



**Toon Boom Harmony 11.1
Gaming Guide**

Legal Notices

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

Tel: +1 514 278 8666
Fax: +1 514 278 2666

toonboom.com

Disclaimer

The content of this guide is covered by a specific limited warranty and exclusions and limit of liability under the applicable License Agreement as supplemented by the special terms and conditions for Adobe®Flash® File Format (SWF). For details, refer to the License Agreement and to those special terms and conditions.

The content of this guide is the property of Toon Boom Animation Inc. and is copyrighted.

Any reproduction in whole or in part is strictly prohibited.

Trademarks

Toon Boom® is a registered trademark and Harmony™ is a trademark of Toon Boom Animation Inc.

Publication Date

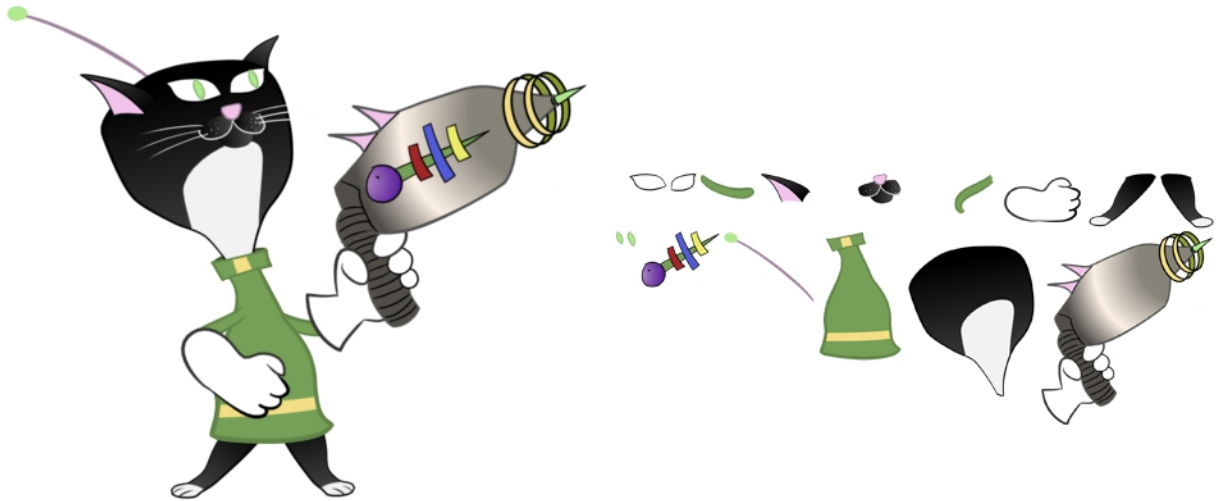
2015-02-20

Copyright © 2014 Toon Boom Animation Inc. All rights reserved.

Contents

Toon Boom Harmony 11.1 Gaming Guide	1
Legal Notices	2
Contents	3
Gaming	4
Frame-by-Frame Export	4
Raw Game Data Export	5
Creating Animation in Harmony	7
Rigging Guidelines	8
Deformation Guidelines	16
Cutter Guidelines	17
Animation Tips	18
Animating Multiple Sequences	18
Exporting the Animation	20
Additional Notes	22
Importing Harmony Animations into Unity	23
Opening the Sample Unity Project	23
Working in Unity	23
The Unity Interface	24
Game Objects	24
Using an Empty Game Object	25
Using a Plane Game Object	27
Placing a Target in an Imported Animation	28
Bounding Boxes and Collisions	29
Working with Audio	30
Clearing Preferences	32
Additional Resources	33
Troubleshooting the Toon Boom Game Engine SDK	34
Unity Questions	34
General Questions	35
Glossary	37
Index	48

Raw Game Data Export



Raw game data export is the way to go when you want to stay as light as possible! Toon Boom lets you extract all the data from Harmony to incorporate into a game engine. You can extract the skeleton information, drawing information, and keyframe animation data, as well as deformations (bones and articulations only), cutter, transparency nodes, and timing columns.

- **Advantage:** This is the lightest export, and will keep file sizes small which is ideal for mobile applications.
- **Disadvantage:** You are somewhat limited in the tools you can use in Harmony. You can use tools like Morphing and Deform, but you'll need to bake this out to drawings so they're interpreted properly in the game engine. Also you can't use advanced effects like masking.

However, even with these limitations, you can create really great cut-out character animation in Harmony and extract all the data out. By moving, rotating, scaling, and skewing the different drawing layers, you can create advanced looking animation.

When you extract the data though, you get nice, tight sprite sheets that only contain the drawings of the body parts used in your Harmony scene file. You can also support multiple animations, such as idle, run, and jump, and reuse the same skeleton and drawings when possible by simply exporting new animation data.

Toon Boom has fully integrated this solution with the Unity game development rendering engine. If you are creating your game in Unity, you have a seamless pipeline without the need to re-treat the data in your game engine.

This Gaming Guide describes the process of extracting raw game data from Harmony:

- *Creating Animation in Harmony* on page 7
- *Rigging Guidelines* on page 8
- *Animation Tips* on page 18
- *Importing Harmony Animations into Unity* on page 23
- *Placing a Target in an Imported Animation* on page 28

Creating Animation in Harmony

When creating character rigs and animation for games, there are a few things to think about before you get started. Artists and programmers should work together to make sure their needs are met.

- What platform will your game be going on? Windows, Mac, Mobile, iOS, PS, XBOX, etc.
- What game engine will you be using?
- What is the style of the game? Hand drawn, cut-out, with textures, etc.

These are just a few of the questions you need to answer before getting started. They all have an impact on how you design, build, and animate your characters.

For example, if you're planning a mobile game for smart phones, then you will most likely want to keep your game under 50 MB, so it can be downloaded without having to be on Wi-Fi. In that case, your most important consideration is to create efficient characters with very tight sprite sheets and reuse a lot of the animation to keep the file sizes small. During the process, you will need to:

- Rig and animate characters in Harmony
- Extract out the Harmony data
- Import the Harmony data into the game engine

This guide covers the entire process, including how to import the data into Unity. Keep in mind, if you're working with a custom engine, you can also process the Harmony data that's exported and use it in a custom game engine. If you need assistance with adapting data for your engine, please contact support@toonboom.com.

If you're making a game for consoles, like the PS or XBOX, then you have the freedom to create larger textures. You may want to animate frame-by-frame, or you may want to animate with a cut-out character. You can feel free to follow the flow in this guide if you want to extract the Harmony data.

If you are simply going to export on a frame-by-frame sequence, then you can use all of Harmony's tools without limitations. You can then process an exported image sequence into a sprite sheet. Harmony can easily support this process as well, though it will not be explained in this guide as there are no special guidelines required.

Rigging Guidelines

The following is a list of general guidelines to keep in mind when rigging your character. For information on how to use Harmony, refer to the Harmony [User Guide](#) or the Harmony [video tutorials](#).

As you're planning the character rig for your game, keep in mind the style of the character, and create your custom colour palette. However, there are some limitations to consider if you plan to extract the game data:

- **Create rigs in the Timeline view only.** The Network view can have complex connections that are not always interpreted well by game engines.

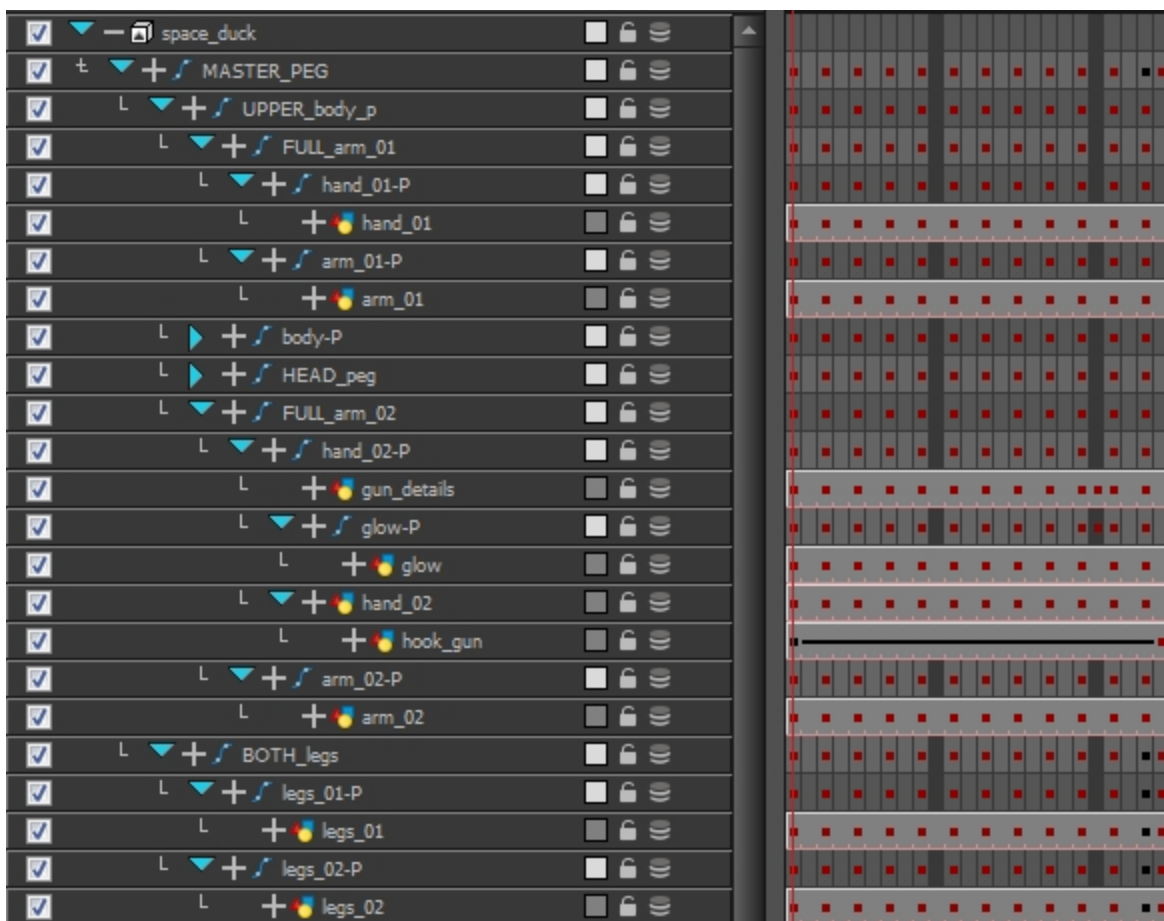
- **Nudge layers in Z space** if you need to reorder layers. However, significant Z offsets are not supported within a character rig.

- **Make every layer in your game engine a separate scene** in Harmony. If you have two characters at different depths, put them in separate scene files.

- **Character rigs in groups** at the root level of your Harmony scene will render to a single plane in Unity, but use separate sprite sheet and animation data sets. Keep this in mind for scenes in which you may have more than one character interacting with each other.

- **Don't use any effects.** Effects like masks and colour overrides are not interpreted by game engines. Use only direct hierarchy, drawing swaps, and keyframe animation.

- **Don't use 3D space.** If you want to set things up in 3D space, you can do this when you get to your game engine.
- Set your **pivot points** on Peg layers using the Rotate tool to set the pivot on the entire layer. Peg pivots are recommended over drawing pivots. You should also set the pivot points on your drawing layers, even if you don't animate on them, as this will allow you to retrieve that information later on in the game engine if you need to put a locator on a drawing layer.
- **Don't use Morphing.** This is not yet supported in game engines.
- Both Morphing and Deform can be used and then **baked out** to individual drawings. These drawings will then show up as new drawings in your sprite sheet. Be wary of doing this too often as it will increase your texture space!
- For your pegs, when you animate, use **Bezier curves** and set them to Separate. If you use 3D Path, it's heavier than Separate.



Keeping these tips in mind will allow you to create a tight, efficient 2D game character in Harmony while taking advantage of all the great tools.

Here are some things you should do:

- Create a simple parent-child relationship hierarchy in the Timeline view.
- Use peg layers to contain keyframe animation data, set to Separate.
- Use drawing layers to draw on, creating new drawings when needed.
- Use the Rotate tool to set the pivot points on the peg layers.
- Name your layers properly so if you need to fetch a specific layer's pivot point later on in the game engine, you can recognize the layer you need easily. If you have a top-level Group A, which has a child group inside it (Group B), and the drawing layer is a child of Group B, then the drawing layer is exported as A_B_DrawingLayer.

That's about it! You can use any of the drawing tools you want: Pencil and Brush tools, textured lines, solid areas, and gradients. Each individual drawing will be rendered out and assembled into a sprite sheet later.

Deformation Guidelines

Harmony can export deformation networks to the game engine XML format. Only hierarchies made of bones and articulations can be successfully exported. Although deformation comes equipped with a wide range of features, many were restrained to ensure that performance is on par with gaming industry standards. The following features are not compatible with the game engine SDK:







- Zones of influence
- Curve joints in the bone hierarchy
- Only a single pose is allowed per deformation hierarchy

The deformation in the game engine SDK does not behave exactly as the deformation in Harmony.

To comply with most game engines and maintain fast calculations, the SDK implements a linear base skinning algorithm to linearly blend the bones at articulations. You may notice some differences depending on the curvature of the articulations used.

* This feature is not currently available in the Cocos2dx implementation of the game engine SDK.

How to set this up for Deform

1. In the Deformation tab of the Preferences dialog box, deselect the **Automatically Create New Deformer Structure for Each Pose/Drawing** option.
2. Select the drawing layer on which you want to add the Deform rig.
3. In the Deformation toolbar, select the Rigging  tool.
4. In the Tool Properties, switch to the Bone  mode. This will force you to create a bone structure that always works as long as the zone of influence is not changed.
5. In the Deformation toolbar, click the Setup Mode  button, then click on the drawing where you want to create the joints for your rig.
6. In the Tools toolbar, click the Transform tool  and adjust the size of the articulations.
7. When the rig is set up to your satisfaction, select the Deform Group in the Timeline view.
8. In the Deformation toolbar, click the Copy Resting Position to Current  button.
9. Disable the Setup Mode , test the structure and animate.

Cutter Guidelines

The cutter, or mask, operation in Harmony is used for cutting off drawings with custom shapes. The game engine SDK implements both cutters and inverted cutters with the following limitations: For a sprite, only a single cutter drawing can be applied when it is rendered. This also applies to a composite of multiple matte drawings. The game engine SDK will only use the first matte drawing during rendering and discard the others. A deformed drawing cannot be cut, but a cut drawing can be deformed.

* Not currently available in the Cocos2dx implementation of the game engine SDK.

Animation Tips

When animating for games, depending on the type of game you're doing, you may need to limit your animation. For example, if you're doing mobile games and you want to keep the file sizes small and playback fast on all devices, then you should limit yourself to simple keyframe animation with as few drawing swaps as possible. If you're creating console games, you have the freedom to create more drawings and have a higher complexity. You can explore with your programmer the limitations of the platforms you're exporting to, and what your game engine supports.

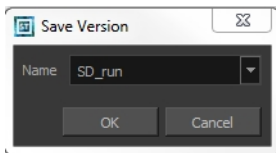
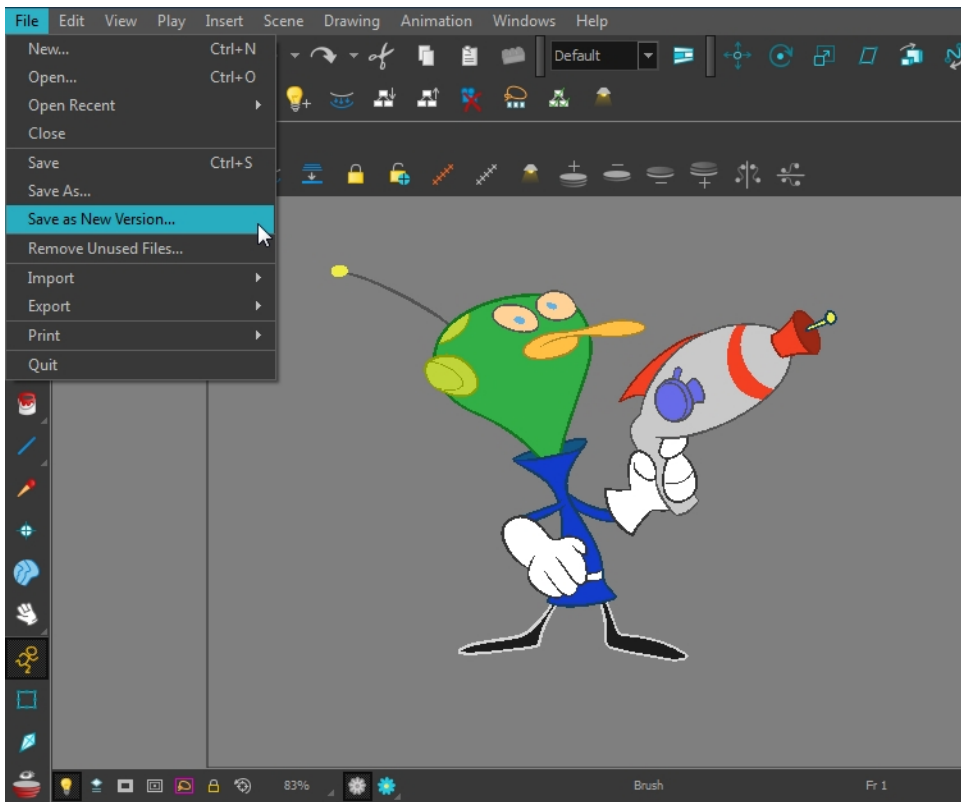
Here are some tips for efficient, lightweight animation:

- Use mainly transformations, such as move, rotate, scale, skew.
- Create additional drawing swaps when needed.
- If you use Deform or Morphing, you'll need to bake out the drawings for export. Be careful when doing this, as you may want to keep the number of drawings small. Don't bake out an entire sequence, just selected drawings.
- The bigger the drawings are in the Drawing view, the more pixels they will occupy in the texture size on the sprite sheet. When setting up your rig, make sure to not scale individual layers by using a keyframe with the Transform tool. If you want to scale things up or down, use the Select tool. This will keep things the same relative size on the sprite sheet. When you export the sprite sheets, in the script you can also set the resolution of the sprite sheet so the drawings can be downscaled for smaller devices.
- Only drawings which are exposed in the scene will be exported to the sprite sheet. For example, if you have 10 drawings in your Library view, but only two of them are showing in your scene, only those two will export out. This keeps the sprite sheet as tight as possible.

Animating Multiple Sequences

You will always have multiple animations for your characters. For example, an idle sequence, a run sequence, an action sequence, and so on. You need to work in a specific structure so you can export all of these animations to a single sprite sheet.

First, create a scene file with the name of the character, such as **Space Duck**. This is the file where you can create or import your game rig. In the top menu, select **File > Save As New Version**, and give this new version the name of the animation. For example, **Idle**.



Every time you need to do a new animation using the same character, perform a Save As New Version. In the end, you may have something like this:

Scene: Space Duck

Versions:

- Run
- Idle
- Jump
- Shoot

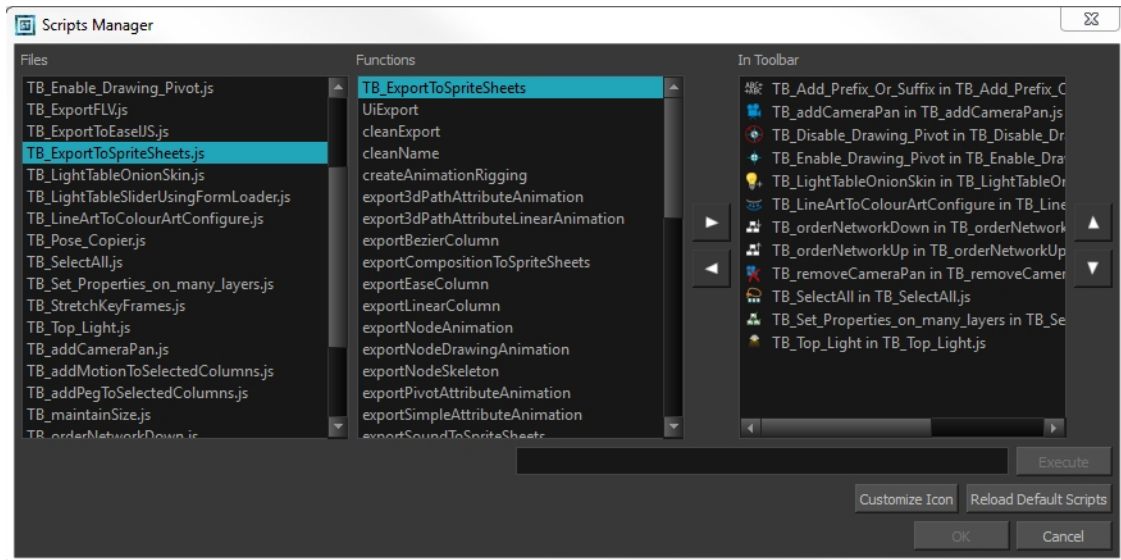
When you run the export script, it will export the drawings from the current scene into the export folder. It will also let you know if there are any other scene versions that were already exported to that folder. If so, then it will recompile the sprite sheet to include all the drawings from all the animations. This allows the maximum possible reuse of drawings.

Exporting the Animation

You'll first need to put the script in your Scripting toolbar (TB_ExportToSpriteSheets). This script is included in the latest build of Harmony. Verify that you have the latest version of Harmony.

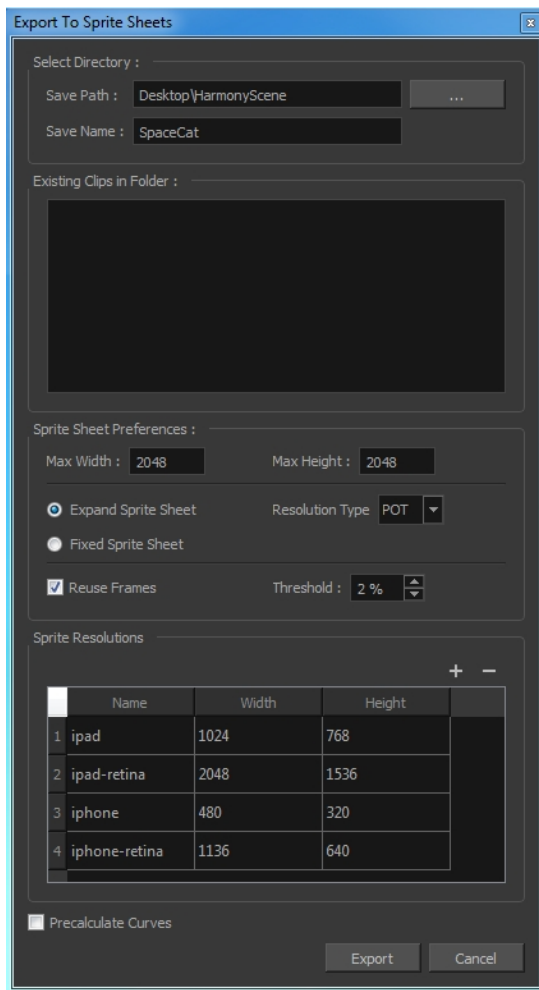
How place the script in the Scripting toolbar

1. From the top menu, select **Windows > Toolbars > Scripting**.
2. In the Scripting toolbar, click the Manage Scripts  button to display the Scripts Manager.



3. In the Scripts Manager, select **TB_ExportToSpriteSheets** from the Files column.
4. In the Functions column, select **TB_ExportToSpriteSheets** and then click the right arrow button to move add it to the toolbar.
5. Click **OK**.
6. Run the script by clicking the New icon on the right side of the Scripting toolbar.

The Export to Sprite Sheets dialog box opens.



7. Set the Save Path to the correct folder in your Unity project if you want it to update automatically. If not, you can save anywhere, then transfer the animation into your Unity project, or any other game engine you want to use.
 - It is possible to save directly to a mounted WebDAV server if you want to view your animation on your device right away (you need to have Unity player installed and WebDAV set up).
8. Set your sprite sheet preferences:

Parameter	Description
Resolution Type	<p>POT: Exports to sprite sheets that are sizes that are a power of 2. For example: 1024 x 1024. This is optimized for many graphics cards, but consumes more memory.</p> <p>NPOT: Some game engines are optimized specifically to render to non powers of two, so that it will avoid those numbers. Example: 1000 x 1000.</p>
Fixed Resolution	<p>Off: Uses the minimum size necessary up until it reaches the max resolution.</p> <p>On: Creates a texture of the specified size even if it does not</p>

	fill it up completely.
Sprite Resolutions	Size of the render of the individual sprite, when it exports each drawing out.
	<p>The exporter will export to multiple resolutions based on the settings.</p> <p>Save different animations of the same character into the same Save Name. For example, if you have an idle, run, and jump animation, these should all share the same Save Name. Think of this as the overall collection of animations. Inside you will have the different scene versions saved out, so you can reuse drawings for all the animations in that character set. You will then see each Scene Version show up as an item in the list.</p> <p>When you export an animation it exports only the drawings used in that scene. It exports all the drawings individually first. Then atlases those drawings together into a sprite sheet.</p> <p>If you save multiple animations to the same Save Name (i.e. SpaceDuck: run, idle), then it will re-atlas the sprite sheet to include all the drawings from all the animations in that folder, creating a new animation file, but reusing the same skeleton.</p>
Precalculate Curves	Makes linear values for interpolation. This increases the amount of memory used but frees up the calculation so it doesn't have to do this on the fly.

Programmers may be interested to note that the xml data exported by Harmony can be converted afterwards to a more optimized binary data structure. The binary converter is not available in Harmony yet, but it can be compiled manually in the game engine package. The C++ code that handles the data structure can be reused and parsed in your own code if you want to integrate with other game engines.

Additional Notes

- The Unity implementation creates different rendering primitives to make sure that the performances of assets exported to the game engine SDK render as before. Although the SDK supports more complex rendering, assets that do not use effects will still render at the same speed.
- The binary file format has changed. While the XML file format remains compatible, pre-existing binary assets must be reconverted to the new format to be compatible with the current game engine SDK.
- The Transparency parameter attached to the element module is exported along with the animation. You can set the transparency through the Layer Properties window (Advanced tab).

Importing Harmony Animations into Unity

At this point, if you have a custom game engine, you can take the exported data and do what you need to do with it. Or you can modify the export script to fit your convention.

If you're going into Unity, here are some things you need to do:

- [Opening the Sample Unity Project on page 23](#)
- [Working in Unity on page 23](#)

Opening the Sample Unity Project

Toon Boom includes a sample Unity project that contains all the scripts necessary to import the data exported from Harmony. Inside this project is an Assets folder which contains the following subfolders:

- Plugins (where we have loaded our plugin)
- Scenes (where you can find some demo scenes and a previewer to preview all your assets)
- Scripts (where all the Harmony scripts are)
- StreamingAssets (where all the Harmony scene files should be exported to)

All of the exporting animations from Harmony can be automatically exported to the StreamingAssets folder from Harmony by configuring the script. This way, Unity will dynamically load the most up-to-date assets as soon as they are dropped in this folder. You can also manually drop exported Harmony data into this location.

You do also have the ability to set up a WebDAV server. You can host your own server where you can store your assets, which will allow the artists working on the animation to preview the artwork without having to go through the programmer first to see what it looks like in the game. You will need to set up the server, copy the files onto the WebDAV server after you export them from Harmony, then install the Unity app on your iPad/Android device to open the files and preview them there. For more information, contact support@toonboom.com.

Working in Unity

This Gaming Guide is a basic guide on how to import Harmony projects and configure them, and includes some tips. It is not meant as a guide on how to program games.

The Unity Interface

- **Game:** Equivalent to Harmony's Camera view which shows you what you see through the game itself.
- **Scene:** Equivalent to Harmony's Perspective view, where you set the scene.
- **Inspector (right):** When you select an element, its properties are displayed.
- **Hierarchy:** This is set up on the lower-left to make it easier to set up the hierarchy of elements in your scene, and set up parent-child relationships for different game objects



- **Main camera:** This is what will show your scene, and you can find this listed in the hierarchy. To show a Harmony object, you must add a special component. With the main camera selected, in the Inspector, select **Add Component > Scripts > Harmony Camera**.

Game Objects

Basically everything that you create in Unity will be a Game Object. There are two main types of game objects you can use to import Harmony data:

- **Create Empty:** This creates a regular game object in which you can load the exported Harmony data. All the individual body parts on the sprite sheet come in as separate elements, but are treated as one whole object with a dynamically created bounding box. If you were to put a transparency on this object, you would see the overlapping transparencies of all the individual objects. But this in general creates a higher-quality render and should be the default if you're not animating the transparency of an object. With this option, direct rendering will read the depth information in the scene but not write it. Note that with this option, the rendering cannot be customized.
- **Create Other > Plane:** Renders the entire character to a plane. This renders your character to texture data. With this option, if you animate the transparency of the plane, then it animates the transparency of the rendered output, so you don't see the transparencies of the overlapping elements. However you will have one static bounding box the size of the plane that you render to. Use this option when you want to animate the transparency of a character

in a scene. Also you can write your own custom shaders. Simply set the Render Target to `eRenderTarget`. You can see an example of this in the sample project `DemoRenderTarget`.

There is a quick way to import Harmony data:

- **Create Other > Harmony Object:** Lets you browse for the exported Harmony project folder. If you've already saved this to your `StreamingAssets` folder, you can browse to it through there. Then it adds the Harmony scene to your Unity file, and sets up the rendering, audio, and animation scripts you need to get going.

To understand fully how it works, let's set it up by hand. Everything you do will be based on the Game Object you create to load your Harmony data. So let's go ahead and create a Game Object.

Using an Empty Game Object

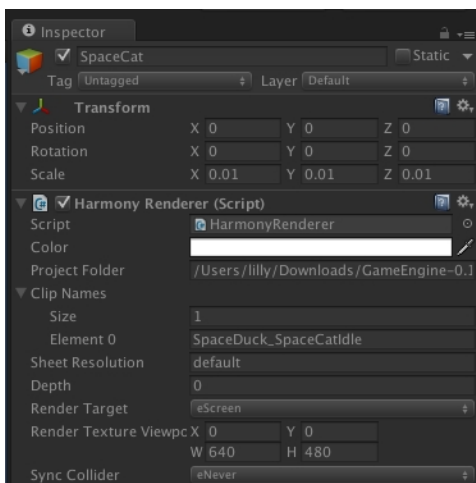
GameObject > Create Empty

You should rename this Game Object to be what you're working with, so it's more clear in the Hierarchy tab. In our example, we're using the `SpaceCat` demo file, so we'll rename the Game Object to `SpaceCat`. You can do this by clicking twice on the name and renaming it in the Hierarchy tab, or you can select the Game Object and rename it in the Inspector.

At this point, it's an empty object. Accessing Harmony data is done through scripts. First, select the Game Object. In the Inspector, go to **Add Component > Scripts > Harmony Renderer**.



When you do this, a new section appears in the Inspector called *Harmony Renderer (Script)*.



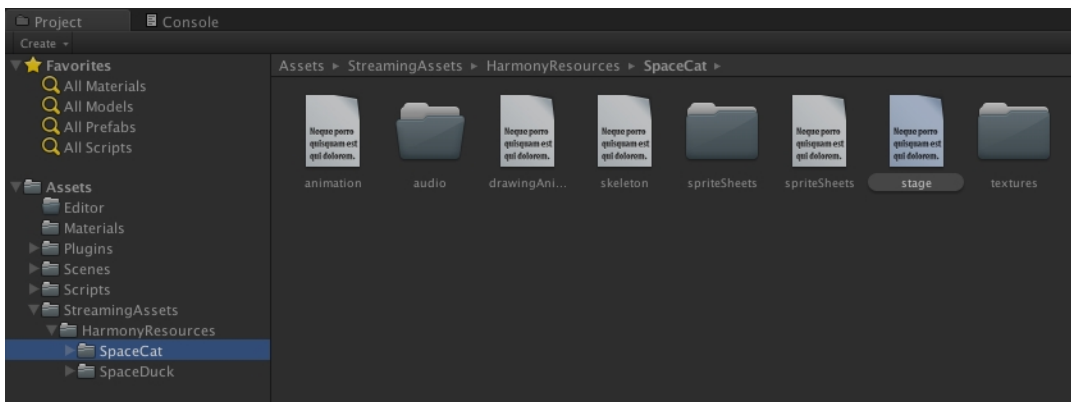
This will let you load an animation file. What you want to do is put the path of the asset relative to the HarmonyResources folder. Before this point, you will already have exported from Harmony directly to this folder, or manually copied and pasted the data in here.

You exported a scene called *SpaceCat*. In the Project Folder, place HarmonyResources/SpaceCat under the Transform information at the top, under Scale. You should set these values to 0.01 for X, Y, and Z because the default Harmony project is for an HD screen.

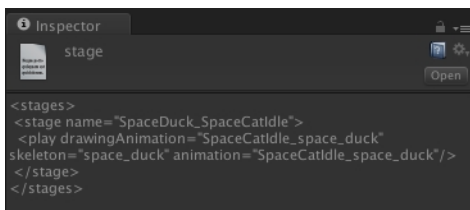
You won't see anything until you define a few parameters. First, set the Clip Name Size to 1. A second field will appear, Element 0. Enter the clip name for the animation you want to load.

If you double-check the right Clip Name to put in Element 0, you can check the output that you exported from Harmony. Double-check the stage.xml file, which was exported to the folder you specified when you did the Harmony Export. Our default export path is to StreamingAssets, so that it will load up in the game engine automatically.

Take a look in the Project tab, under Assets > StreamingAssets > HarmonyResources > [ExportName]. You'll find the data that's been exported. In this case, we're looking for *SpaceCat*. Select the file marked **stage** to load the data to double-check.



The text data appears in the Inspector.



The name to use for loading the object is the stage name. In this case, **SpaceDuck_SpaceCatIdle**.

If your Game Object has animation in it, and you want it to load the animation data from the Harmony export, you need to add one more components. With the Game Object selected, in the Inspector select Add Component and select Scripts > Harmony Animation.

If you'd like your animation to loop, then you need to set a game object to define this. Go to GameObject > Create Empty. Then rename this to *LoopAll*. Click on this game object to show the properties in the Inspector. Click on Add Component > Scripts > Loop All.

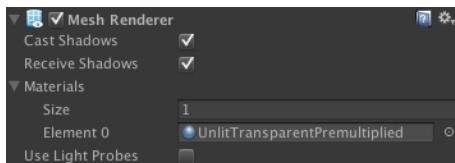
Now, when play the animation, you'll see it on the screen looping as it plays back. The Loop All command simply iterates on all game objects on runtime and asks them to loop, which is useful for

when you're testing your animation. When you are programming the actual game, you should write additional scripts to control the different elements separately.

Using a Plane Game Object

How to perform the same process using a plane

1. Select **GameObject > Create Other > Plane**.
2. Select **Add Component > Scripts > Harmony Renderer**.
3. Set Size to **1** and Element to **0** on the clip name as per previous instructions.
4. Set Render Target to **eRender Texture**.
5. Render Texture Viewport:
 - ▶ **X**: To reposition the animation correctly; should be $-\frac{1}{2}$ the width of your Harmony project.
 - ▶ **Y**: To reposition the animation correctly; should be $-\frac{1}{2}$ the height of your Harmony project.
 - ▶ **W**: Width of your Harmony project
 - ▶ **H**: Height of your Harmony project.
6. You will also want to use the custom Shader created by Toon Boom to have a transparent background with premultiplied alpha channel. This is what produces nice, antialiased drawings in your output.
7. With the Plane selected in the inspector, go to **Mesh Renderer > Materials**. Under Element 0, select **UnlitTransparentPremultiplied**.
8. If you're starting with the Toon Boom demo project, this renderer should already be loaded in your list.



You can also write your own custom shaders and use them here.

9. You will need to also add the animation component. Select the Plane. In the inspector, click **Add Component > Scripts > Harmony Animation**.

Placing a Target in an Imported Animation

It's possible that after importing the Harmony animation into the Unity scene, you may want to assign a target to some piece of the animation. For example, in the Space Duck scene, the character has a gun. Perhaps you want to attach some other action, such as shooting spheres and cubes out of the end of the gun.

You can see some examples of this in the provided Unity project. The first example in the demo scene is called *DemoLocator* which shows a simple cylinder attaching to the end of the gun.

The second example called *DemoCallback* shows the gun shooting spheres and cubes.

When you set up your character in Harmony, you can access any drawing layer in Unity later and use its pivot point to attach things. For this reason, you should set the pivot points on your drawing layers with the Rotate tool.

You can also create an empty drawing layer. Use Create Empty Drawing to create a blank drawing, ensuring that this drawing layer will be exported out. Then you can use the Rotate tool to set the pivot point on this drawing layer. In the Space Duck example, an empty drawing layer called **hook_gun** was created on which you can attach the locator.

When you access the imported Harmony character in Unity, you can access the data with a script.

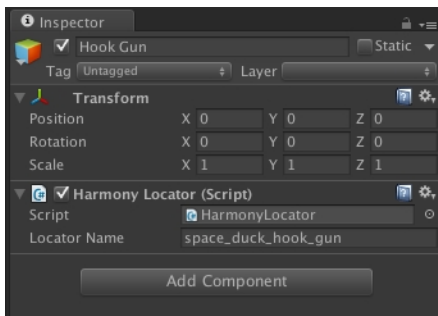
To set a target to a part of your character, first add a new Game Object to extract the position, rotation, and scale of the target. Select **Game Object > Create Empty**.

Now drag and drop the new object so that it's a child of your character. It will become indented. You should also rename it, in this case it is named **Hook Gun**.



In the Inspector of the target object, select **Add Component > Scripts > Harmony Locator**.

There's only one parameter that needs to be entered here, and this is the name of the drawing layer that you want to attach to.



To find the exact name for the Locator Name, take a look at the skeleton.xml file located in the Streaming Assets folder. You can find it in Streaming Assets > Harmony Resources > Name of Harmony File > skeleton.

```
<read name="space_duck_gun_details"/>
<peg name="space_duck_glow-P"/>
<read name="space_duck_glow"/>
<read name="space_duck_hand_02"/>
<read name="space_duck_hook_gun"/>
<peg name="space_duck_arm_02-P"/>
<read name="space_duck_arm_02"/>
<peg name="space_duck_BOTH_legs"/>
<peg name="space_duck_legs_01-P"/>
<read name="space_duck_legs_01"/>
```

This lists all the drawing layers and peg layers. In our case, we have one that we're looking for called **space_duck_hook_gun**. This is why it helps to name drawing layers with distinctive names, so you'll recognize them when you need to retrieve them later on.

If you copy the name of the layer into the Locator Name, then you should see the Transform information updating dynamically as you run the simulation.

Now you can attach anything as a child of this Locator Object. For testing purposes, you could attach a cylinder for example. In a real example, you'd want something to shoot out of the end of the gun, so you can see the Callback example to verify how that works.

- [Bounding Boxes and Collisions on page 29](#)
- [Working with Audio on page 30](#)

Bounding Boxes and Collisions

Extracting the bounding box information is useful when you want to make something collide with your character. When you put a 2D character into a 3D scene, you may want to use Physics to make the 2D character collide with the 3D plane. To use the following scripts, you will need to be using Unity 4.3 or higher.

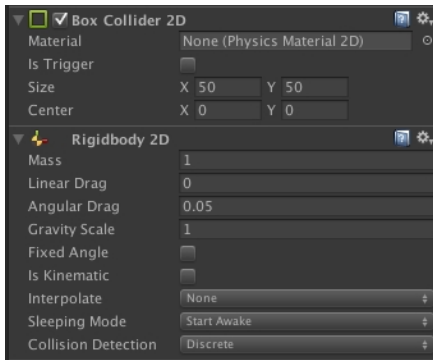
In the Demo scene, open the DemoPhysics example to see collisions and bounding boxes in use. In this example, there's a 3D plane and some 2D characters. The 2D characters have a collision defined with their bounding box, and they are also reacting to the physics of the scene to fall with gravity.

Colliders were improved to provide several different types of colliders for use in Unity. You can select polygon colliders and box colliders for a more accurate bounding box.

How to enable a collision based on the bounding box of the character

1. Select your imported character.
2. In the inspector, select **Add Component > Physics 2D > Box Collider 2D**.
3. To enable the Physics on the character, go to **Add Component > Physics 2D > Rigid Body 2D**.

You can also use 3D physics on your imported files as well.



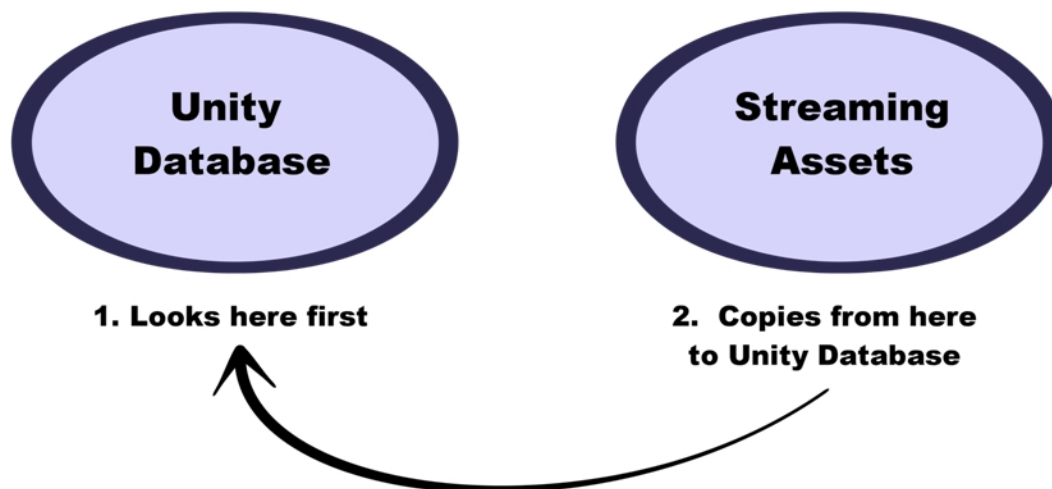
Now adjust the variables and see the results!

Working with Audio

When you create scene files in Harmony with audio, then the audio is exported into the project folder. However it's important to understand how the game is built to determine the best way to handle audio.

When a game is built from Unity, it creates its own database for holding the assets. Within the export, the Streaming Assets folder is preserved intact. This is what enables the Toon Boom plugin to access those assets on the fly, and render them in the camera to create the final animation you see on the screen.

However when it comes to audio, the game first looks for the audio in the Unity database. So this is the most efficient place to store the audio for the real game.



If audio can't be found in the Unity database, then the game looks in the Streaming Assets folder to find the audio. When it does, it will actually copy the audio on-the-fly back into the Unity Database. Since this action is quite costly, don't leave the audio in the Streaming Assets folder for the final game. It's fine to leave it in there while you're testing, but not for the final game.

The audio should therefore be copied back into the Unity database, either manually or by automating this process with a script.

Clearing Preferences

If you were already using a previous build of the Gaming Pipeline, you will need to clear out the old scripts from your preferences so you can use the new scripts.

How to clear your preferences

1. Close Harmony if it is open.

Windows: Go to [path on Windows]

Mac: Go to /User/Library/Preferences/Toon Boom Animation/Toon Boom Harmony

2. Delete the **1110-scripts** and/or **1100-scripts** folder.
3. Restart Harmony and test the export.

Additional Resources

The Toon Boom gaming pipeline is evolving constantly! Stay in contact with us to find out the latest.

- Check out our Gaming Demo Video: www.youtube.com/watch?v=eu32sk8Jl3M.
- Check out our Tip of the Week: www.youtube.com/user/ToonBoomTips
- Contact our Sales team for more information: sales@toonboom.com.
- Contact our Support team for technical inquiries: support@toonboom.com.

Troubleshooting the Toon Boom Game Engine SDK

Unity Questions

Q. My Harmony asset does not appear in the Unity Editor.

A. There are several possible reasons for this. Check the following:

- ▶ Has the HarmonyRenderer plugin been copied to the **Assets/Plugins** directory? The plugin is named **HarmonyRenderer.bundle** on Mac OS X and **HarmonyRenderer.dll** on Windows.
- ▶ Is the clip name valid? Make sure the clip name exists in the stage.xml file.
- ▶ Are the binary files up to date? If you updated the Toon Boom Game Engine SDK to a newer version, make sure to regenerate the binary files if you use them so they're compatible with the SDK.

Q. My Harmony asset does not appear in iOS or Android when exported from Unity.

A. There are several possible reasons for this. Check the following:

- ▶ Make sure the Harmony assets are copied in the **StreamingAssets** directory. Since the HarmonyRenderer plugin cannot use Unity's asset database, the Harmony assets were packaged as plain files so they are accessible to the plugin.
- ▶ Is the plugin for iOS the same version as the one used for the editor? The plugin for iOS is in the **Assets/Plugins/iOS** directory.
- ▶ Is the clip name spelled correctly? Mac OS X and Windows aren't usually case sensitive file systems, but iOS and Android are. This implies that an asset that is visible on the editor will not appear on the mobile platform because it cannot be found.
- ▶ Make sure the size of your sprite sheets does not exceed the hardware limit of the device you're working on. Otherwise, the texture might not appear at all.
- ▶ Is the path to the project folder an absolute path? If it is, it might not work on the mobile device as that path will not exist on the platform. Make sure to use relative paths to the **StreamingAssets** directory.

Q. My Harmony asset does not composite properly with other 2D sprites in Unity. The 2D sprites always appear behind the Harmony assets even if I change the order index in the game objects.

A. The 2D sprites from Unity and the Harmony assets cannot currently composite together. There are, however, several solutions for working around this problem:

Use 2D textures instead of 2D sprites. The 2D textures, when attached to a geometry in Unity, will composite like the other 3D elements in the scene and you can change the compositing order by incrementing or decrementing the Z value of the transform.

Use multiple cameras to render your scene in parts. Here's how:

Harmony Camera

1. Create a new camera.
2. Make sure you set the Harmony game objects to a specific layer (top-right of the game object inspector).
3. In the camera, set the culling mask to this layer (and the other layers you want rendered in the background).
4. Add the HarmonyCamera component to this camera so the Harmony game objects render properly.

Sprite Camera

1. Create a new camera.
2. Make sure to set the 2D sprite objects to a specific layer.
3. In the camera, set the culling mask to this layer (and the other layers you want rendered in front of the 2D sprites).
4. Make sure to set the clear flags to **Depth Only** or **Don't Clear**.
5. Set the depth to be higher than the Harmony camera.
6. Afterwards, if you move the camera, make sure to move both cameras simultaneously if you want to keep game objects in sync.

Q. How can you specify the frame rate of the animation? My animation does not play at the same frame rate in Harmony and in the Game Previewer in the Unity Editor.

A. The Game Previewer previews the animation at a default of 24 frames per second. In your game, using the HarmonyAnimation component, you can schedule animations at the speed you want.

Example

```
HarmonyAnimation animation = GetComponent<HarmonyAnimation>();
if (animation != null)
{
    animation.PlayAnimation( frameRate, clipName );
}
```

Q. I cannot see my character in the scene view in Unity. Is there any way to preview the Harmony asset?

A. Since the rendering of Harmony assets is implemented in a plugin, it does not update any Renderer component in the game object, and so cannot be shown directly in the scene view.

However, you should still be able to preview your assets in the game view (even when not playing). To help find and position assets, we also added bounding box shapes rendered in the editor and the game view.

Q. How is memory management handled in Unity for Harmony assets?

A. Most of the memory extensive algorithms are executed in the C++ plugin. The data structure is maintained alive as long as a game object refers to it. This means you can instantiate as many clones of a character without reloading the data. Upon destruction, the game object will issue a cleanup call to the plugin and delete its associated rendering object. The sprite sheet and animation data are kept alive as long as a single game object uses them and a cleanup call is issued once they are destroyed.

Q. The Harmony assets memory does not appear in the Unity profiler. Can we profile the assets memory?

A. The Unity profiler does not pick up Harmony assets in its traces. Since the Harmony game object refers to a C++ plugin for its asset management and rendering, its memory cannot be recorded in the profiler. Profilers such as the one provided with XCode or Visual Studio should allow you to track your application's memory including the Harmony assets and should also provide a more accurate result.

General Questions

Q. How can you specify the resolution used for the drawings in the sprite sheet?

A. The drawing resolution is not fixed in Harmony. Since you are exporting the scene to multiple sprite sheets in multiple resolutions, every drawing will have a different resolution in the end. Now, you can use the Drawing view to see the resolution of a drawing compared to the rest of the scene. By enabling the light table in the Drawing view, you will see all your scene's drawings untransformed in the same proportion as they will appear in the sprite sheet. More specifically, the resolution used in the sprite sheet for a drawing is the size it appears untransformed in the Camera view, scaled to the sprite sheet resolution ("Sprite Resolutions" in the Export Sprite Sheet dialog box).

Once you know that, you can easily adjust the size of a single drawing. Use the Select tool to scale down a drawing and use the Transform tool to scale it back up so it does not change visually in the Camera view. This should allow you to alter the size of your drawings in the sprite sheet.

Q. The assets take time to load on mobile devices. Is there any way to speed up the process?

A. Harmony exports an interchange XML format that is intended to be compatible with multiple game engines. However, reading data from the XML files is slow and this is mostly intended to test the assets in your project. At some point, they should be converted to a binary compact data structure that can be read into memory a lot faster than the XML.

You can use the Xml2Bin tool provided with the game engine SDK to convert your XML files to binary. This tool is in the `Utils/Xml2Bin` folder and a precompiled version for Mac OS X is available in the `Utils/macosx` folder.

Q. There are problems with the character I exported in the game engine. Pieces are not exported or are not showing in the proper order. What can I do ?

A. To integrate with the game engine SDK, all animations in Harmony must follow certain guidelines. Here are some of the key points for building a character that is compatible with the rendering in the game engine SDK:

- Build your rig in the timeline. The game engine SDK data structure is similar to what is displayed in the timeline and an advanced network rig might not always export properly.
- Because every top level node in your scene will render separately in the game engine SDK, they may not intersect. A top level node is also isolated in its separate sprite sheet to allow the artist better memory segmentation of the character. If you created a character with top level nodes and this was not the intended result, create a top level group or peg to group them all together.
- Use micro-Z ordering if necessary. With no Z ordering, a child layer will appear behind its parent (as it does in Harmony). To change the ordering, add some Z offset to the child layer.
- The export script does not handle plain bitmap images. Toon Boom vector drawings and Toon Boom bitmap drawings are supported though.
- The export script does not handle symbols.

Glossary

alpha channel	The image's channel carrying the transparency information. An image already has three channels: red, green and blue (RGB). The alpha channel is the fourth channel (RGBA). The matte, or the transparency information, is stored in this fourth channel. An image without an alpha channel is always opaque.
animatic	A movie with sound that is developed from the storyboard. The storyboard panel is exposed for the duration of the scene and at times, the characters are placed on a trajectory to indicate motion. The camera moves are also animated. The animatic is used to determine the rhythm of a project and provides a good overview of the project before beginning production.
animation	A simulation of movement created by displaying a series of pictures or frames.
anime	An animation style known for its sinister and dark feel; popular in Japan.
arc	Action rarely occurs in a straight-forward manner; rather it typically unfolds in what storytellers refer to as an arc. The purpose of a story arc is to move a character or a situation from one state or scenario to the next.
aspect ratio	The relationship between the width and height dimensions for any scene, frame or film format. Television ratio is 4:3 and widescreen ratio is 16:9.
auto-feed	An automated method of feeding drawings into a scanner in which multiple drawings are stacked into a sheet feeder. When the scanner is activated, the drawings are scanned consecutively, without further user intervention.
automatic lip-sync detection	Automatically mapping drawings in an element to the mouth chart generated for a sound. This can save time when lip-syncing a voice track.
axis	An imaginary line around which an object rotates. For 2D graphics, there are two axes: <ul style="list-style-type: none"> • X: Horizontal • Y: Vertical For 3D graphics, there are three axes: <ul style="list-style-type: none"> • X: Horizontal • Y: Vertical • Z: Depth
background	The part of a scene that is farthest to the rear. The background is the artwork, or decor, against which the animation takes place.
Bézier	A method of defining curved lines invented by French mathematician

	<p>Pierre Bézier. A Bézier curve is a mathematical or parametric curve. Bézier curves use at least three points to define a curve.</p> <p>In Toon Boom, a function can be hooked to a Bézier curve and vary along with the curve value information. Bézier curves are also very useful in vector graphics. They are used to model smooth curves and can be scaled indefinitely.</p>
bitmap	<p>An image composed of pixels with a single resolution (size). If it is enlarged too much, it will lose definition and pixels will begin to appear. This is known as <i>pixelation</i>.</p>
breakdown	<p>In cut-out animation, breakdown is the action of breaking a character into pieces to create a puppet with articulations. To break down a character, the artist cuts part, such as hands and arms, from the character's model and pastes them in separate layers. Next, the joints are fixed and the pivots set.</p> <p>In traditional animation, a breakdown is an animation pose generally found between two key poses. The key poses are the main poses in an animation and the breakdowns are secondary poses, ones that help to describe the motion and the rotation curve.</p>
camera shake	<p>Camera shake occurs in a scene when the camera moves slightly and quickly in several directions. This gives the impression of an impact, vibration or, for example, bumps on the road.</p>
caption	<p>In a storyboard, a caption is a text field containing dialogue, effects, sound, or slugging information.</p>
cel	<p>In traditional animation, a cel, also known as a <i>celluloid</i>, is a transparent sheet on which the animation is inked and painted before being sent to the camera. The picture's outline is drawn on the front of the cel and then coloured along the back.</p> <p>In Toon Boom, a cel is an individual space encountered in an Xsheet's column, from which you can expose a drawing or a function's coordinate.</p>
character design	<p>Each character in an animated film is drawn from multiple angles in poster-style format, called a <i>model sheet</i>, which serves as a reference for the animators.</p>
clean up	<p>After rough drawings have been tested and approved, all the noise in the image (excess lines, notes, etc) is removed to create final drawings which can be inked, painted and shot. The cleanup process refers to either tracing a clean line over a rough drawing to achieve the final version, or removing dirt and extra lines left by the scanning process.</p>
CMYK	<p>Cyan, Magenta, Yellow, Black. Refers to the process used by printers to define colour on the printed page.</p>
colour card	<p>A Colour Card is a solid colour card that is the same size as the camera. The Colour Card can be used to fill the background with a solid colour when there is no background image included.</p>

colour model	In animation, a colour model is the official colour design that must be used to paint the animation. A model is the definitive character, prop, or location design that each artist must follow for the production.
colour wheel	A display of the colour spectrum in the form of a circle.
compositing	Compositing is the action of incorporating all of a scene's elements to create the final result prior to rendering. For example, the compositing artist will import all the animation sequences, background, overlays and underlays in the scene and position them correctly. The artist will then set the camera frame and animate it, if needed. Finally, the animator will create all the computer-generated effects for the project.
cross dissolve	An effect used to fade two scenes, one into the other.
cut	A direct transition between two scenes. When a cut is used, there are no transition effects inserted to pass from one scene to the next. The first scene ends and the second one starts immediately.
cut-out animation	The action of animating characters made of several pieces by moving them around frame by frame. Cut-out animation can either be computer generated or done traditionally using paper.
cycle	A group of images that together make up an action, such as walking. A cycle is an action repeated as a loop over a period of time. It can be a series of animated drawings or keyframes.
dialogue	The text spoken by a character in a movie or animation.
dope sheet	Used by animators, directors and other members of a crew to track the sequence and timing of images, dialogue, sound effects, sound tracks and camera moves. Also known as an <i>exposure sheet</i> or <i>Xsheet</i> .
doping	To assign a particular drawing to a range of frames.
DPI	Dots Per Inch is the standard measure of resolution for computerized printers. It is sometimes applied to screens, in which case it should more accurately be referred to as <i>pixels per inch</i> . Either way, the dot is the smallest discrete element making up the image.
ease/velocity	In animation, the ease, also known as <i>velocity</i> , is the acceleration and deceleration of a motion. It can be a motion created by a function curve, or a series of animated drawings. Other common terms for ease-in and ease-out are slow-in and slow-out.
ease-in	Gradual acceleration in the action. Another common term for ease-in is slow-in.
ease-out	Gradual deceleration in the action. Another common term for ease-out is slow-out.
establishing shot	A scene in which the viewer can see the whole area in which a sequence is happening. For example, if a child is playing on the ground in front of his house, the establishing shot would be a scene

	where the viewer can see the house, the ground, a part of the street and the buildings around the central point of action. This helps the viewer understand the story location and scene orientation.
exposure	In animation, an exposure is the number of cels on which a drawing appears in the scene. For a drawing to appear longer, the exposure must be extended over a greater number of cels.
exposure sheet (Xsheet)	<p>The exposure sheet or Xsheet, is a sheet with several vertical columns and horizontal frames used to indicate a scene's timing.</p> <p>Each column represents a scene's layer. In each column, the drawing's numbers are indicated and repeated over the particular amount of frames they need to appear.</p> <p>The exposure sheet is used by animators, directors and other members of a crew to track the sequence and timing of images, dialogue, sound effects, sound tracks and camera moves. Also known as a <i>dope sheet</i>.</p>
fade in/fade out	Fade in or fade out is a transition effect used to open or close a sequence. A fade in occurs when the first scene appears progressively, from complete transparency to its complete opacity. A fade out occurs when the last scene progressively disappears, going from complete opacity to complete transparency.
fast-in	Dramatic acceleration at the start of the action.
fast-out	Dramatic acceleration at the end of the action.
field	In animation, a field is a measurement unit used to calculate motion, registration and camera positioning. A standard animation scene will vary between 6 to 12 fields.
field chart	A guide containing all the field units that animation and layout artists use to determine a scene size or camera motion.
film-1.33	Use this resolution setting for the widescreen film format that conforms to the standard 4:3 pixel aspect ratio.
film-1.66	Use this resolution setting for the widescreen film format that conforms to the 16:9 pixel aspect ratio. (The pixels are wider than they are high).
flipping	In traditional animation, flipping is the action of going through the drawings of an animation sequence very quickly in order to see the animation in motion. Flipping can also be the action of creating a mirror transformation of an object.
follow-through	The follow-through is the secondary motion caused by the main action. For example, a character wearing a cloak is running. The main action is the body running. This will cause the cloak to follow the motion, although it will not move at the same time, but react a few frames later and follow the main motion curve.
forward kinematics	Forward kinematics is a feature used to animate principally 3D characters and cut-out puppets with hierarchy. It is used to animate a puppet from one of parent parts, such as a shoulder, and make

	the rest of the arm move with it as a single piece.
frame	A frame is a single photographic image in a movie. In traditional animation, the North American standard generally contains 24 frames per second, while in Europe the standard is 25 frames per second.
frame rate	<p>This is the measurement of the frequency (rate) at which an imaging device produces unique consecutive images, called frames. The term applies equally to computer graphics, video cameras, film cameras, and motion capture systems.</p> <p>Frame rate is most often expressed in frames per second (FPS) and in progressive-scan monitors as hertz (Hz).</p> <p>The frame rate is the speed at which the frames are played. They are generally calculated by frame per second. For example, a scene could be played back at 12, 24, 25, 30 or 60 frames per second or any other number</p>
functions	A function is a computer generated motion, trajectory or path that elements, other trajectories and effects parameters can be attached to. The function can be controlled by adding keyframes and control points on the function curve.
gamut	The range of colours that a particular device can represent.
HDTV	High Definition Television delivers a higher quality image than standard television does because it has a greater number of lines of resolution. To take advantage of the superior quality and make full use of your resolution setting, your output device must be compatible with HDTV technology.
hold	This is a frame in the animation in which the character maintains its position without moving. A hold can be created between any two keyframes.
HSV	Hue, Saturation, Value. A method of defining colours in terms of hue (tint), saturation (shade) and value (tone or luminance).
in-between	The drawings that exist between the key poses. These are drawn to create fluid transitions between poses.
ink and paint	The ink and paint process is the action of painting the empty zones and colouring the lines on the final animation drawings, while following a colour model.
interpolation	In animation, the interpolation is the computer generated motion created between two keyframes. You have the choice to create interpolation, or not, between your keyframes.
jump cut	A jump cut is a jerky cut between two scenes. Typically, a jump cut is not visually pleasing. It is generally caused by one scene ending, and a second one starting, with similar a image. The lack of difference causes the eye to see a little jump between the two scenes.
key pose	Important positions in the action defining the starting and ending points of any smooth transition. Keys, or key poses, are the main

	<p>drawings in an animation sequence describing the motion. For example, if an arm is waving, the keys will be of the arm at one extremity of the wave motion and the other extremity. By flipping those drawings, the animator can see the skeleton of the motion without having all of the drawings.</p>
keyboard shortcuts	<p>One or more keyboard keys which, when used, cause an operation to be performed. Keyboard shortcuts are used throughout the Toon Boom software and form an integral part of the workflow. It is, in most cases, possible to customize the shortcuts in the Preferences dialog of the software.</p> <p>The shortcuts are written as follows in the Toon Boom user documentation: Each key in a sequence is shown inside square brackets as in: [Ctrl]+[A]. The brackets "[]" separate the key from the plus sign (+). Neither the brackets, nor the plus sign are part of the sequence. To use a shortcut, press the key and the character simultaneously.</p>
keyframe	<p>Important positions in the action defining the starting and ending points of any action. A keyframe is a computer generated position at a specific moment (frame) on a given trajectory.</p>
layer	<p>In animation, a layer is an individual column, level or character. A scene's layers are superposed to form the final image.</p>
layout	<p>The layout process is the communication step between the storyboard and the animation. The layout and posing process is the action of putting the storyboard on model, that is drawing the character following the design in the model pack, so that the animator can start his work. The layout artist will draw the background, create the camera and field guide matching the scene and the camera motion. Lastly, he will draw on model the main action poses.</p>
layout and posing	<p>The action of putting on model, that is, at the right scale, the storyboard for the animator to start his work.</p>
library	<p>A library is a storage area containing templates and assets that can be reused in any project or scenes.</p>
light table	<p>The Light Table feature allows you to see the other layers in transparency while you are working on a particular one in the Drawing view.</p>
line of action	<p>Direction that the action will follow. Also known as the Path of Action</p>
lip-sync	<p>The lip-sync is the character's mouth synchronization with the dialogue sound track. Frame by frame, the mouth will be adjusted to fit the sound to give the illusion of the character is speaking.</p>
low resolution	<p>This format is ideal for videos destined for the web, where size and fast download of a video file might take precedence over quality. A low-resolution image is one that lacks fine detail.</p>
manual lip-sync detection	<p>The manual swapping of mouth position drawings to match a voice</p>

	track. For this process, both sound scrubbing (listening to a sound wave broken up frame-by-frame) and drawing substitutions from the Library view are used.
master palette	A master palette is a group of colours attributed to a character or a prop. The palette is used throughout the entire production to maintain consistency in the look and to ensure that the same colours are used throughout the production. Also known as palette.
model/colour model	In animation, a model is the definitive character, prop or location design that each artist must follow for the production. A colour model is the official colour design that must be used to paint the animation.
motion keyframe	In Toon Boom, the motion keyframe is a keyframe with computer-generated interpolation.
mouth chart	Adding a lip-sync to a project can really enhance its quality and storytelling. However, it can be difficult to shape a character's mouth so that it matches the sound at the precise frame. To solve this problem, Toon Boom provides a lip-sync feature which analyses the contents of a sound element and generates a mouth chart based on the eight animation phonemes (A, B, C, D, E, F, G, and X, which is used to represent silence).
multiplane	The effect of passing through multiple levels of drawings to create a sense of depth in a shot. A multiplane is a scene in which the layers are placed at different distances from the camera so that when the camera moves, a depth illusion occurs. With a multiplane, all the perspective and scale is calculated automatically.
NTSC	The standard analogue television broadcasting system used in North America. NTSC conforms to North American standards on how rectangular pixels are displayed for computer and television screens.
nudge	A small push (left, right, up, down, forward or backward) done with the keyboard arrow keys on a selected element. Nudging is used to move a selection very slightly and precisely.
onion skin	A feature that lets you see the previous and next drawings of a sequence.
overlay	A part of the scene environment, such as a chair or a bush, that is placed in front of the main animation.
PAL	A resolution that works best with the European format for television and computer screens, as the rectangular pixels are displayed at a different orientation.
palette/master palette	A palette or master palette is a group of colours attributed to a character or a prop. The palette is used throughout the entire project to maintain a consistency in the look and avoid the colour changing during the animation. Also referred to as a master palette.
palette style	A palette style is a second version of an existing palette with a slight change in the tint and value. A palette style can be used to create

	the night version of a palette. It may also be called a clone palette.
pan	To move the camera across the scene in any direction.
panel	In a storyboard, a panel is a frame in a shot. A shot can be composed of one or several panels.
paperless animation/tradigital	The paperless animation process is the action of animating digitally. The main paperless animation process is to draw, frame by frame, the animation directly in the software.
passing position	When drawing a walk sequence for a character, the passing position is the point at which one leg passes the other.
path of action	Direction that the action will follow. Also known as the Line of Action.
peg	In traditional animation, a tool used to ensure accurate registration of action as cel layers move. In digital animation, in which you are doing a more advanced puppet rigging, you can use peg layers. Peg layers are trajectory layers that do not contain drawings. They are motion paths that you can use to add path articulations. For the latter, you can also use the Inverse Kinematics tool.
phoneme	Unit of sound in a language.
pivot	A pivot is the point around which a peg or a drawing rotates.
pixel	Smallest element of an image displayed on a monitor or TV screen. Pixel, short for Picture Element, is a single point in a graphic image. It is a small sample of an image, a dot, a square, or a very small section made out of smooth filtering. If you zoom in close enough on a digital image, you will see the pixels, which look like small squares of different colours and intensity.
pose-to-pose animation	The pose-to-pose animation process is the action of creating all the main action poses, called key poses, and then placing the secondary poses between the keys. The secondary poses are called breakdown. Finally, the animator fills the gaps with the in-between drawings to achieve a smooth animation.
rendering	The final step when animating by computer. During rendering, the computer takes each pixel that appears on screen and processes all of the components as well as adding motion blur before it produces a final image. In animation, the rendering process is the action of calculating the final images after the compositing process.
resolution	The resolution is the size of a scene, generally calculated in pixel. For example, the NTSC resolution is 720 x 480. Resolution type should match your final output: HDTV, film-1.33, film-1.66, NTSC, PAL, low.
RGB	Red, Green, Blue: method of defining colour by specifying amounts of these three colour components.
rigging	The rigging process is the action of attaching the cut-out puppet parts one to the other.

rotary table	The Rotary Table is equivalent to the animation disk/table and allows one to rotate the workspace to be more comfortable while drawing.
rotoscoping	Is an animation technique in which animators trace over live-action film movement, frame by frame, for use in animated films. The act of sketching over live-action footage to create an animated sequence.
roughs	The roughs are the skeleton sketch of an animation or a design. Roughs mainly consist of sketch lines and shapes, though they can also contain design details.
safety area	In animation and movie parlance, the safety area is the zone at the centre of the scene's frame, one safe from being cropped by the TV frame. As a TV frame cuts a margin off the original frame size, maintaining a safety area ensures that the scene's main action will remain clearly visible once the film is screened on television.
scene	A scene is a shot in a movie or show. A sequence is composed of several scenes. A scene changes to another scene by a simple cut, or a transition.
script	The script is the original text containing all the movie or show information. In animation, the script contains all of the location descriptions, dialogues, time and more. A project starts with a script.
sequence	In animation, a sequence is a series of scenes or shots forming a distinct part of the story or movie, usually connected by unity of location or time.
shot	A shot is a scene in a movie or show. A sequence is composed of several shots. A shot changes to another shot by a simple cut, or a transition.
slow-in	Gradual acceleration in the action. Another common term for slow-in is ease-in.
slow-out	Gradual deceleration in the action. Another common term for slow-out is ease-out.
slugging	In Storyboard Pro, slugging refers to indicating the start and stop times of dialogue and relevant actions.
sound scrubbing	A process that lets you hear sound in real time while you move the playhead forward or backward. This is very useful for finely-tuned lip-syncing.
stop-motion keyframe	A stop-motion keyframe is a keyframe with no computer generated interpolation.
storyboard	A visual plan of all the scenes and shots in an animation. The storyboard indicates what will happen, when it will happen and how the objects in a scene are laid out.
straight-ahead	A technique in which an entire sequence is drawn from the first position to the last, in order. There is very little planning in this methodology, and where the character ends up and how it gets

	there can be a surprise for both the audience and the animator. While this approach is a lot more spontaneous and creative, it can create inaccurate results.
strokes	Strokes are invisible vector lines forming the drawing zones. They can be adjusted with Bézier handles.
tablet/pen	Device used in conjunction with, or instead of, a mouse to move a mouse pointer (sometimes referred to as the cursor) around the computer screen.
template	In Toon Boom, a template is an asset stored in the library, one that can be reused in any project. A template can be a drawing, a series of keyframes, a sound file, a panel, a cut-out character, an effect, a trajectory, an animation, or anything else used in the software.
thumbnails	A thumbnail is a very small image used as a reference or indicator.
timecode	Timecode is timing information printed on a movie clip to indicate what scene, hour, minute and second is currently displayed on the screen.
timeline	The timeline is a horizontal representation of the scene's elements, timing and keyframes.
trace and paint	After the rough animations have gone through cleanup and a final line or pencil test, each drawing is traced and painted for the final animation. In today's digital world, this may be done in a variety of ways other than via the traditional celluloid or acetate methods.
track breakdown	The soundtrack for animated film is broken down into individual sounds documenting the precise frame-by-frame position of each sound.
traditional animation	The traditional animation process is the action of drawing on paper all of the animation sequences, before either scanning them or inking them on cels.
trajectory	A computer generated path or trajectory that elements can follow. The trajectory can be controlled by control points, keyframes and velocity.
transition	A transition is an effect placed between two scenes as they pass from one to the other. Common transition effects are cross-dissolve and wipe.
underlay	In animation, an underlay is a specific part of the decor placed behind the main animation.
aspect ratio	The aspect ratio describes the shape of the grid unit. A square grid unit would have the ratio 1:1, whereas a grid unit of aspect ratio 4:3 is a unit with one side 1.33 times as big as the other side.
vector	A vector-based image is composed of points and Bézier curves. The computer reads the points and traces the segments, linking them to reproduce the image shape. There is no fixed size or resolution in a vector image. The graphic can be enlarged and distorted as much as desired and the system will simply recalculate the segments and

	rebuild the shapes. Vector images are translated and displayed in pixels once the calculation is done.
velocity/ease	In animation, the velocity, also known as ease, is the acceleration or deceleration of a motion. This can be achieved by a function curve, or via a series of animated drawings. Other common terms for ease-in and ease-out are slow-in and slow-out.
walk cycle	To avoid making innumerable drawings, animators routinely make a walk cycle for their character. This comprises a series of drawings "on the spot" that describe the walk for that character. The illusion of movement is created via the use of background pans.
workspace	In Toon Boom, the workspace is made up of the views, toolbars, and menus.
Xsheet (exposure sheet)	<p>The Xsheet or exposure sheet, is a sheet with several vertical columns and horizontal frames used to indicate a scene's timing.</p> <p>Each column represents a scene's layer. In each column, the drawing numbers are indicated and spread over the specific amount of frames they need to appear.</p> <p>The exposure sheet is used by animators, directors and other members of a crew to track the sequence and timing of images, dialogue, sound effects, sound tracks and camera moves. Also known as a Dope Sheet.</p>
zone	An area which can be painted with colour.

Index

alpha channel
 _defined 37

animatic
 _defined 37

animation
 _defined 37

anime
 _defined 37

arc
 _defined 37

aspect ratio
 _defined 37
 defined 46

auto-feed
 _defined 37

axis
 _defined 37

background
 _defined 37

Bezier
 _defined 37

bitmap images
 _defined 38

breakdown
 _defined 38

camera shake
 _defined 38

caption
 _defined 38

cel
 _defined 38

celluloid
 See cel. 38

character design

 _defined 38

clean up

 _defined 38

CMYK

 _defined 38

colour card

 _defined 38

colour model

 _defined 43

 defined 39

colour wheel

 defined 39

compositing

 defined 39

cross dissolve

 defined 39

cut

 defined 39

cut-out animation

 defined 39

cycle

 defined 39

dialogue

 defined 39

dope sheet

 defined 39

doping

 defined 39

DPI

 defined 39

ease

 defined 39

ease-in
 defined 39

ease-out
 defined 39

establishing shot
 defined 39

exposure
 _defined 40

exposure sheet
 _defined 40, 47

fade-in
 _defined 40

fade-out
 _defined 40

field
 _defined 40

field chart
 _defined 40

film-1.33
 _defined 40

film-1.66
 _defined 40

flipping
 _defined 40

follow-through
 _defined 40

forward kinematics
 _defined 40

frame
 _defined 41

frame rate
 _defined 41

functions
 _defined 41

gamut
 _defined 41

HDTV
 _defined 41

Hold
 _defined 41

HSV
 _defined 41

in-between
 _defined 41

ink and paint
 _defined 41

interpolation
 _defined 41

jump cut
 _defined 41

key pose
 _defined 41

keyboard shortcuts
 _defined 42

keyframes
 _defined 42

layers
 _defined 42

layout
 _defined 42

layout and posing
 _defined 42

library
 _defined 42

light table
 _defined 42

line of action
 _defined 42

lip-sync

 _defined 42

 auto detection, defined 37

lip-sync detection

 manual, defined 42

low resolution

 _defined 42

manual lip-sync detection

 _defined 42

master palette

 _defined 43

model

 _defined 43

model sheet See character design

motion keyframe

 _defined 43

mouth

 chart 43

multiplane

 _defined 43

NTSC

 _defined 43

nudge

 _defined 43

onion skinng

 _defined 43

overlay

 _defined 43

PAL

 _defined 43

palettes

 _defined 43

 style, defined 43

panel
 _defined 44

panning
 _defined 44

paperless animation
 _defined 44

passing position
 _defined 44

path of action
 _defined 44

pegs
 _defined 44

pen
 _defined 46

phoneme
 _defined 44

pivot
 _defined 44

pixel
 _defined 44

pixelation See bitmap image

pose-to-pose
 _defined 44

rendering
 _defined 44

resolution
 _defined 44

RGB
 _defined 44

rigging
 _defined 44

rotary table
 _defined 45

rotoscoping
 _defined 45

roughs
 _defined 45

safe area
 _defined 45

scenes
 _defined 45

script
 _defined 45

scrubbing
 _defined 45

sequences
 _defined 45

shot
 _defined 45

slow-in
 _defined 45

slow_out
 _defined 45

slugging
 _defined 45

sound
 scrubbing
 _defined 45

stop-motion keyframe
 _defined 45

storyboard
 _defined 45

straight-ahead
 _defined 45

strokes
 _defined 46

tablet
 _defined 46

templates
 _defined 46

thumbnails
 _defined 46

timecode
 _defined 46

timeline
 _defined 46

trace and paint
 _defined 46

track breakdown
 _defined 46

tradigital
 _defined 44

traditional animation
 _defined 46

trajectory
 _defined 46

transition
 _defined 46

underlay
 _defined 46

vector
 _defined 46

velocity
 defined 39, 47

walk cycle
 _defined 47

workspace
 _defined 47

Xsheet view
 _defined 47

zone

 _defined 47