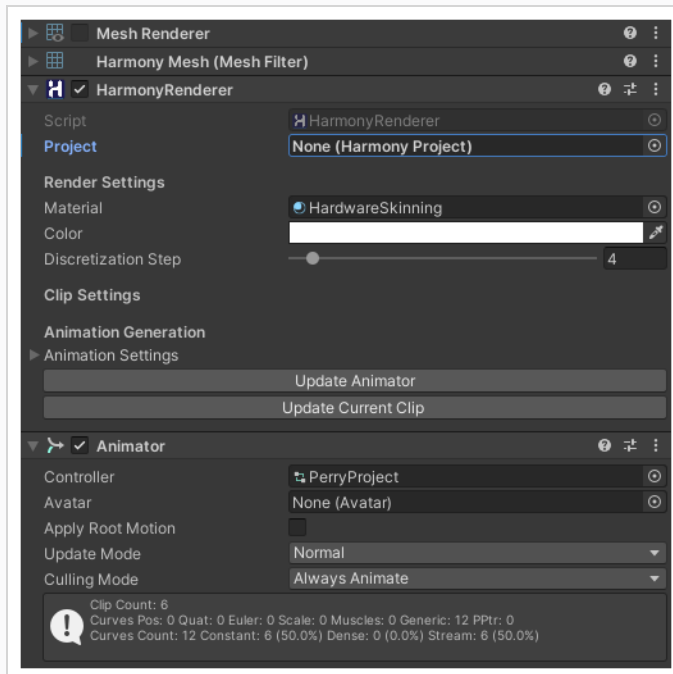
Si vous ne disposez pas du Prefab de votre Harmony Importer, il est toujours possible de créer un nouveau composant Harmony Renderer sur un objet de jeu vide pour obtenir le même résultat final.

Comment utiliser un GameObject vide

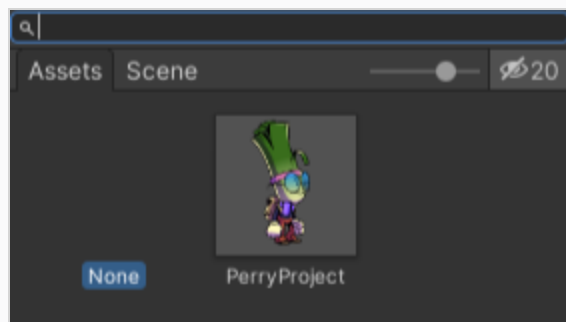
1. Sélectionnez **GameObject > Créer vide**.
2. Renommez le GameObject vide pour qu'il soit clair dans la hiérarchie. Puisque nous utilisons le fichier de démonstration PerryPoireau, renommez le GameObject en **SpaceCat** en effectuant l'une des opérations suivantes :
 - Double-cliquez sur le nom et renommez-le dans la hiérarchie.
 - Sélectionnez GameObject et renommez-le dans Inspector.
3. À ce stade, c'est un objet vide. L'accès Harmony aux données se fait via des scripts.
 - Sélectionnez le GameObject PerryPoireau.
 - Dans Inspector, allez sur **Ajouter un composant > Scripts > Harmony Renderer**.



Quatre nouvelles sections apparaissent dans Inspector appelées Mesh Renderer, Harmony Mesh (Filtre de maillage), et Harmony Renderer.



De là, vous devez joindre un Projet Harmony à ce Moteur de rendu Harmony Renderer nouvellement créé. En cliquant sur le cercle à droite de l'emplacement vide (Projet Harmony), vous pouvez sélectionner le projet que vous avez déjà importé en procédant à la création d'un Projet Harmony à l'aide du workflow de dossier XML.



Configuration des collisions dans Unity

L'extraction des informations de la zone englobante est utile lorsque vous voulez faire entrer quelque chose en collision avec votre personnage. Lorsque vous placez un personnage 2D dans un plan 3D, vous pouvez utiliser la fonction Physique pour que le caractère 2D entre en collision avec le plan 3D.

Les collisionneurs ont été améliorés pour fournir plusieurs types différents de collisionneurs à utiliser dans Unity. Vous pouvez sélectionner des collisionneurs de polygones et des collisionneurs de boîtes pour un cadre de sélection plus précis.

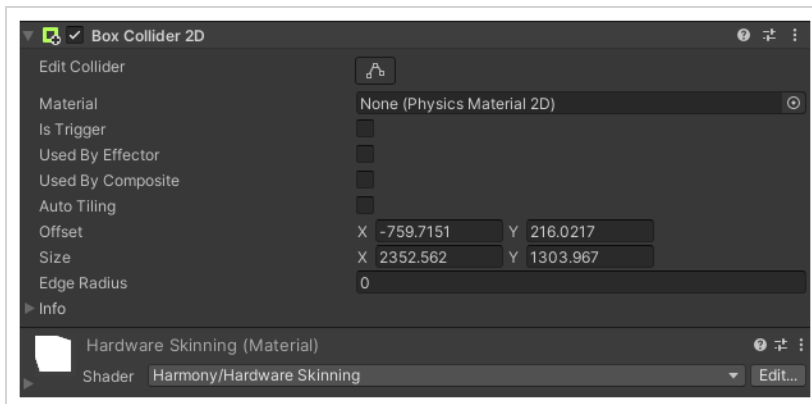
Comment configurer les collisions

1. Sélectionnez votre personnage importé.
2. Dans Inspector, sélectionnez **Ajouter un composant > Physique 2D > Collisionneur de boîte 2D**.
3. Pour activer la fonction Physique sur le personnage, allez dans **Ajouter un composant > Physique 2D > Corps rigide 2D**.



REMARQUE

Vous pouvez également utiliser la fonction physique 3D sur les fichiers importés.



Vous pouvez maintenant ajuster les variables et voir les résultats.

Glossaire des jeux

2D Animation Package	Plugin de première partie pour Unity (réalisé par Unity) qui facilite le traitement et l'animation de sprite 2D avancés (déformations, rigging, tessellation)
Ancre	Dans Harmony, les nœuds peuvent être marqués comme des ancres pour exposer leur position, leur rotation et leur échelle dans le moteur de jeu pendant l'exécution. Les fichiers TBG ne nécessitent pas d'ancres.
AnimationClip	La ressource Unity Asset qui contient une série de courbes pour différents champs de données dans un Prefab (changements de position, de rotation, d'échelle)
AnimatorController	Les contrôleurs d'animateur décident quelle animation doit être lue pendant le jeu en fonction de l'état actuel du personnage
Automatic Importing Process	Importera le fichier automatiquement sans intervention de l'utilisateur lorsque le fichier est créé ou est modifié / écrasé / mis à jour
bake_group	Les bake_groups sont des groupes dans la Node View qui sont renommés en « bake_... », en outre, un bake_group ne peut prendre en charge qu'1 port d'entrée et qu'1 port de sortie. Pendant l'exportation à partir d'Harmony, ces groupes sont analysés pour détecter les changements et exporter autant de sprites qu'il y a de variations uniques de données dans le groupe, ainsi qu'une animation pour montrer chaque sprite le cas échéant. N'importe quel effet peut être utilisé à l'intérieur d'un bake_group : lueurs, déformations de forme libre, dessin couleur / dessin au trait.
Bones	Segments rigides du corps d'un personnage, utilisés comme guide pour déformer un personnage - généralement les jambes et les bras
Create Animator Controller	Option permettant de créer un Animator Controller lorsque le fichier TBG est importé pour la première fois
Discretization Step	Finesse avec laquelle un sprite sera segmenté s'il est déformé. Plus le nombre est élevé, plus les formes des sprites intensément déformés seront lisses
FBX	Format de fichier FBX - format de fichier standard dans l'industrie de la modélisation 3D et du développement de jeux 3D, popularisé par 3ds Max
Feuille de sprite	Plusieurs sprites tous aplatis en une seule image. Certaines feuilles se

	trouvent dans une grille d'une taille fixe, d'autres ont des sprites de taille variable. Le moteur de jeu doit comprendre où se trouve chaque sprite dans la feuille pour pouvoir extraire correctement les données de sprite de la feuille.
fichier .tbg	Format de fichier Toon Boom Game (.tbg), créé à partir d'Harmony et repris par Unity et à l'avenir par d'autres moteurs de jeu.
Filter Mode	Comment mélanger les pixels voisins - Le plus proche affiche les bords durs autour des pixels (pixel-art), tandis que bilinéaire et le trilineaire tentent de mélanger les données de couleur des pixels voisins ensemble pour rendre l'image plus lisse
Fréquence d'affichage	Nombre d'images par seconde qui s'affiche lors de la lecture d'une animation
Game Bones	Déformation dans Harmony qui émule le fonctionnement des bones dans les moteurs de jeu, similaire aux déformations de Bones normales dans Harmony, mais avec un algorithme de fusion différent
Habillages	D'autres collections de dessins peuvent être créées dans le montage d'un calque puis basculées à l'aide de la vue Habillage dans Harmony. Les exemples incluent - le costume alternatif d'un personnage qui peut être changé, ou de nouvelles parties du corps d'un personnage, qui sont éventuellement montrées ou cachées pendant le jeu
IK Manager 2D	Plugin de première partie pour Unity (réalisé par Unity) qui facilite la cinématique inverse pour les personnages 2D
Importer Settings	Paramètres disponibles sur la ressource à partir de la fenêtre Inspecteur qui modifient la façon dont la ressource doit être convertie à partir des données du fichier en données comprises par Unity
Inspector View	Vue dans Unity qui affiche le contenu d'un ObjetJeu - la transformation, les composants pour le rendu et la fonction physique, et les Comportements personnalisés
MipMap	Coché dans les paramètres d'importation dans Unity, MipMap indique que l'utilisateur souhaiterait avoir des versions de résolution inférieure des textures générées pour le rendu à des échelles plus petites / des distances plus éloignées
Prefab	Hiérarchie d'ObjetsJeu, Transformations, Composants et comportements qui peuvent être créés dans la vue Hiérarchie, puis enregistrés dans la

	<p>vue Projet. Les instances de Prefabs peuvent être lacées dans la hiérarchie pour avoir plusieurs doublons du même objet (nombreux personnages, nombreuses puces, nombreuses arborescences)</p>
Preview Window	<p>Section de l'onglet Inspecteur qui montre à quoi ressemble la ressource, généralement avec des Modèles, des Matériaux et des Textures</p>
Scene View	<p>Affichage visuel du jeu dans Unity. Complète la vue Hiérarchie, où Scene View montre à quoi elle ressemblera pour un joueur, et la Hiérarchie montre à quoi ressemble le jeu pour le moteur.</p>
Shader	<p>Les instructions qui informent le jeu sur la façon d'afficher les données de maillage et de texture fournies (coloration, éclairage, lissage, courbure, mélange, etc.)</p>
Shadergraph	<p>Format de fichier Unity fourni par Unity avec une interface utilisateur et un importateur de scripts, qui permet la création et l'édition visuelles de shaders - aucune création de code n'est requise</p>
Sprite Atlas	<p>La version d'Unity d'une feuille de sprite - prend une collection de sprites et les aplatit en une seule image. Chaque sprite de l'Atlas pointe vers les données de l'Atlas pendant le rendu, ce qui réduit les performances de nombreux sprites différents.</p>
SRGB Texture	<p>Coché dans les paramètres d'importation dans Unity, sRGB indique à Unity que les données de texture ont été encodées dans l'espace colorimétrique linéaire, ce qui correspond au mode de rendu linéaire d'Unity</p>
Stepped	<p>Si l'animation doit être maintenue à la fréquence d'affichage prévue par l'artiste, ou si elle doit deviner les images intermédiaires lorsqu'elle est affichée à des fréquences d'affichage plus élevées</p>
Sub-Asset	<p>Sous-ressource qui est intégrée dans une ressource existante et qui est généralement associé à la ressource principale. Pour les fichiers TBG, un personnage se compose d'une ressource Prefab principale et possède de nombreux sous-ressources pour les textures, les sprites, les AnimationClips, les matériaux et TBGStore</p>
TBG	<p>Format de fichier Toon Boom Game (.tbg), créé à partir d'Harmony et repris par Unity et à l'avenir par d'autres moteurs de jeu.</p>
TBG Renderer	<p>Comportement d'exécution attaché à un Prefab pendant le processus d'importation TBG qui gère le changement des résolutions, des palettes,</p>

	des habillages et la mise à jour des sprites de couteau.
TBG Store	Ressource générée pendant l'importation par TBG qui stocke des données utiles sur le personnage qui sont référencées pendant l'exécution par TBG Renderer
Unity Asset	Dans Unity, une partie des données générée par Unity est native d'Unity et elle est intégrée dans un jeu pendant le processus de création de jeu. Celles-ci sont générées par les outils d'Unity et lors de l'importation de fichiers externes.
Unity Custom Importer	Importateur dans Unity qui est défini dans l'interface de script C# par projet, qui mappe une extension de fichier (par ex. « .tbg ») à un processus de conversion des données de fichier en ressources Unity Asset
Workflow TBG	Processus d'exportation d'un personnage depuis Harmony et d'importation dans un moteur de jeu en utilisant des fichiers TBG comme type de fichier intermédiaire
Workflow XML	Processus d'exportation d'un personnage depuis Harmony et d'importation dans un moteur de jeu en utilisant des dossiers XML comme type de fichier intermédiaire
XML	Format de fichier XML - format de fichier standard pour stocker les données structurellement, de manière similaire au format JSON