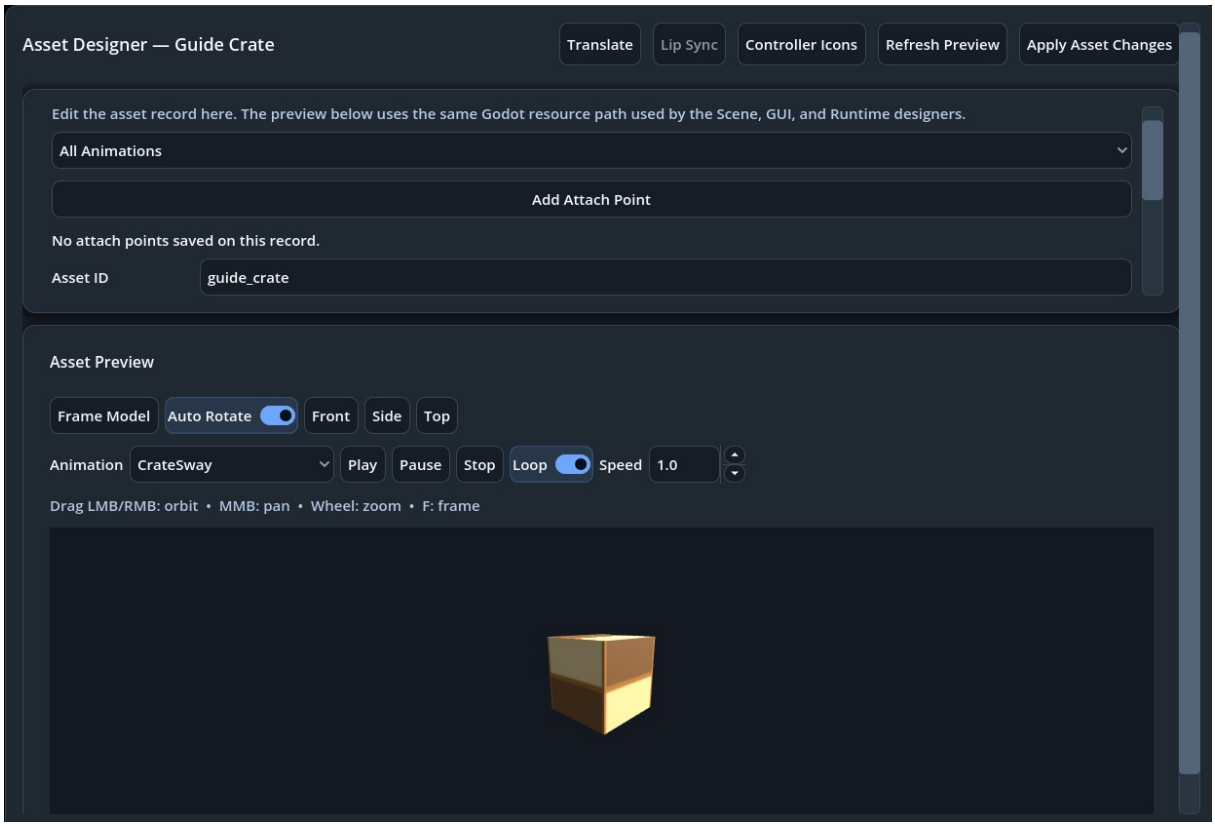
PCA 3D ADVENTURE STUDIO / 0.86.3.4

The Asset Importer

A practical guide to importing, checking and using your game assets



Step-by-step lessons | Real editor screenshots | Practice files

Windows / macOS / Linux

November 2026

How to use this guide

Keep the editor beside the book and try one small import at a time. Chapters 2-4 explain the common controls and a first image import. After that, choose the chapter for the type of file you need.

The practice archive is attached to this PDF and supplied as a separate ZIP. If your PDF viewer cannot open attachments, download the separate archive and extract it. The examples are source files for the lessons, not a standalone game.

Your next job	Read
Find a field or understand Apply and Save	Chapters 2-3.
Import a prop or character model	Chapters 5-14.
Animate 2D artwork or a Krita drawing	Chapters 15-18.
Bring in sound, voice, fonts or movies	Chapters 20-23.
Use shaders, data or controller icons	Chapters 24-26.
Replace media, translate it or troubleshoot	Chapters 27-30.
Look up a range, format or term	R1-R4.

Screenshots are cropped to the relevant part of the running editor. Enlarge a page to read a small control. The instructions cover the three desktop operating systems; file pickers and installation paths naturally vary.

Known limitations observed during the examples are described where they matter. In particular, Chapter 25 explains a Data/Other preview issue in this version and how to check the imported source.

Contents

How to use this guide 2

Contents 3

1. Before you import anything 5

2. Finding your way around 6

3. Identity, folders and saving 9

4. Your first image import 12

5. Importing a 3D model 14

6. The model preview, control by control 17

7. Replacing model textures 20

8. Attach points on models and images 25

9. Blender, FBX, Collada and OBJ 30

10. Source-model settings 33

11. Quake and Half-Life models 35

12. MagicaVoxel and Qubicle 36

13. LightWave and DirectX 39

14. Blockbench 42

15. Sprite sheets, from picture to frames 44

16. The 2D animation preview 49

17. Separate numbered images 50

18. Krita drawings and animation clips 53

19. Live2D Cubism bundles 56

20. Audio files 58

21. Lip sync for imported recordings 59

22. Fonts, modern and bitmap 63

23. Video 66

24. Godot and GLSL shaders 67

25. Data and Other files 69

26. Controller and keyboard icons 71

27. Language-specific asset replacements 74

28. Putting the imported asset to work 76

29. Reimporting and keeping projects organised 78

30. When something goes wrong 79

R1. File-format desk reference 81

R2. Settings and destination reference 82

R3. Short recipes to keep beside the editor 84

R4. Glossary and credits 85

1. Before you import anything

The Asset Designer is where a picture, model, recording or other file becomes a named part of your game. It lets you keep that file in the project, check it, and choose any import settings it needs. Other designers then refer to the asset by its ID.

This guide follows PCA 3D Adventure Studio 0.86.3.4. The instructions apply to Windows, macOS and Linux. The screenshots were taken in the running desktop editor with its dark theme. Your operating system's file picker, window borders and folder locations may look different. The controls inside PCA 3D have the same purpose.

You do not have to read the entire book before importing a picture. Start with Chapters 2-4, try the image lesson, then turn to the chapter for your next file type. The later reference tables are there for the day a particular file refuses to cooperate.

The practice files

The companion ZIP contains a small crate model, a numbered lamp animation, a Krita drawing with two clips, several source-model examples, a chime, a short video, fonts and two shaders. Extract the ZIP before browsing for a file. Do not try to import a file from inside the ZIP window.

Use a spare project while following the lessons. These files are deliberately small and simple. Their purpose is to make a setting easy to see, rather than bury it in a complicated character or a large room.

File or folder	What you can practise
Guide_Crate.glb	Model import, animation preview, materials and attach points.
Guide_Crate.gltf and its companion files	Importing a model that has files beside it. Keep the bundle together.
Guide_Crate.blend	Converting a Blender project into a reusable model asset. Requires Blender.
Guide_Crate_BaseColor.png and Guide_Crate_Roughness.png	Importing images, then choosing them as model textures.
Signal_Lamp_Sheet.png	Slicing a 4-column, 2-row sheet into seven frames.
Numbered-Frames	Importing Lamp_1, Lamp_2 and Lamp_10 in the correct order.
Signal_Lamp.kra	A saved still image, or separate Idle and Pulse paint-animation clips.
Source-Models	Small MD2, VOX, QB, LWO, X and Blockbench examples.
guide-chime.wav and Guide_Test_Clip.ogv	Audio and video playback. The chime is a test tone, not spoken dialogue.
GuideFont.ttf, GuideFont.agi and GuideFont.charset	Modern, Sierra AGI and SCUMM font examples.

File or folder	What you can practise
guide-tint.frag and guide-material.gdshader	A convertible 2D fragment shader and a native 3D Godot shader.
guide-data.csv, guide-data.json and guide-package.bin	Text data and a generic file with no specialised player.

Three things that are easy to confuse

An **asset file** is the picture, model or recording on disk. An **asset record** is the entry in your adventure database: its ID, name, type, imported path and settings. A **placed object** is something using that record in a room or GUI.

Importing a character model creates an asset record. It does not, by itself, create a playable character, assign walking animations or place the character in the room. Those are later jobs in the Character and Scene designers.

> Check your work. You should have the examples extracted into an ordinary folder and a separate working project open. Keep the original demo project available for comparison.

2. Finding your way around

Open the Asset Designer from the designer toolbar, or select an existing entry under **Assets** in the Adventure Database. The workspace selector above the designer lets you move between asset records. Its entries show the friendly name and the ID, for example **Guide Crate(guide_crate)**.

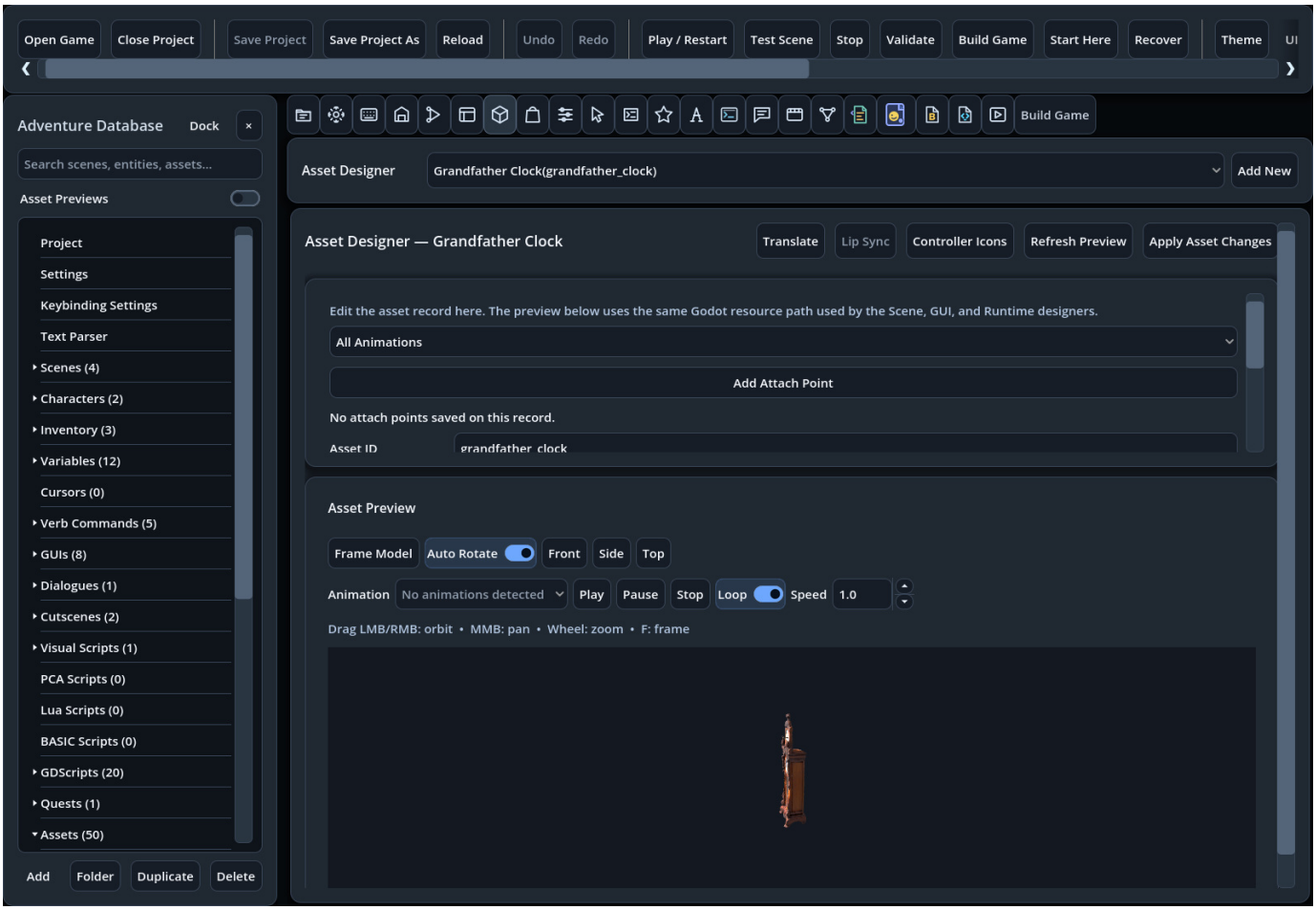


Figure 1. The Asset Designer with a supplied Clockmaker model selected. The upper area holds settings; the lower area previews the selected file.

The workspace controls

Use **Add New** beside the record selector to create another asset. Select the new record, give it a useful ID and name, and choose its Kind before browsing for the file. The database's Add control is another route to creating records.

The database search helps find existing records. **Folder** groups entries in the database. **Duplicate** makes another record based on the selection. A duplicated record can still refer to the same imported file; it is not automatically a separate physical copy. **Delete** removes the selected database record. Check where a record is used before deleting it.

The Asset Previews switch in the database controls thumbnails there. It is separate from the larger Asset Preview in the designer. A thumbnail is useful for finding a picture; the larger preview is where you inspect the file and play animation or sound.

The row of buttons inside the designer

Button	What it does	When you need it
Translate	Opens game translations for this asset.	Use another image, voice recording or compatible asset for a language.
Lip Sync	Opens recording-based mouth-cue editing. It is available for a saved Audio asset.	Prepare and check a voice recording's lip-sync track.
Controller Icons	Opens the built-in controller and keyboard icon browser.	Import a prompt icon as an Image asset.
Refresh Preview	Rebuilds the preview and checks model animation names again.	After a conversion finishes, an external file changes, or a preview appears stale.
Apply Asset Changes	Updates the asset record in the working adventure.	After changing identity, file selection, import settings, texture overrides or attach points.

On a narrow window, a toolbar may scroll sideways. If you cannot see Apply Asset Changes, widen the window or scroll that row. Do not assume a button has disappeared simply because it is off the right-hand edge.

Make room for the part you are using

Drag the divider between the form and Asset Preview. Move it downward while editing a long list of settings, or upward while examining a model. The form has its own vertical scrollbar. Scroll inside that area to reach fields lower down.

The preview has different controls for different kinds of asset. A model has a camera and animation player. An audio asset has playback and a seek bar. A data file shows readable text. A generic Other asset may have no visual player at all.

> Check your work. Select a supplied model, drag the divider, then select a supplied audio asset. Notice that the preview controls change with the asset kind.

3. Identity, folders and saving

Every import begins with a small set of common fields. Give them a little thought now; a clear database is much easier to work with once a game contains hundreds of files.

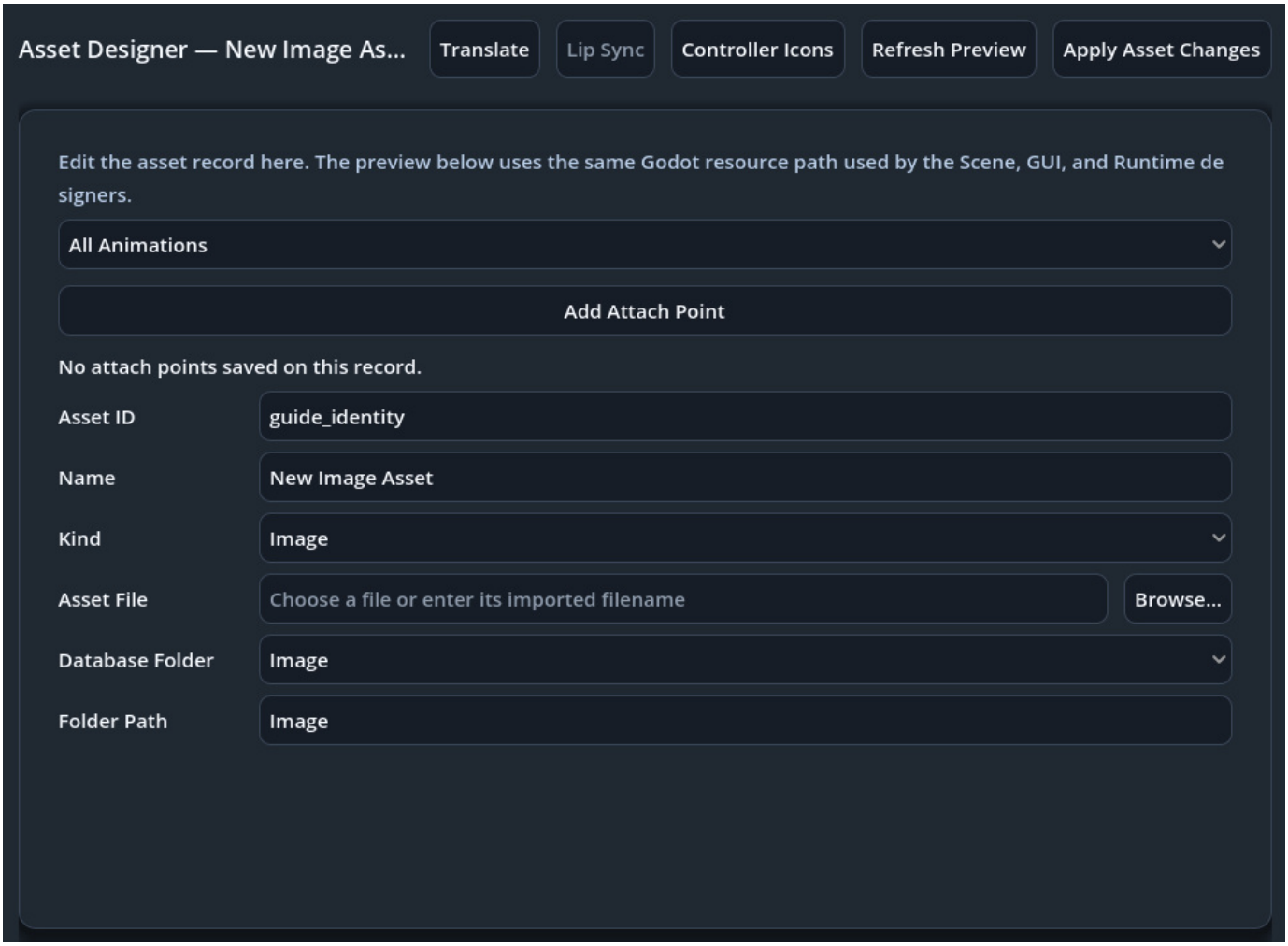


Figure 2. The common identity fields on a new Image asset. Browse selects the actual source file; the folder controls organise the database.

Field	Meaning	A sensible example
Asset ID	The stable reference other records and scripts use. It must not be blank.	guide_lamp_still
Name	The readable name in the editor. If left blank, the ID is used.	Signal Lamp Still
Kind	Tells the importer what sort of file and settings to expect.	image

Field	Meaning	A sensible example
Asset File	Shows the selected or imported file. Use Browse to choose or replace it.	Signal_Lamp_Still.png
Database Folder	Chooses a standard database category.	Image
Folder Path	Allows a nested database folder. It must not be blank when applying.	Image/Interface/Lamps

Keep IDs short, distinctive and stable: `ivo_voice_01`, `workshop_clock` or `hud_inventory_background`. Names can be friendlier: **Ivo - First Greeting**, **Workshop Clock**, **Inventory Background**. Avoid reusing an ID for an unrelated asset.

Choose **Kind first**. Changing it can put the record back into that kind's default database folder. Set your custom Folder Path after choosing the kind.

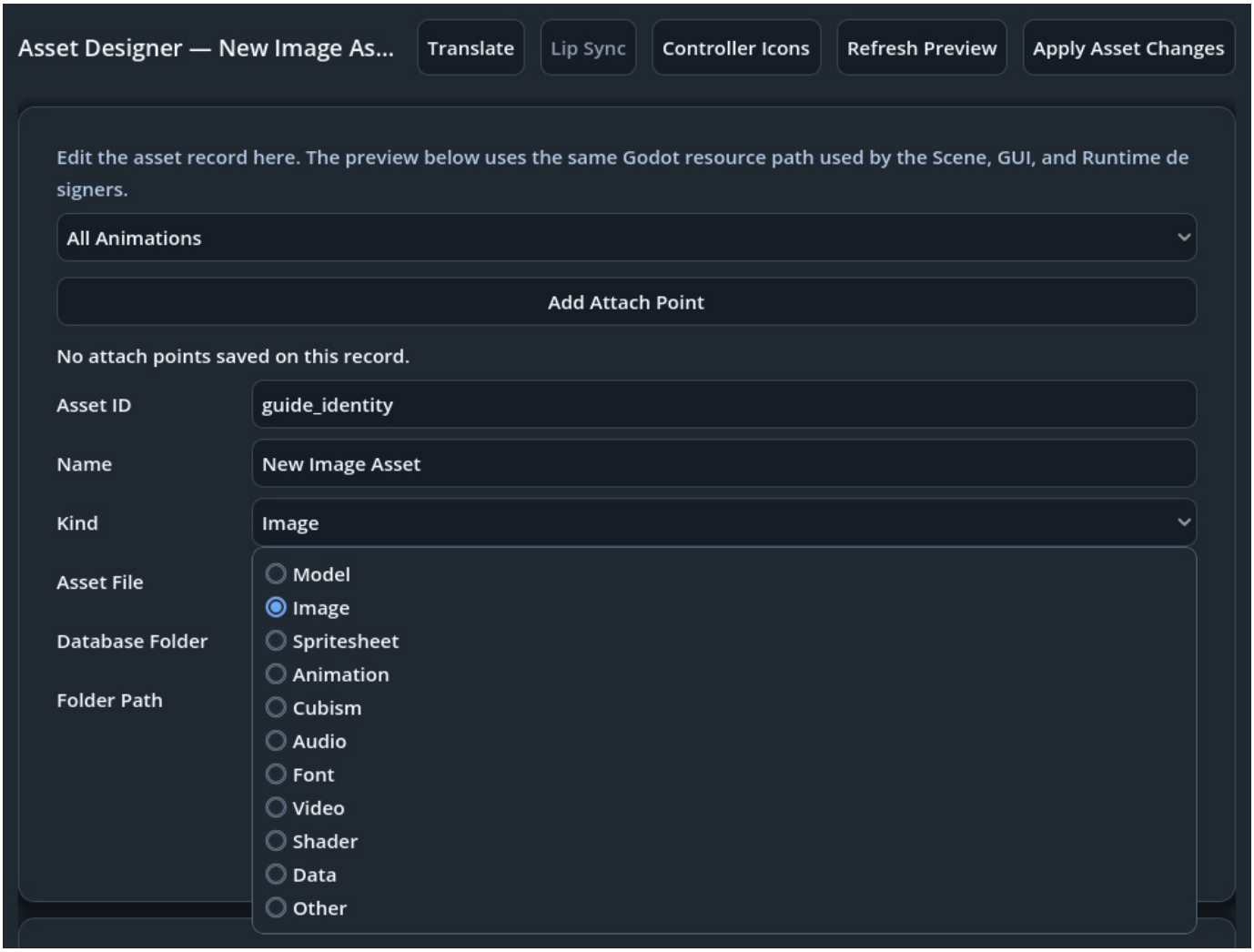


Figure 3. The Kind menu lists the 11 asset categories. Kind controls the import route; changing it does not magically convert the file.

Database folders are not disk folders

Folder Path changes where the record appears in the Adventure Database. It does not move the original picture or model into a matching folder on your computer. The importer manages its own copies and generated files.

Paths beginning with `res://` refer to the source package's resource root. They are not paths relative to whichever folder contains a loose `.pca3d` file. Keep the imported media with the source package when making backups or moving work to another computer. Saving the project document alone is not the same as collecting every original media file.

Apply, then save

1. Make the change in the Asset Designer.
2. Click **Apply Asset Changes**. This updates the record in the open working project.
3. Click **Save Project** on the main toolbar to write the adventure document.
4. If you replaced a widely used asset, check one of its users in a room or GUI before continuing.

Browsing can copy a file before you apply the record. Conversely, having a file copied does not mean your new settings are saved. Treat Apply and Save as two separate steps.

Do not replace a file by merely typing a different filename into Asset File. Browse performs the import work and records the resulting path. Renaming the visible text is not a substitute for it.

Using the Browse window

Browse opens a file-selection window, not another asset record. On some systems it is the operating system's usual picker; the screenshots use the editor-style picker. Navigate to an extracted source folder, select the file, and use Open to confirm it.

Part of the editor-style picker	What it is for
Back and Forward arrows	Retrace the folders visited in this picker.
Up arrow	Open the parent folder.
Path	Shows the current disk folder; enter a known folder path to navigate directly.
Favourites and star	Keep convenient folder shortcuts. These are not imported assets.
Recent	Return to a recently used location.
File list or thumbnail grid	Select the source. Double-click a folder to open it.
View, visibility and sorting controls	Change how files are listed, including applicable hidden-file visibility and sorting. They do not change the imported media.
Filename field	Shows or accepts the source filename in the current folder.

Part of the editor-style picker	What it is for
File-type filter	Restricts the displayed extensions for the selected Kind. It does not convert an unsupported file.
Open	Confirms the selection and begins the relevant import route.
Cancel or close	Leaves the picker without selecting a replacement.

For a numbered-frame Animation, select the whole frame set before Open. For a model bundle, select its main file and keep its companions in the source folder; do not select unrelated texture files as extra models. For Krita Animation, select one KRA rather than mixing it with frame pictures.

Preview changes and saved changes

Action	What it changes
Orbit, pan, zoom, Front, Side, Top or Frame Model	The inspection camera only.
Model preview Speed or Loop	The current audition of the clip. Configure gameplay animation separately.
Lower animation-preview FPS	The preview player's timing.
Upper Animation FPS, Unit Scale or Forward Axis	Import settings; apply and save them.
Replace Model Textures, channel choices or mapping	Asset appearance overrides; apply and save them.
Attach point name, position, rotation or contact radius	Asset attachment data; apply and save it.
Save Track in Lip Sync	Updates the recording's mouth-cue data; save the project afterwards.

4. Your first image import

This is the shortest complete import lesson. Once you have done it, most of the other categories follow the same basic pattern.

1. In the Asset Designer, click **Add New**.
2. Set Asset ID to `guide_lamp_still` and Name to **Signal Lamp Still**.
3. Choose **image** in Kind.
4. Click **Browse...**, open the extracted examples folder, and choose `Signal_Lamp_Still.png`.
5. Wait for the preview. If an import is still being prepared, allow it to finish and click Refresh Preview once.

6. Set Folder Path to `Image/Practice` if you want a practice folder.
7. Click Apply Asset Changes, then Save Project.

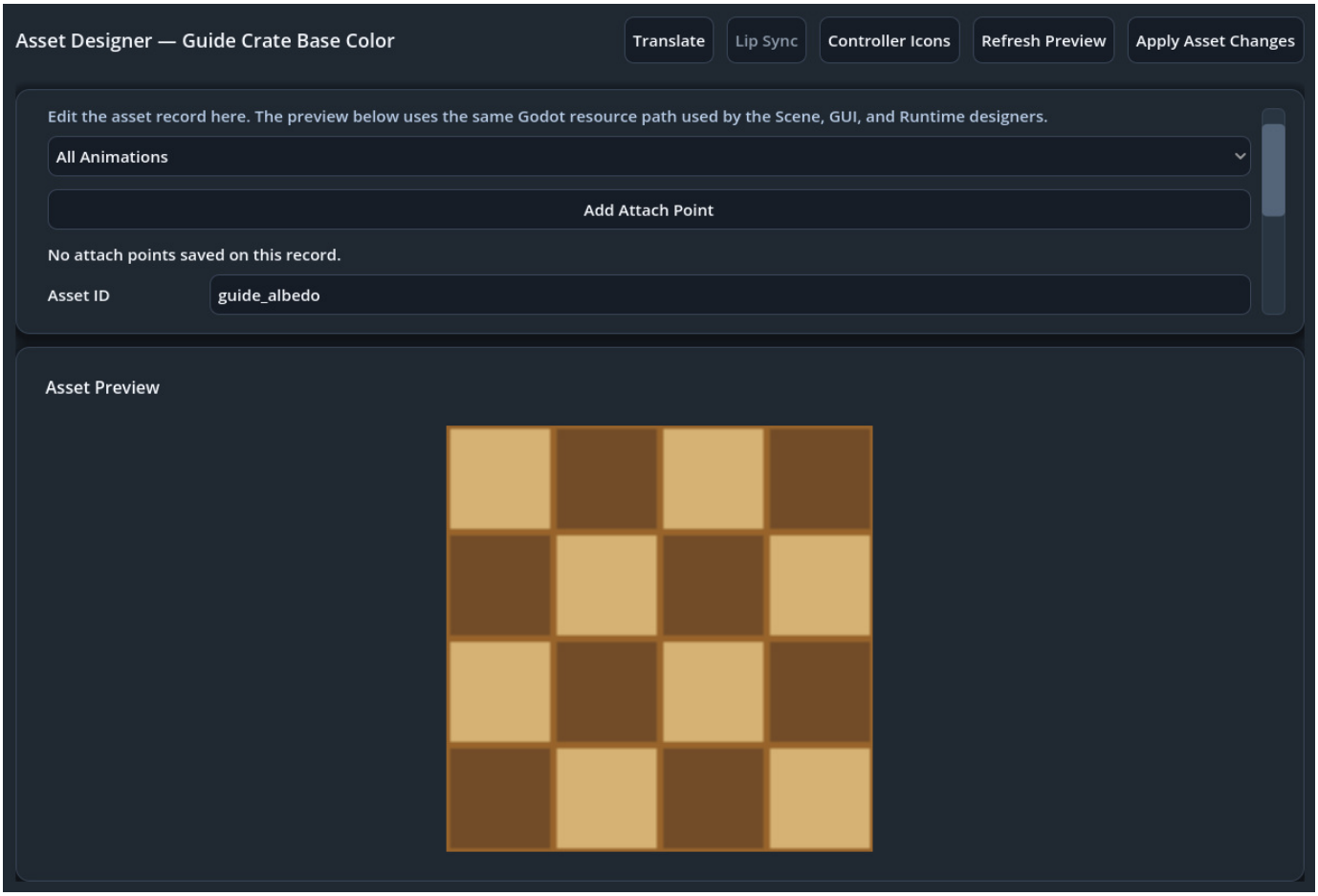


Figure 4. An imported image shown in the editor. This example is the crate's base-colour texture; the same steps import the lamp still.

What Image accepts

The picker offers PNG, JPG/JPEG, WebP, SVG and Krita `.kra` files. PNG is a useful choice when a GUI image needs transparency. A JPEG does not acquire transparency simply because the editor displays it on a dark background.

Image preview fits the picture into the available area. It does not crop the source or decide how a GUI will stretch it. A button's background fit, hover image and pressed image are set on the button in the GUI Designer. An inventory item and a room background also need their own assignments.

An imported Image can be reused as a GUI picture, an icon or a model texture where a compatible asset selector is provided. The import does not assign it to all of those places automatically.

Replacing an image

Select the existing record and Browse for the new picture. Apply and save. Keeping the same ID lets existing users continue to refer to the record, but the new picture may have a different aspect ratio or frame layout. Check those users after the replacement.

Choose distinct source filenames. Ordinary imports can use a destination based on the source basename, so two unrelated files both called `background.png` can collide. Names such as `workshop_background.png` and `harbour_background.png` are safer.

> Check your work. The image should still be listed after you select another asset and return. Then try selecting it in an Image field in the GUI Designer.

5. Importing a 3D model

For a first model import, use the supplied `Guide_Crate.glb`. It has a simple material and a short animation, making both easy to inspect.

1. Add an asset with ID `guide_crate` and Name **Guide Crate**.
2. Choose Kind **model**.
3. Browse for `Guide_Crate.glb`.
4. Wait for the import and preview to finish. Avoid starting another import while the current one is still busy.
5. Click Frame Model if the object is not comfortably in view.
6. Inspect the crate and its animation as described in Chapter 6.
7. Apply Asset Changes and Save Project.

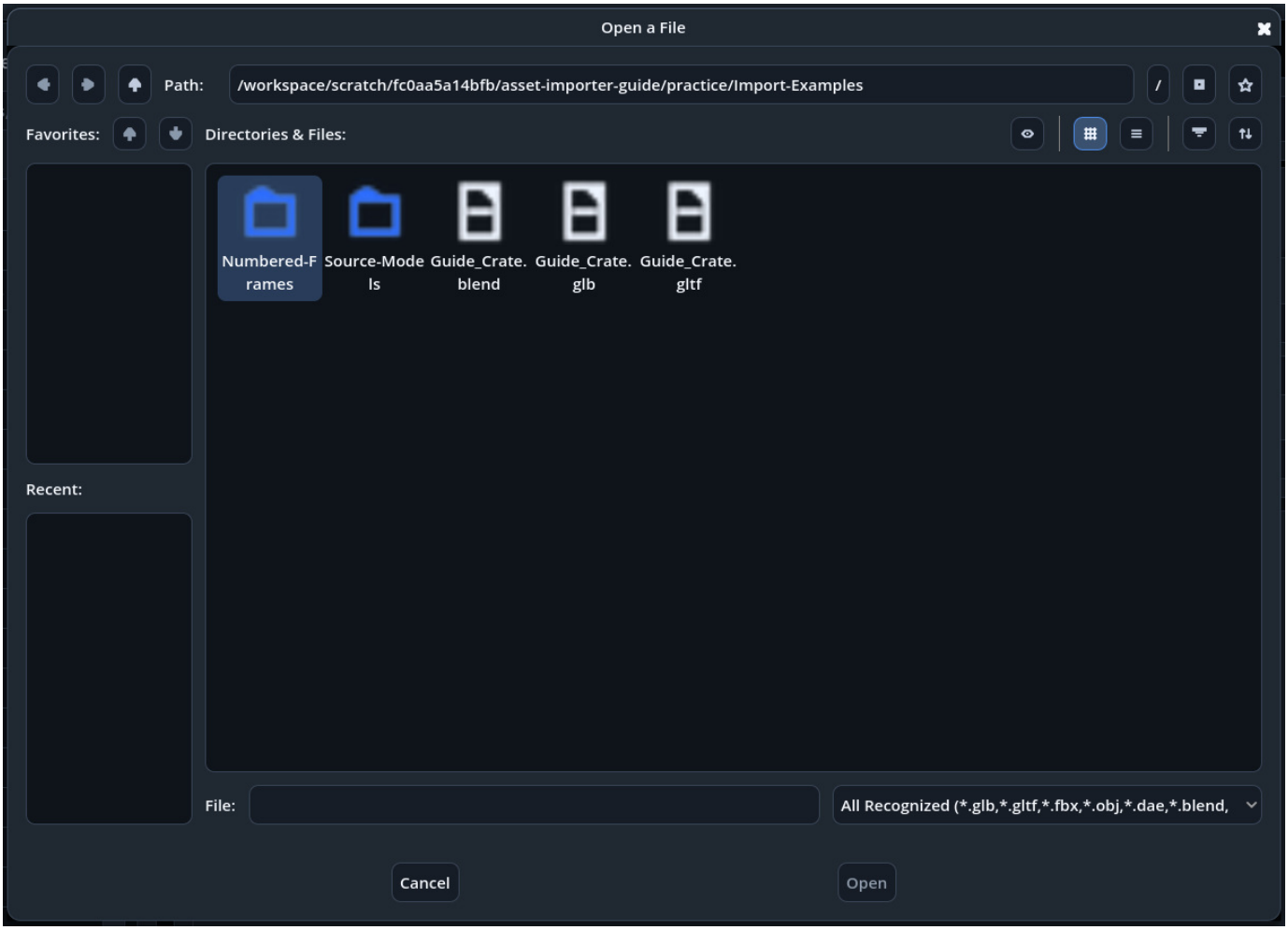


Figure 5. Browsing for a model. The exact file-picker appearance depends on the operating system and whether a native picker is being used.

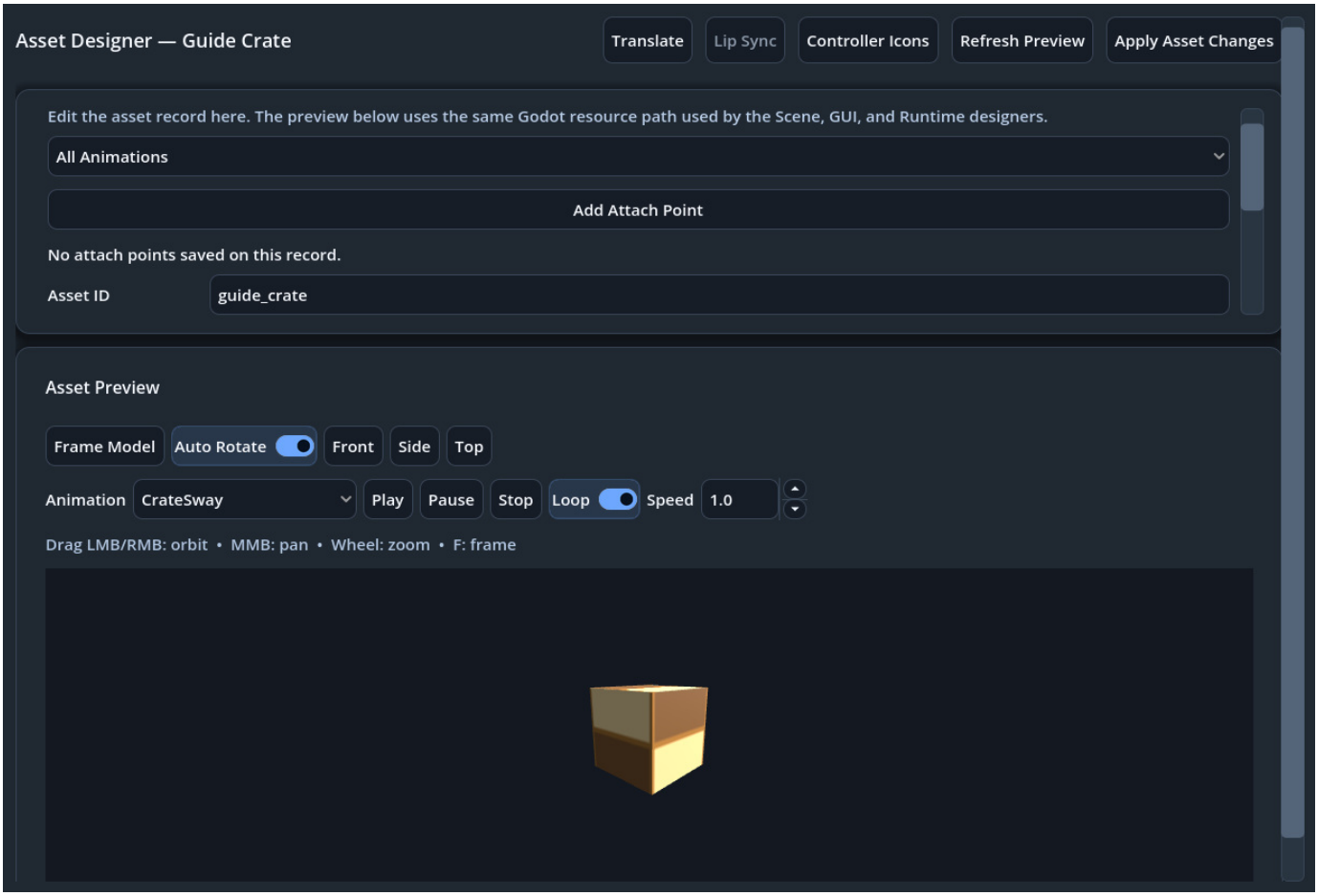


Figure 6. The practice crate after importing. A successful model preview is the first check, before assigning the asset to a room or character.

GLB and glTF

A GLB can contain meshes, materials, textures, a skeleton and animation in one file. That makes it a convenient handoff format. A `.gltf` often refers to a `.bin` and separate texture images. Those companions are part of the model, even though you select only the `.gltf` in the picker.

Keep a glTF bundle's relative folder structure intact before importing. Do not move the JSON file away from its textures or rename its `.bin` without updating the source. The importer attempts to collect dependencies, but it cannot recover a file that was never supplied.

The practice folder contains both versions of the crate. Import the GLB first. Then, if you want to try the bundle route, create a second record and choose the glTF with all of its companion files still beside it.

What a successful import proves

It proves that this model can be loaded and previewed with the settings in front of you. It does not prove that every animation is mapped to a gameplay state, every texture was collected, every character mouth target is configured or every exported platform has been tested.

A model with no animation can still be a perfectly good door or piece of furniture. A model with animation still needs a Character definition if you want it to be a playable actor.

When you see an import message

The source editor may need time to prepare a model or image for use. Read the status line rather than repeatedly clicking Browse. After the work finishes, try Refresh Preview. If the model still fails, check the file and its dependencies before changing unrelated character settings.

Keep the source in a reasonably short folder path. Deeply nested downloads and long filenames can exceed the importer's path limits. Moving a complete source bundle into a short temporary import folder is often an effective first test.

6. The model preview, control by control

The model preview is an inspection booth. You can look underneath a prop, check a rig's clips and place an attach point without moving the actual game object.

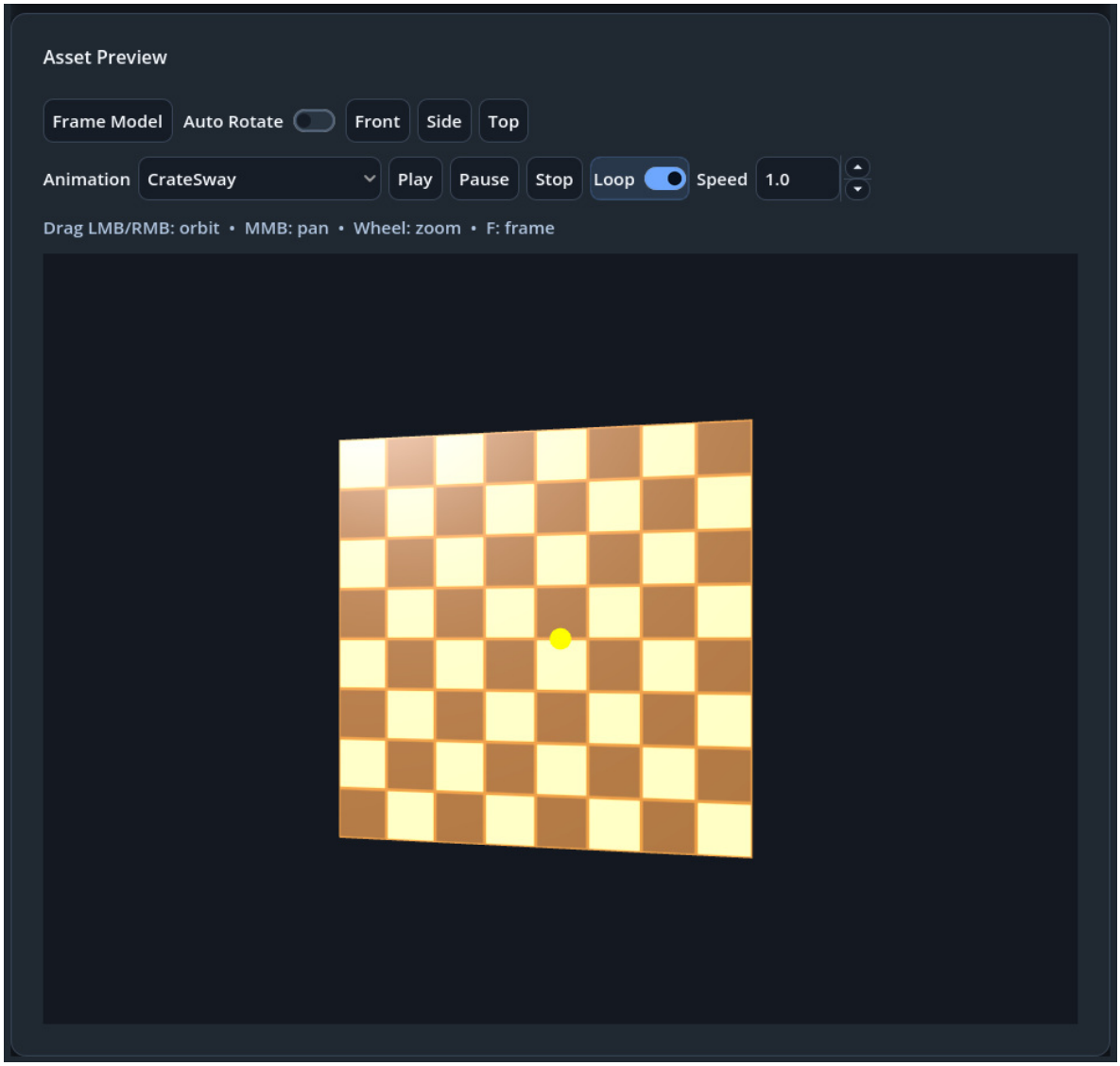


Figure 7. Front view of the crate with automatic rotation switched off. The preview controls do not change the object's placement in a game room.

Control or gesture	Result
Frame Model, or F while the preview has focus	Fits the model into the preview and resets the inspection view.
Auto Rotate	Turns the preview model automatically. Switch it off for careful inspection.
Front / Side / Top	Changes to a standard inspection view.
Left-drag or right-drag	Orbits around the model. Manual camera movement switches off automatic rotation.
Middle-drag	Pans the inspection view.
Mouse wheel	Moves the preview camera closer or farther away.
Animation menu	Chooses an imported animation clip. The RESET clip is omitted.

Control or gesture	Result
Play / Pause / Stop	Controls the selected clip.
Loop	Repeats the selected clip during preview.
Speed	Auditions a clip from 0.05 to 8 times normal speed. The default is 1.

On a touch device or a laptop without a middle mouse button, the named view and Frame Model buttons remain useful ways to inspect the asset. Do not mistake the lack of a particular mouse button for a missing model setting.

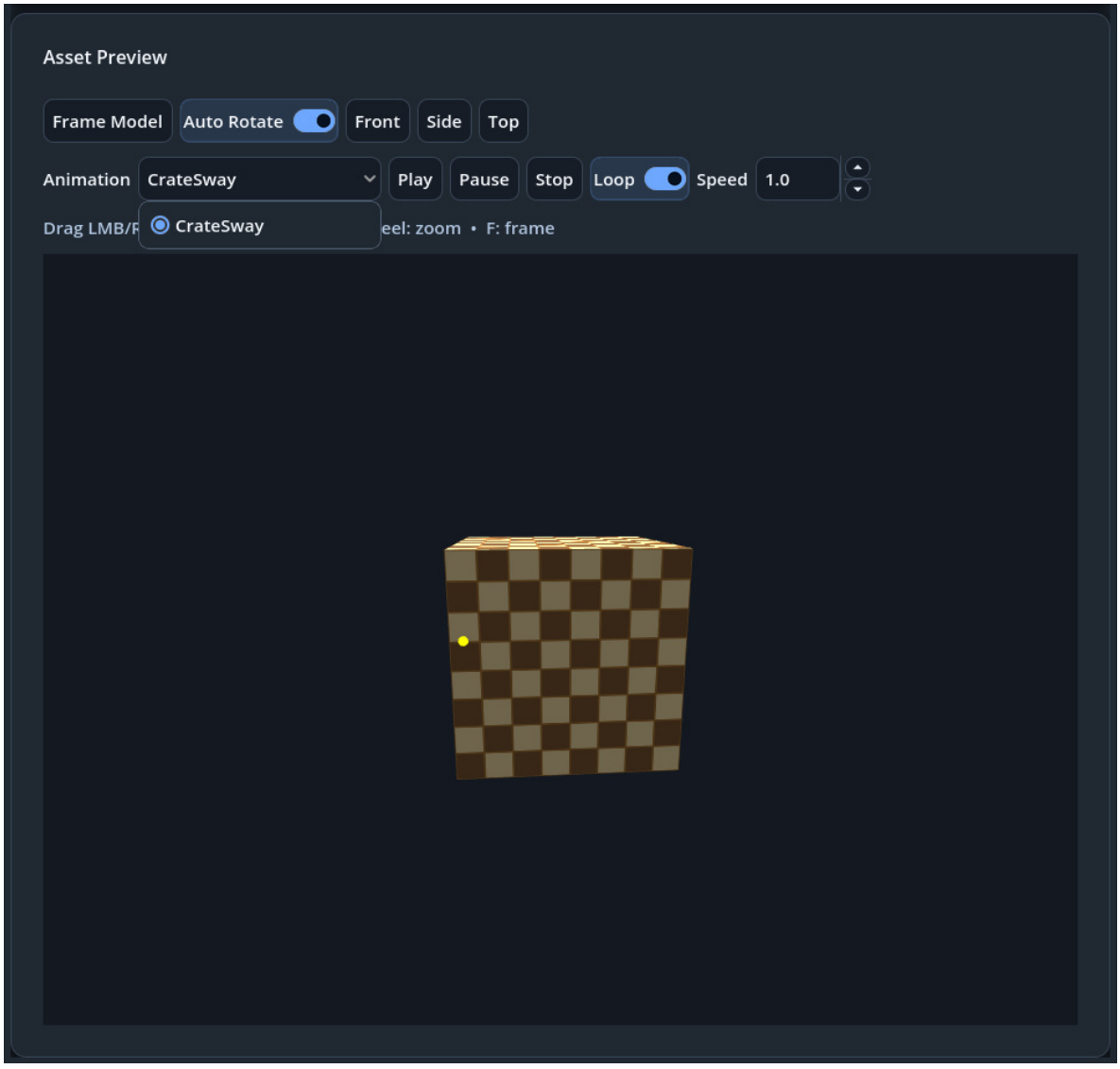


Figure 8. The model's animation menu. The preview finds clips on the first AnimationPlayer it discovers in the imported model.

Checking animation properly

Choose each clip in turn. Watch the feet, hands and mesh, rather than checking only that the clip name appears. Look for sliding, a mesh left at the origin, a sudden scale change, missing textures and an unexpected turn at the loop boundary.

The preview normally begins with the first usable clip. That is an inspection convenience, not a decision about which clip the character should use while standing. Configure idle, walk, talk and other states in the Character Designer.

If the menu says no animations were detected, the file may be static, the exporter may have omitted its actions, or the model may contain animation in a structure the preview does not combine. This preview uses the first AnimationPlayer it finds; it does not merge every player in a complicated scene.

Changing **Speed** here is different from changing a source-model **Animation FPS** in the upper form. Speed is for auditioning. Source-model FPS is an applied import setting.

> Check your work. Play the crate's CrateSway clip, pause it, choose Front, then press F. The model should remain an imported asset with the same ID throughout.

7. Replacing model textures

You can replace standard model texture channels without editing the original model file. Start by importing the replacement pictures as **Image** assets. They then appear in the model's texture selectors.

For the practice crate, import `Guide_Crate_BaseColor.png` as `guide_albedo` and `Guide_Crate_Roughness.png` as `guide_roughness`. Apply both Image records before returning to `guide_crate`.

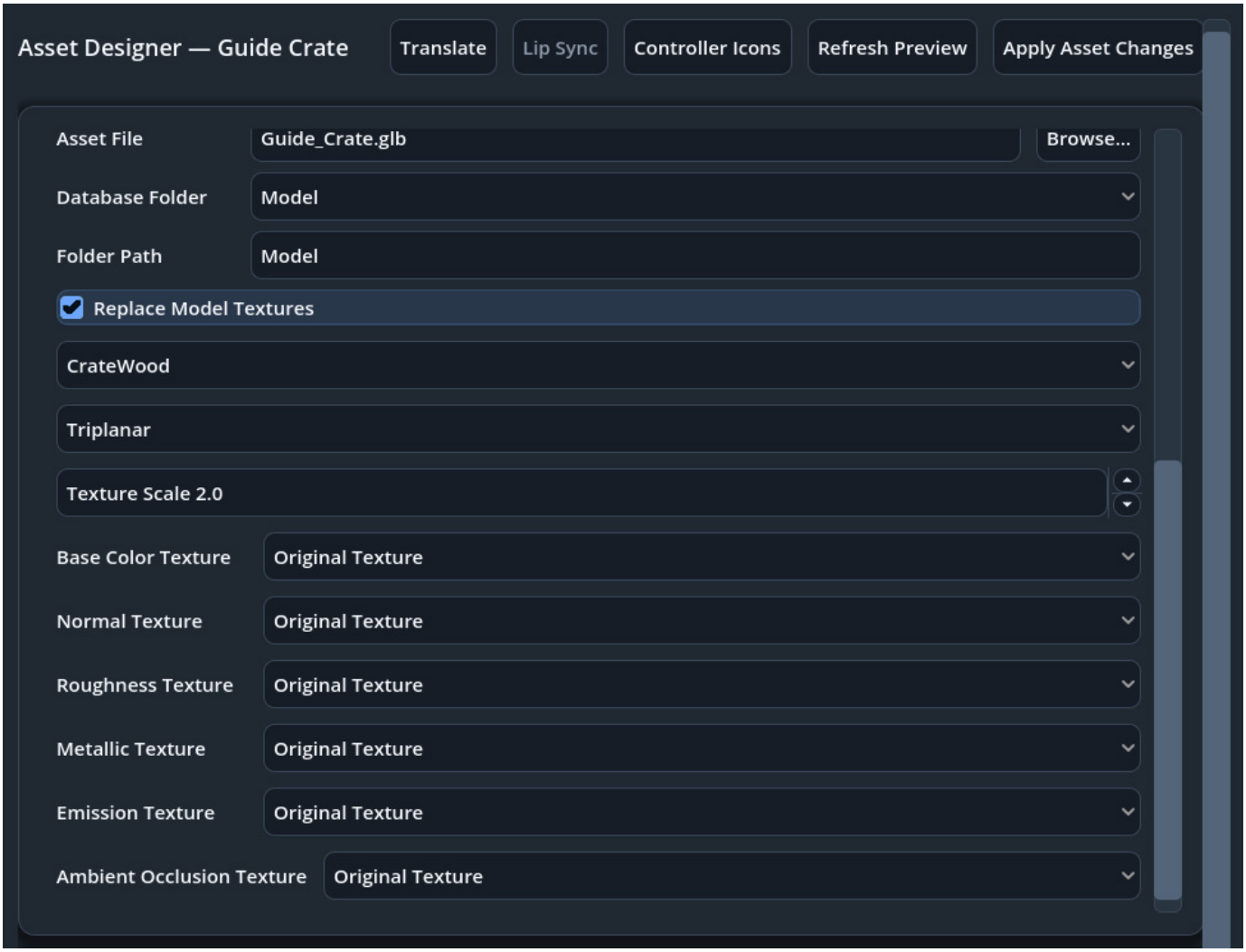


Figure 9. Replace Model Textures reveals the material selector and texture channels. Import replacement pictures as Image assets first.

1. Select the model asset.
2. Turn on **Replace Model Textures**.
3. Select the material to change. On the crate it is the crate material.
4. Choose the base-colour image in **Base Color Texture**.
5. If wanted, choose the grey roughness image in **Roughness Texture**.
6. Leave the other channels on **Original Texture** for this first test.
7. Apply Asset Changes, refresh the preview if needed, and save the project.

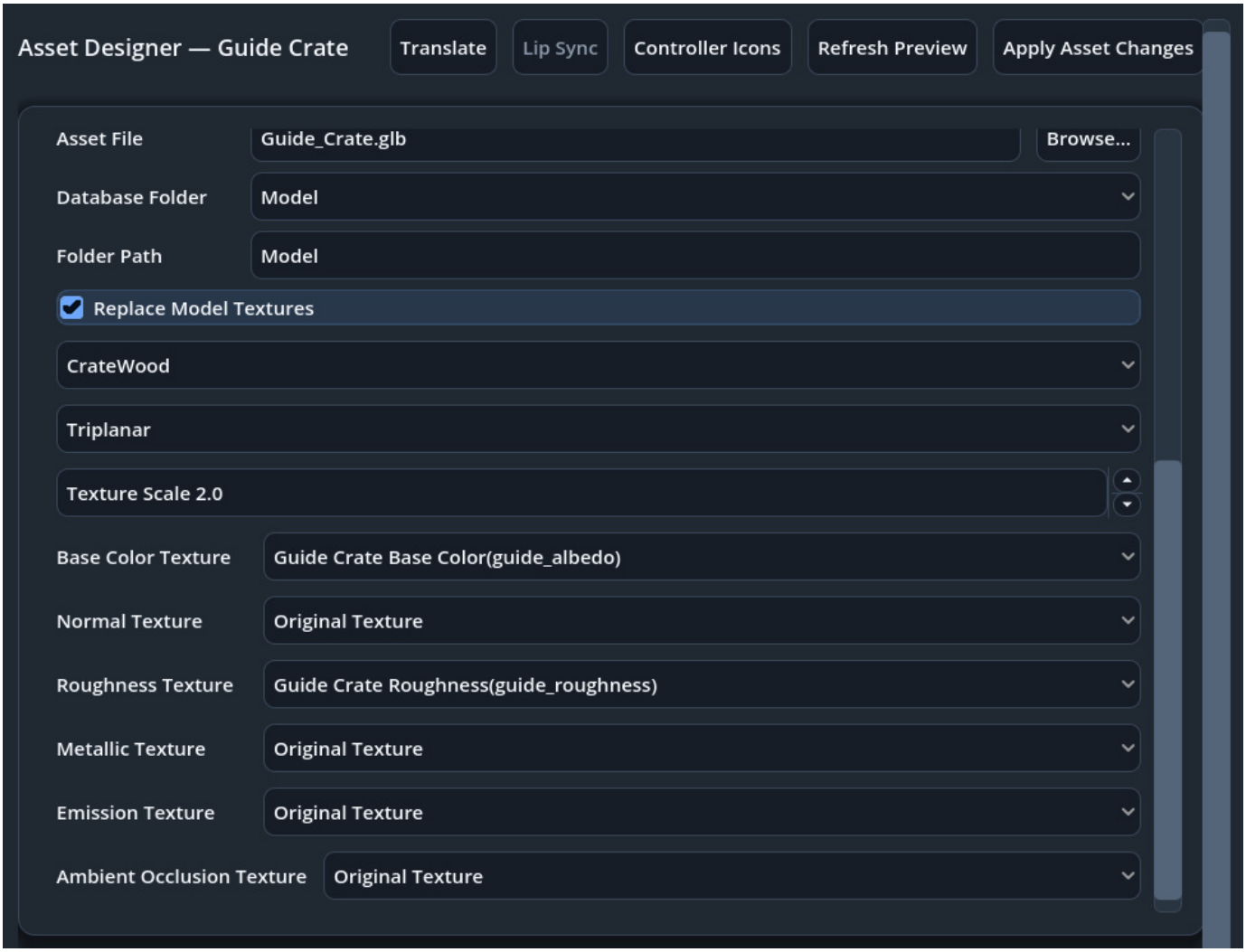


Figure 10. The crate's texture assignments. Each channel can be replaced separately, or returned to Original Texture.

What the six channels mean

Channel	What the image contributes	Common mistake
Base Color Texture	The surface's visible colour or painted pattern.	Supplying a lighting-baked picture when the material is also being lit.
Normal Texture	Fine surface direction, creating the appearance of small bumps.	Using an ordinary colour photograph as a normal map.
Roughness Texture	Where reflections look sharp or spread out on a standard material.	Expecting the map to create a complete mirror camera by itself.
Metallic Texture	Which parts behave like metal in the material's lighting model.	Painting everything metallic without a suitable environment.
Emission Texture	The parts of the surface that appear self-lit.	Expecting the texture alone to illuminate the entire room.
Ambient Occlusion Texture	Small creases and contact shading already represented in the map.	Using it as a replacement for all scene lighting.

Select **Original Texture** to remove that channel's replacement. It means return to the model's original channel, not paint the surface black or remove all of its material.

The material selector matters. A model can contain several materials. Replacing one does not mean all of the others were changed. Surfaces sharing the same material name share that override. A custom ShaderMaterial keeps its own texture setup rather than being treated like a standard material.

Mapping and Texture Scale

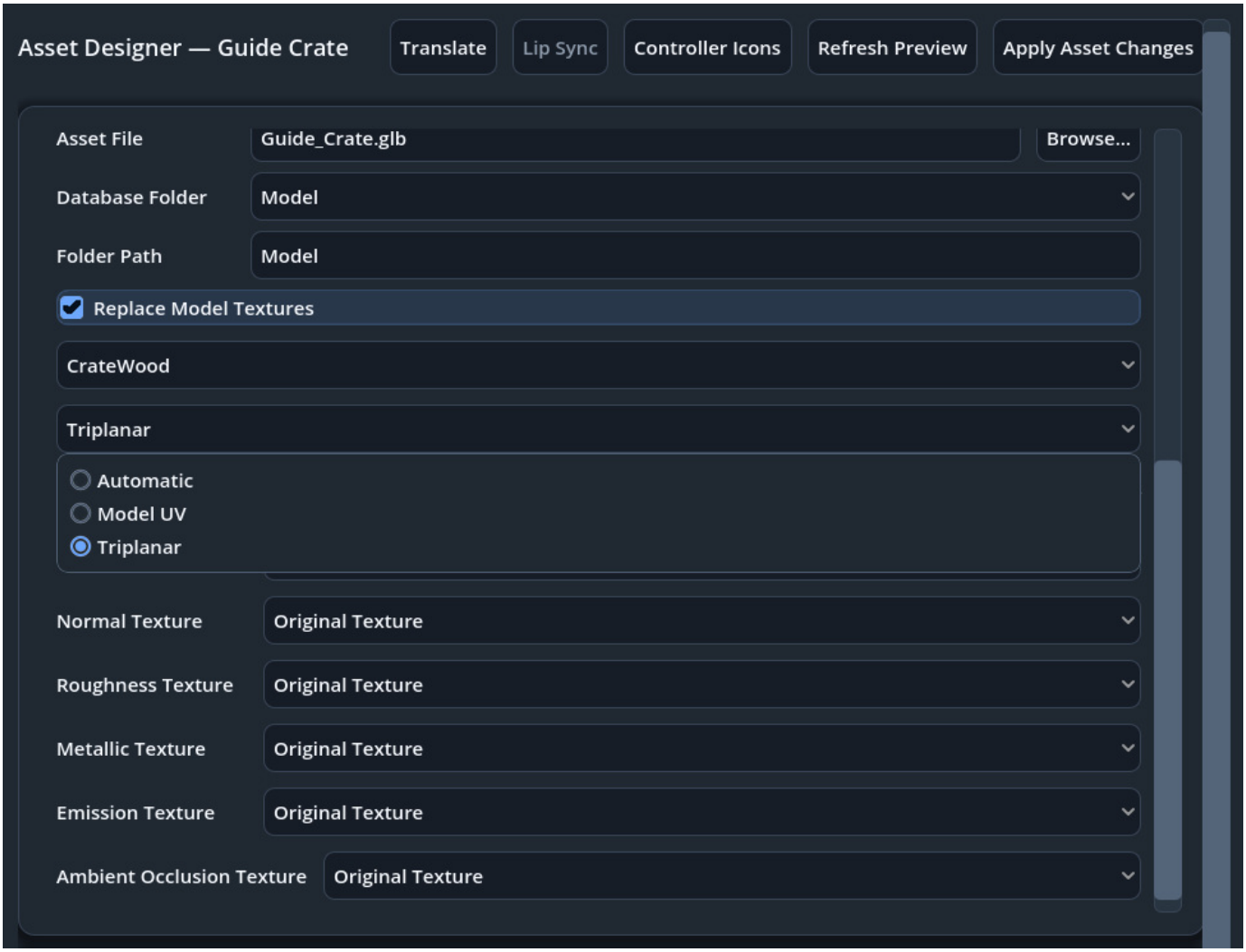


Figure 11. The mapping menu offers Automatic, Model UV and Triplanar. Texture Scale controls repetition, not object size.

Mapping	Use it when...	What to watch for
Automatic	You want valid original UVs to be used, with projection as a fallback when UVs are missing.	Check the result rather than assuming every imported mesh has good UVs.
Model UV	The model was unwrapped for its own texture or atlas.	A poorly unwrapped model remains poorly unwrapped.
Triplanar	You want a tiled surface projected from three directions.	Lettering, faces and carefully painted atlases usually need original UVs instead.

Texture Scale ranges from 0.01 to 100. At 1 the texture uses its normal repetition; at 2 it repeats twice as often. This does not make the crate twice as large. Triplanar repetition is based on local model units, so scale and mapping should be judged on the actual object too.

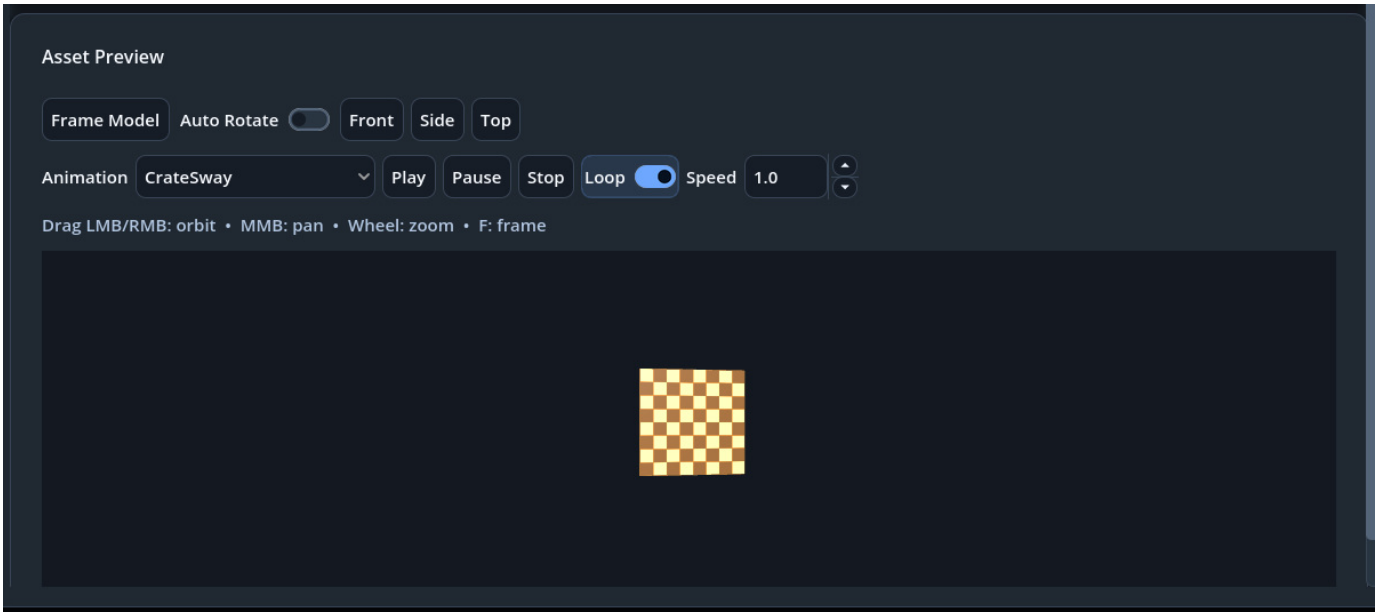


Figure 12. The crate preview after applying replacement textures. The source model file has not been rewritten.

Turn off Replace Model Textures to restore the original rendering while retaining the override configuration. Apply and save that change. This makes it easy to compare your replacement against the original.

> Check your work. Change only the base-colour channel first. Once that works, add the roughness image. If you change six channels at once, it becomes harder to tell which one caused an unwanted result.

8. Attach points on models and images

An attach point is a named location on an asset: a hand grip, a hat position, a muzzle or a point where two actors should meet. It gives later attachment or contact actions a reliable target. Creating a point does not attach anything by itself.

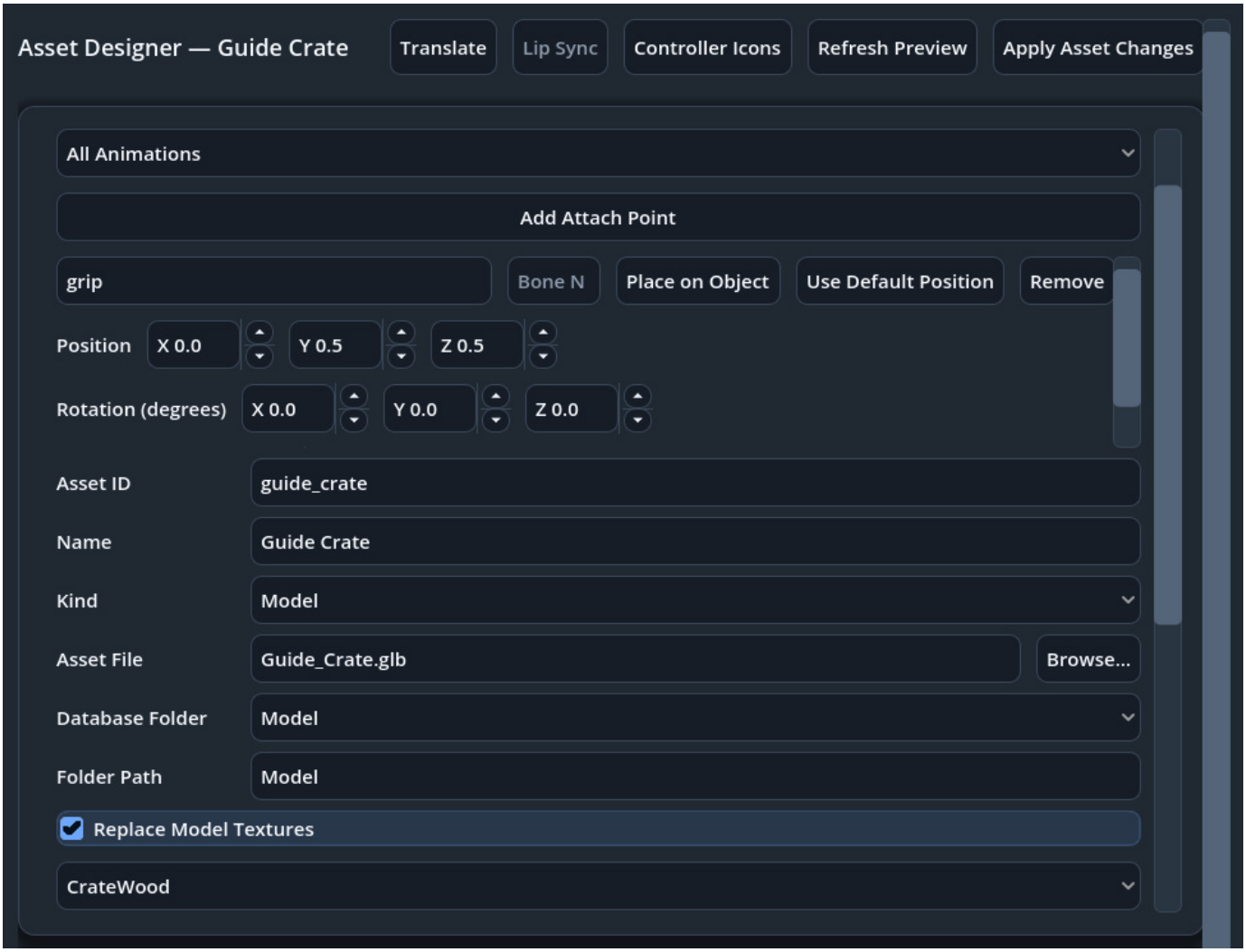


Figure 13. A named grip point on the crate. Position, rotation, scope and contact radius describe the point; later actions decide what uses it.

A first model point

1. Select the model and choose a useful preview animation, if it has one.
2. In the attach-point area, select **All Animations** for a general point, or choose a particular clip for an animation-specific adjustment.
3. Click **Add Attach Point** and give the point a clear name such as `grip`.
4. Click the model in the preview to place it. Placement pauses the preview animation and automatic rotation.
5. Use **Place on Object** if you want to pick the position again. Press Escape to cancel a placement operation.
6. Adjust Position X, Y and Z, then Rotation in degrees if needed.
7. Set Contact Radius and Enabled as appropriate.
8. Apply Asset Changes and Save Project.

The model preview can be made larger by dragging the form divider upward. Use Front or Side to check whether the point sits on the intended surface rather than somewhere behind it.

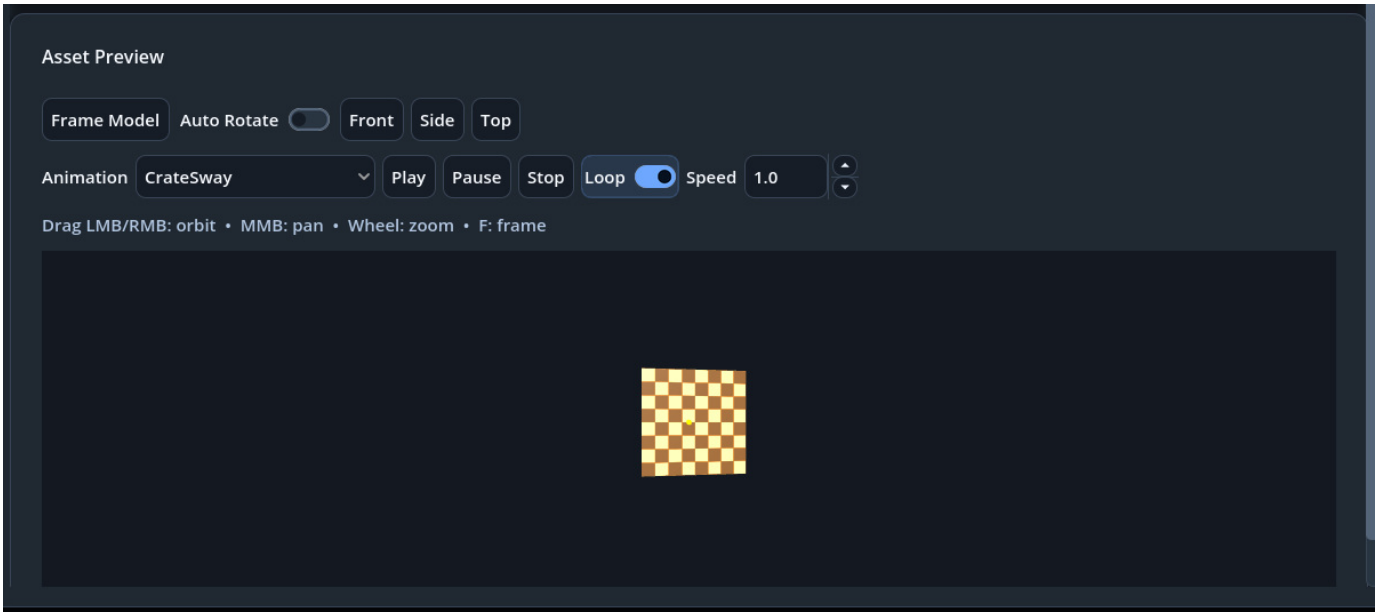


Figure 14. The model preview after adding the grip point. A point belongs to the asset's local space, not to a fixed position in a room.

Every point field

Field or control	Meaning
Name	The name later attachment or contact actions look for. Duplicate point names are given a suffix.
Bone Name	Optional bone association on a rigged model. It must match an actual bone.
Position X / Y / Z	The point's local position. A bone-linked point uses its bone's local space.
Rotation X / Y / Z	Local orientation in degrees. Useful when a held object faces the wrong way.
Contact Radius	The distance used by point-contact logic: game units for 3D, native pixels for a 2D point. It is not a collision-box size.
Enabled	Allows the point to be used without deleting its settings.
Place on Object	Starts another placement pick in the preview.
Use Default Position	Clears the current animation's position override so the default can be used. It does not delete the whole point.
Remove	Removes that point from the edited asset data when applied.

Do not use Contact Radius to fix an object's walking obstacle. Attach-point contact and the object's collider or navigation blocking are different systems.

Animation-specific positions

A point can have a general position and a position adjusted for a particular animation. This is useful if a grip needs a slight correction during one pose. The selected preview clip also sets the animation scope for editing these positions. Check the scope before moving the point.

An imported skeleton can move a correctly associated bone point with the hand. A free model-space point does not acquire a hand association simply because you gave it the name hand.

Points on images

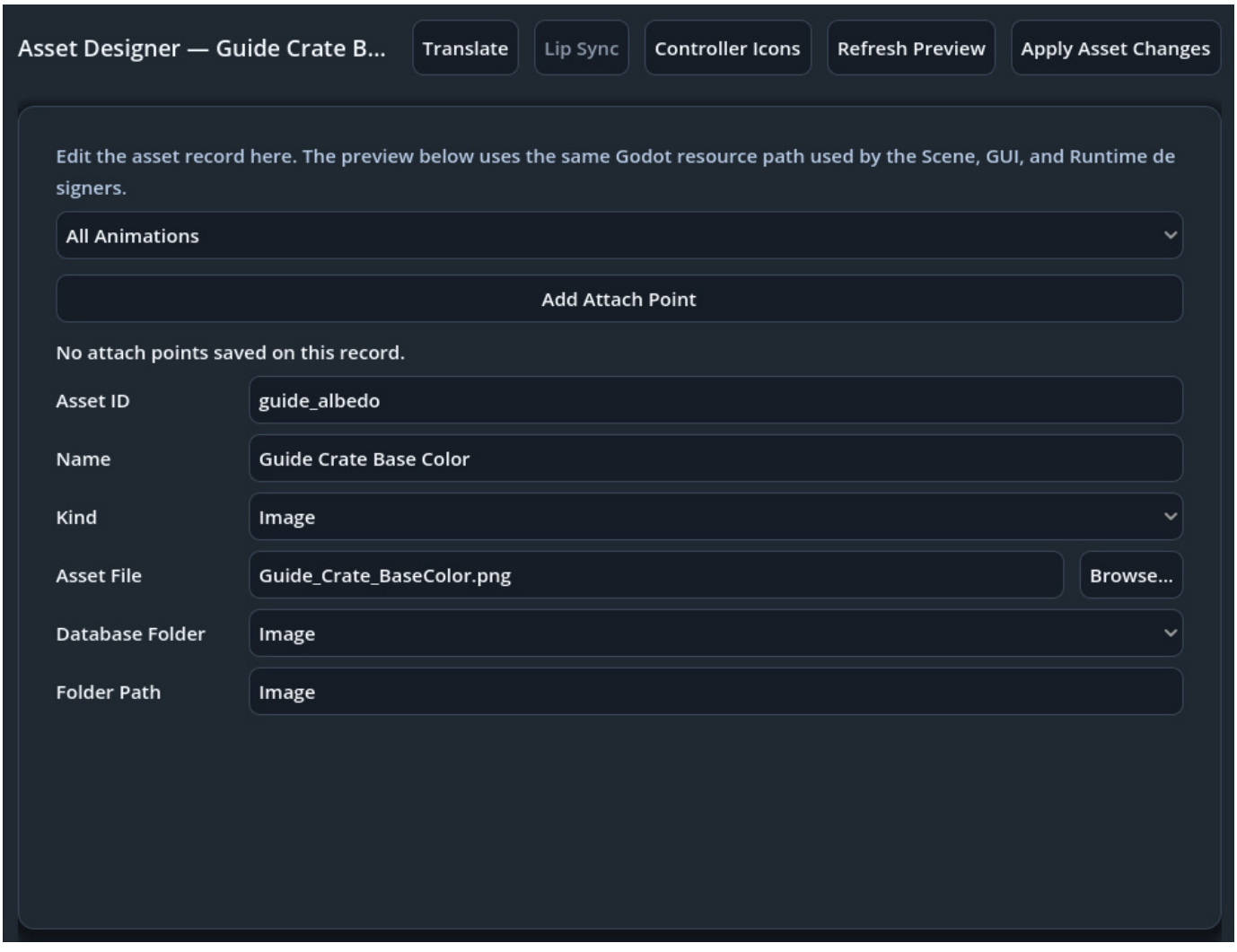


Figure 15. An attach point on an Image asset. Image positions use normalised image coordinates rather than room coordinates.

For an image, placement records a normalised position: roughly 0 at the left or top edge and 1 at the right or bottom edge. The centre is about X 0.5, Y 0.5. That keeps the point relative to the picture. Its contact radius still uses the 2D contact system's pixel units.

Points inherited from characters and scenes

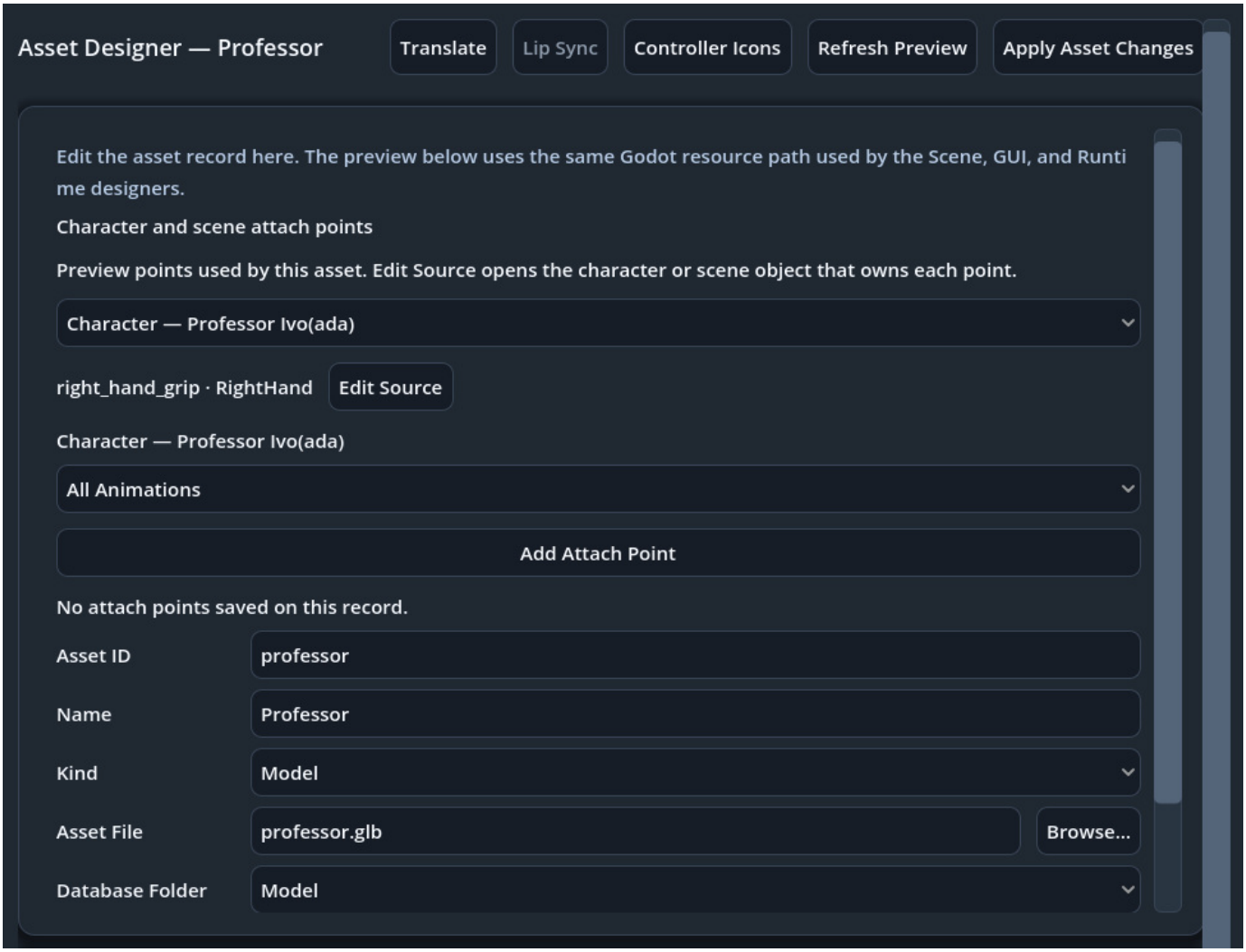


Figure 16. A supplied character model shows the source browser for points defined on characters or scene instances using that asset.

The source browser lets you inspect points belonging to Character definitions and scene instances which use this shared model. **Asset points only** shows the asset's own points. **Edit Source** opens the actual owner of a selected inherited definition.

A character or scene can refine an asset's shared point. Think from general to specific: asset, character, then scene instance. If an adjustment appears to have no effect, check whether a more specific definition is taking precedence.

9. Blender, FBX, Collada and OBJ

The model picker accepts several authoring formats. They do not all travel through the same route or preserve exactly the same features. Use the preview to check the result, and keep the editable source outside your release build.

Blender .blend

When you choose a `.blend`, PCA 3D starts Blender in the background and exports a self-contained GLB. The converted GLB is what the model asset uses afterwards. Players do not need Blender installed just to run that converted model.

1. Install a suitable Blender version, or have a portable installation available.
2. Create a Model asset and Browse for `Guide_Crate.blend`.
3. Read the conversion status and wait. Starting Blender and exporting a scene takes longer than loading a small GLB.
4. Check the converted model, its material and CrateSway clip.
5. Apply Asset Changes and save.

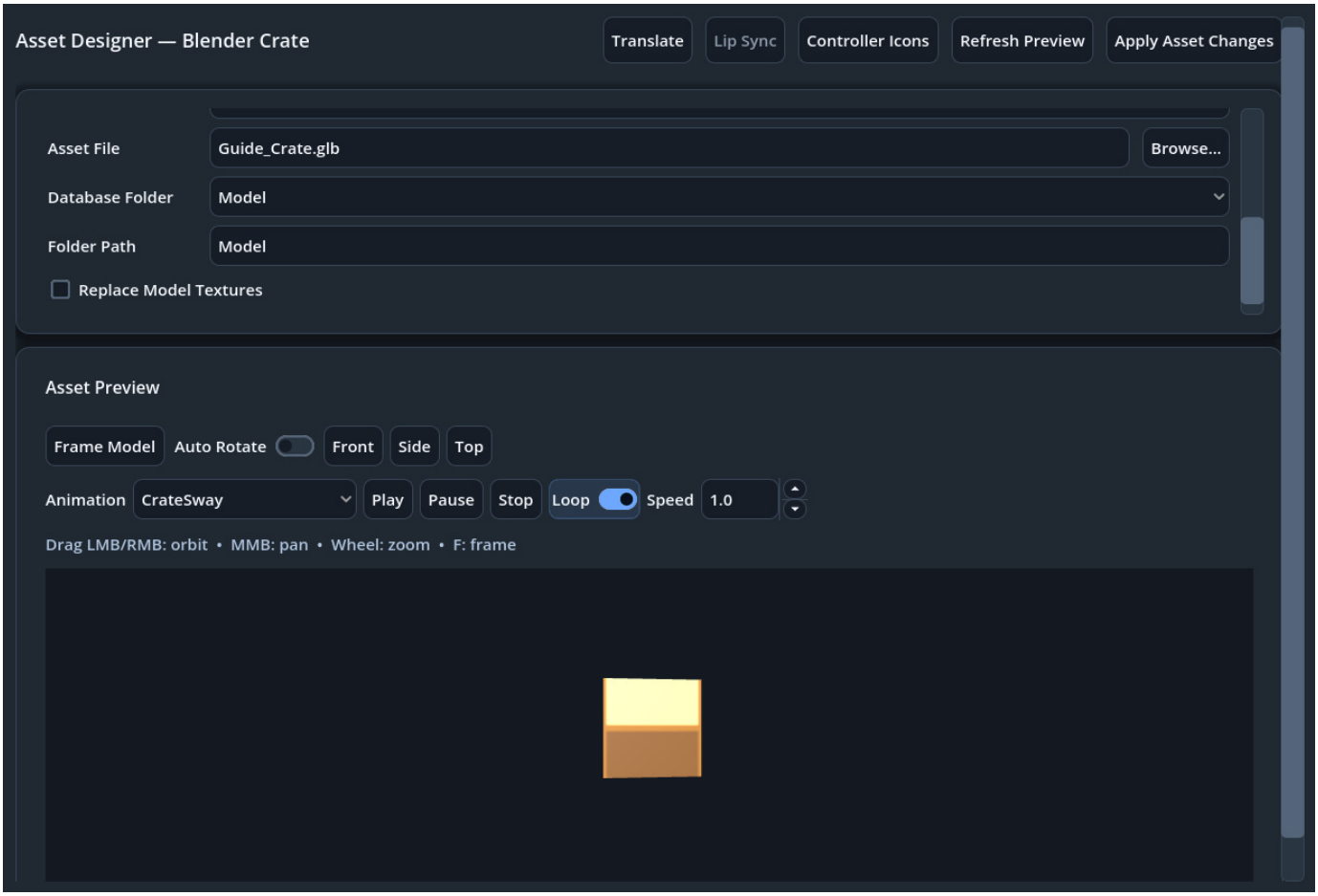


Figure 17. The Blender practice file after conversion to GLB. The resulting model can be used through the normal asset workflow.

The bridge searches common Blender installation locations. If it cannot find Blender, its error message describes the executable setting. For a custom installation, the tool setting is `pca3d/tools/blender_path`; the environment overrides are `PCA3D_BLENDER_PATH` and `BLENDER_PATH`. These must identify the real executable. This is a tool-location setting, not an Asset Designer field.

Materials, armatures, shape keys and exported animation can travel through GLB, but a procedural Blender material is not guaranteed to look identical. Bake or simplify effects that depend on Blender-only nodes. Keep external textures and linked source resources available during conversion.

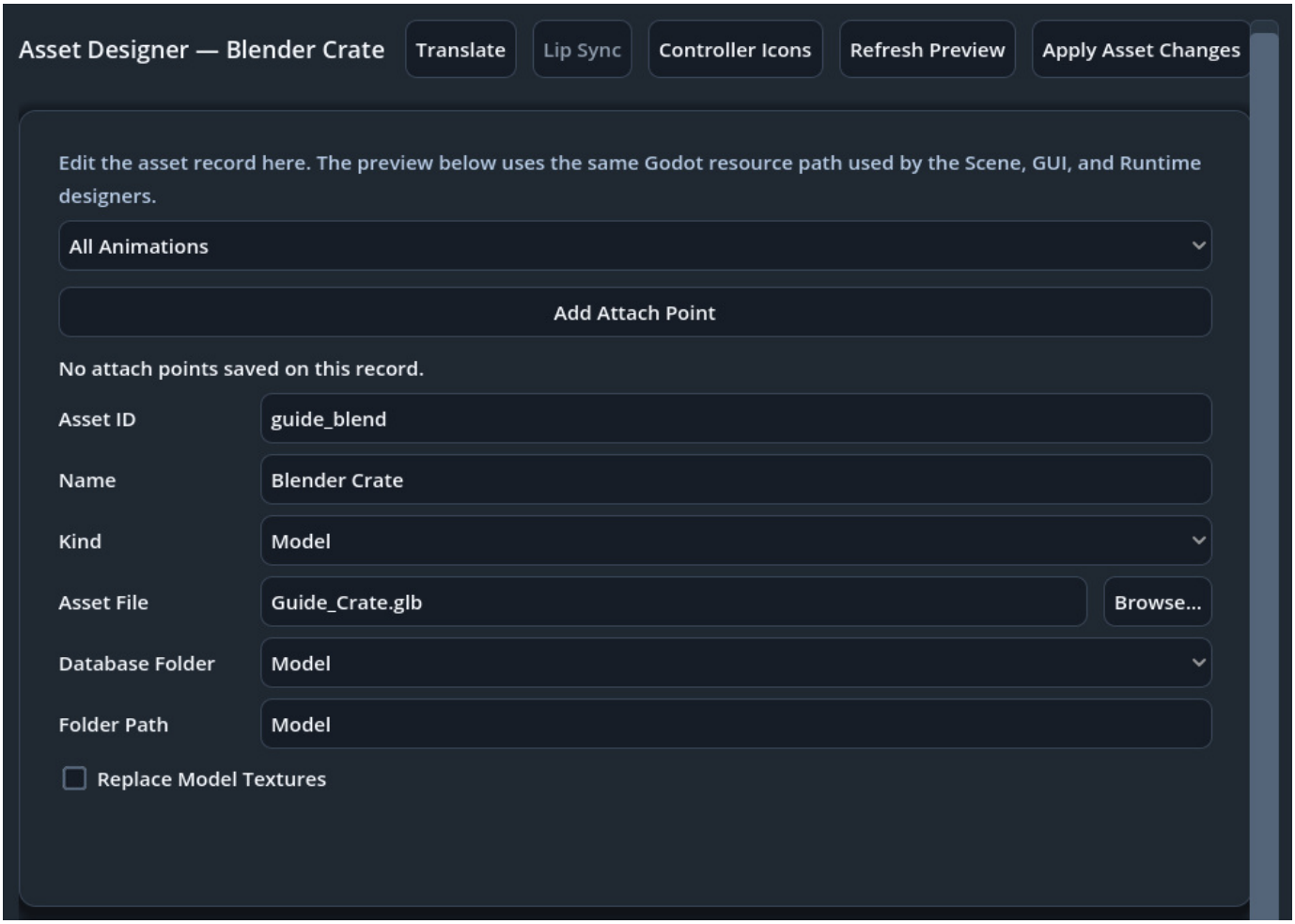


Figure 18. The Blender asset's identity after conversion. The imported runtime file is the generated GLB, rather than the editable .blend project.

FBX and Collada .dae

Import these as Model assets with their texture dependencies available. PCA 3D attempts to gather the files and repair compatible texture references. Check every clip and material after importing; an FBX filename alone does not prove that its animation or texture package is complete.

If a difficult source keeps losing materials, exporting a well-prepared GLB from the authoring application is a useful comparison. First determine whether the problem is in the source export or in its dependency paths.

OBJ

The current OBJ route reads the geometry and creates a Godot Mesh resource. It is a **single-surface** result, with a fallback albedo texture when one is found. Treat it as a static mesh import. Do not expect it to preserve a rig, animation or a complicated multi-material layout as if it were a full animated scene.

For a multi-material animated character, choose a format that carries that information, such as GLB. Changing the asset Kind cannot add a skeleton to an OBJ.

10. Source-model settings

MD2, MDL, MD3, VOX, QB, LWO, X and Blockbench files have their own import controls. The heading changes for the selected format, but the main ideas are shared.

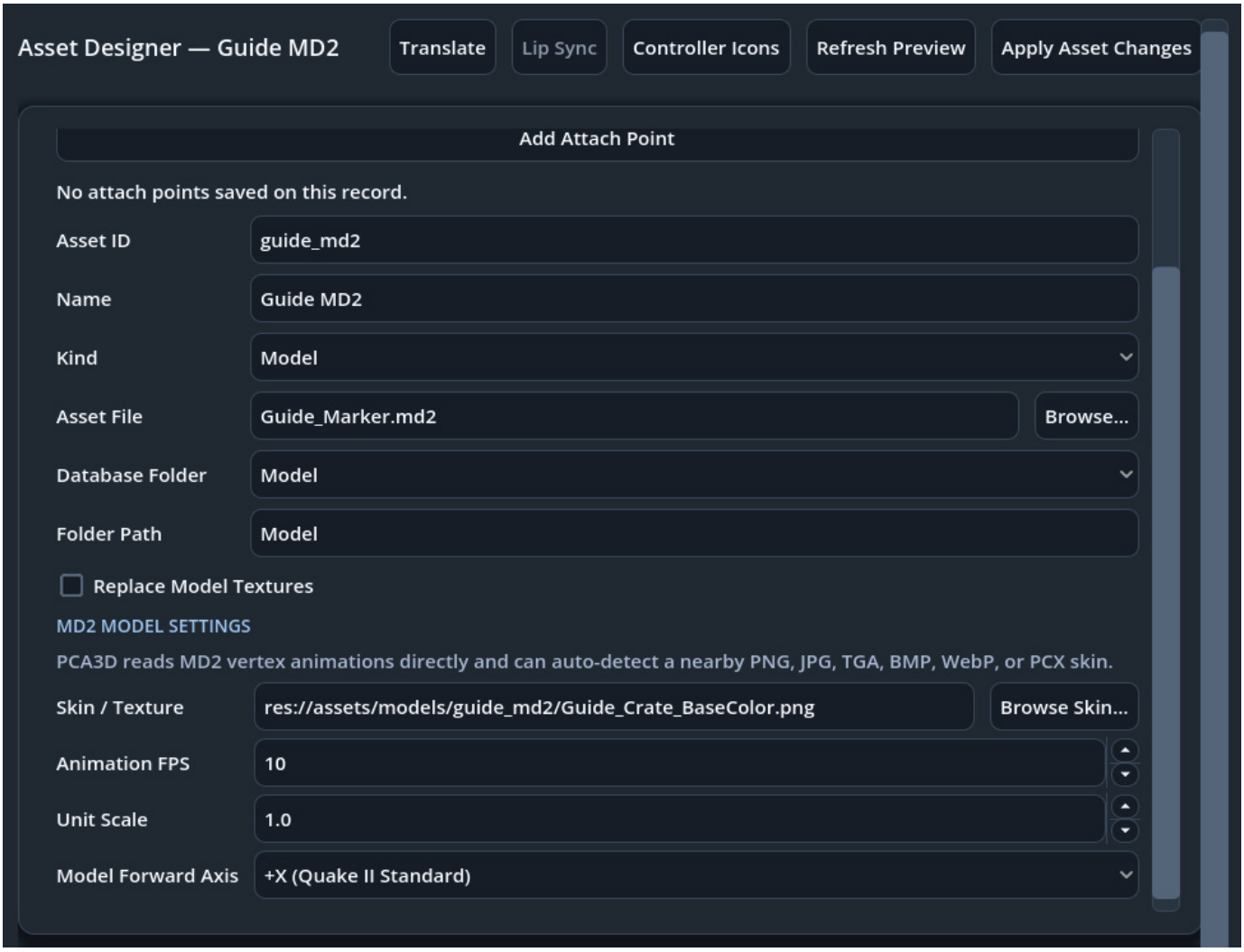


Figure 19. The MD2 settings area. Skin selection is specific to MD2; the other source-model formats expose the controls that apply to them.

Setting	What it changes	How to choose a value
Skin / Texture and Browse Skin	The MD2 model's skin image.	Choose a matching skin when automatic discovery did not find the right one.
Animation FPS	Fallback or sampling rate for supported source animation. Range 1-120.	Start with the source's intended rate; this cannot create missing animation.

Setting	What it changes	How to choose a value
Unit Scale	Converts source units into game units. Range 0.0001-100 in the editor.	Compare with a known object or actor in a room.
Model Forward Axis	Tells the importer which direction the source considers forward.	Use the source format's expected axis, then check the character's facing.

Unit Scale can be deceptive in preview

Frame Model fits the object into the inspection area. A tiny imported model and a huge one can therefore look much the same size on screen. Judge scale beside another object in the Scene Designer, not only by how much of the asset preview it fills.

Unit Scale changes the source conversion. A scene instance's transform scale is a later placement adjustment. Keep those jobs separate so that you do not fix one character by accident and make all of its shared props wrong.

Forward Axis

The menu offers **+X (Quake II Standard)**, **-X**, **+Z (PCA3D Standard)** and **-Z**. Quake and GoldSrc-style sources commonly begin with +X. Voxel, LightWave, DirectX and Blockbench imports generally begin with +Z in this workflow.

Forward Axis corrects how the imported model faces. It is not a replacement for moving a camera or rotating a room. If a walking actor travels one way while looking sideways, check this setting as well as the character's animation and visual rotation.

Useful starting scales

These are the importer's usual defaults, not rules for every model ever made in the format. Existing records can retain their own saved values.

Source	Usual Unit Scale	Timing note
MD2, supported MDL and MD3	0.025	Quake-style vertex animation commonly starts at 10 FPS; embedded sequence timing may apply.
VOX and QB	0.1	VOX can contain timeline frames; QB is static.
LWO	1	Animation FPS is disabled for this static model route.
X	1	File tick-rate information takes precedence when available; the usual fallback is 25 FPS.
BBModel	0.0625	Sixteen Blockbench units become one game unit; animation sampling commonly starts at 30 FPS.

Apply and save after changing these fields. Use Refresh Preview if needed, then inspect a room instance as well.

11. Quake and Half-Life models

These formats are useful for older game-style assets. They also have strict format versions and companion-file conventions. A matching extension is only the beginning of the check.

MD2

MD2 stores vertex-frame animation. The importer groups suitable frame names into clips. Browse for `Source-Models/Guide_Marker.md2`, turn off Auto Rotate, then choose its idle clip.

If the skin is missing or wrong, use **Browse Skin...**. The MD2 skin path supports familiar image formats and PCX skins. Automatic discovery checks nearby matching images, but a folder containing several unrelated skins may need a manual choice.

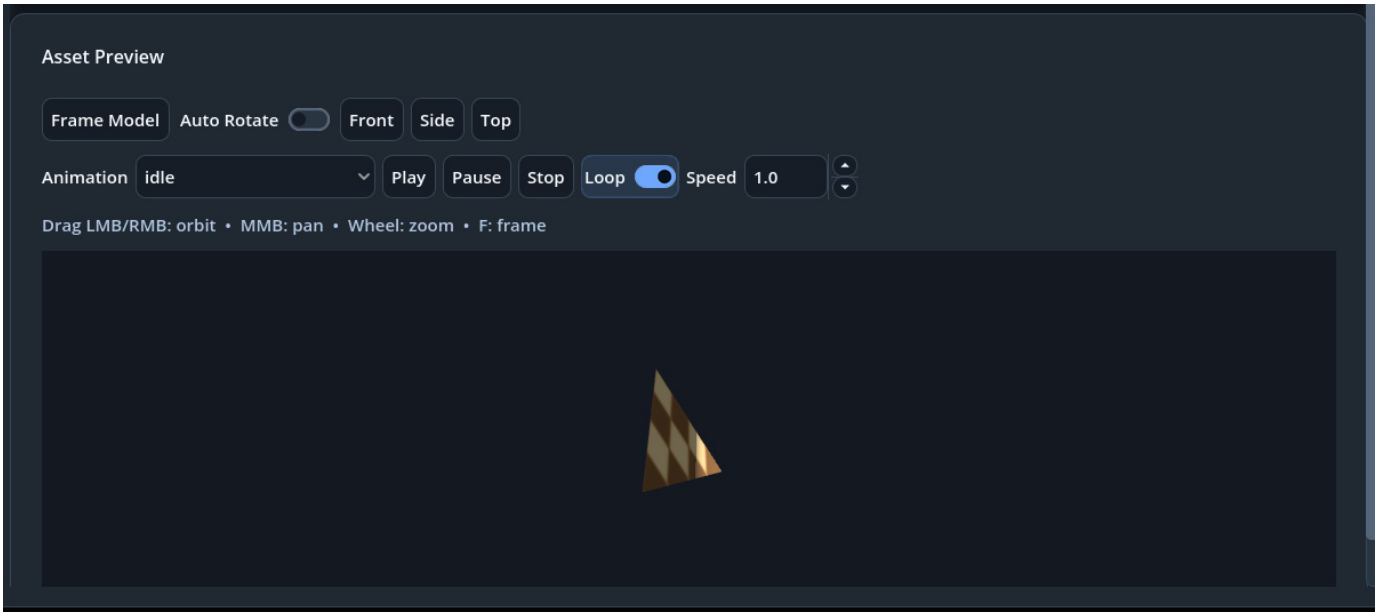


Figure 20. The small MD2 example in the model preview. Its animation consists of vertex frames rather than a modern skeletal rig.

Use a clip speed of 1 while checking the source FPS. Slowing the preview can help inspect a bad frame, but it does not save a corrected source rate.

MDL is not one universal model format

The importer recognises supported **Quake MDL version 6** and **Half-Life/GoldSrc Studio Model version 10** data. Quake MDL uses vertex animation; GoldSrc models can contain a skeleton and named sequences.

Many unrelated applications also use the extension `.mdl`. An arbitrary MDL from another tool is not promised to load just because it appears in the picker. Read the error, identify the real format and export a compatible version if necessary.

MD3

MD3 is the Quake III-style vertex-animated model route. Keep nearby textures, relevant `.skin` mappings and any `animation.cfg` with the source. Those companions can affect what is discovered during import.

A multipart character needs more than a single filename. Check each imported part and the intended assembly in your game. Do not assume that selecting one MD3 automatically constructs every head, torso, leg and tag arrangement from an original game's character system.

A careful check for any retro model

1. Confirm the supported file version.
2. Confirm the skin is visible.
3. Inspect each clip and its start/end pose.
4. Check Forward Axis.
5. Put the asset beside a known-size prop in a test room.
6. Only then build the Character definition and state mappings around it.

12. MagicaVoxel and Qubicle

Both formats describe voxel content, but they do not carry identical animation information.

VOX

Choose Model and Browse for `Source-Models/Guide_Marker.vox`. The importer builds visible voxel surfaces and, when supported timeline data is present, exposes animation in the usual model preview.

The current VOX animation route supports VOX 200 scene data such as shape-frame changes and transform frames. A static VOX remains static. Two objects or two layers in a file are not automatically evidence of an animation.

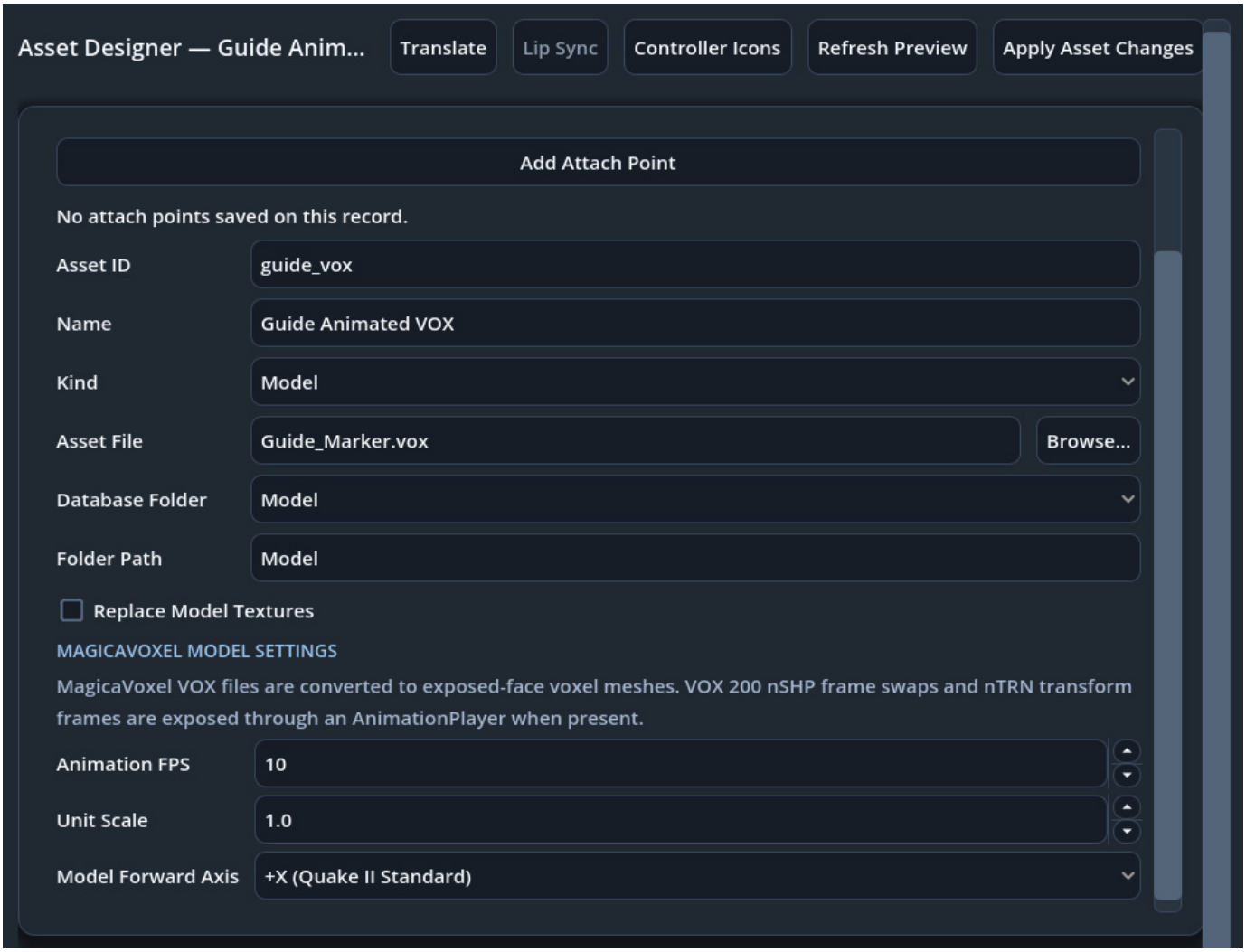


Figure 21. The MagicaVoxel settings. Unit Scale controls the voxel-to-game conversion; Animation FPS supplies the timeline rate.



Figure 22. The animated practice VOX. Its lamp_pulse clip appears in the same preview controls as other imported model clips.

Voxel animation may swap shapes or transform model parts rather than deform a skeleton. That is expected for this kind of source. Check both the clip menu and the actual changing shape.

QB

Qubicle `.qb` imports supported raw or compressed matrices as a static model. Try `Source-Models/Guide_Marker.qb`, then inspect its scale and colours.

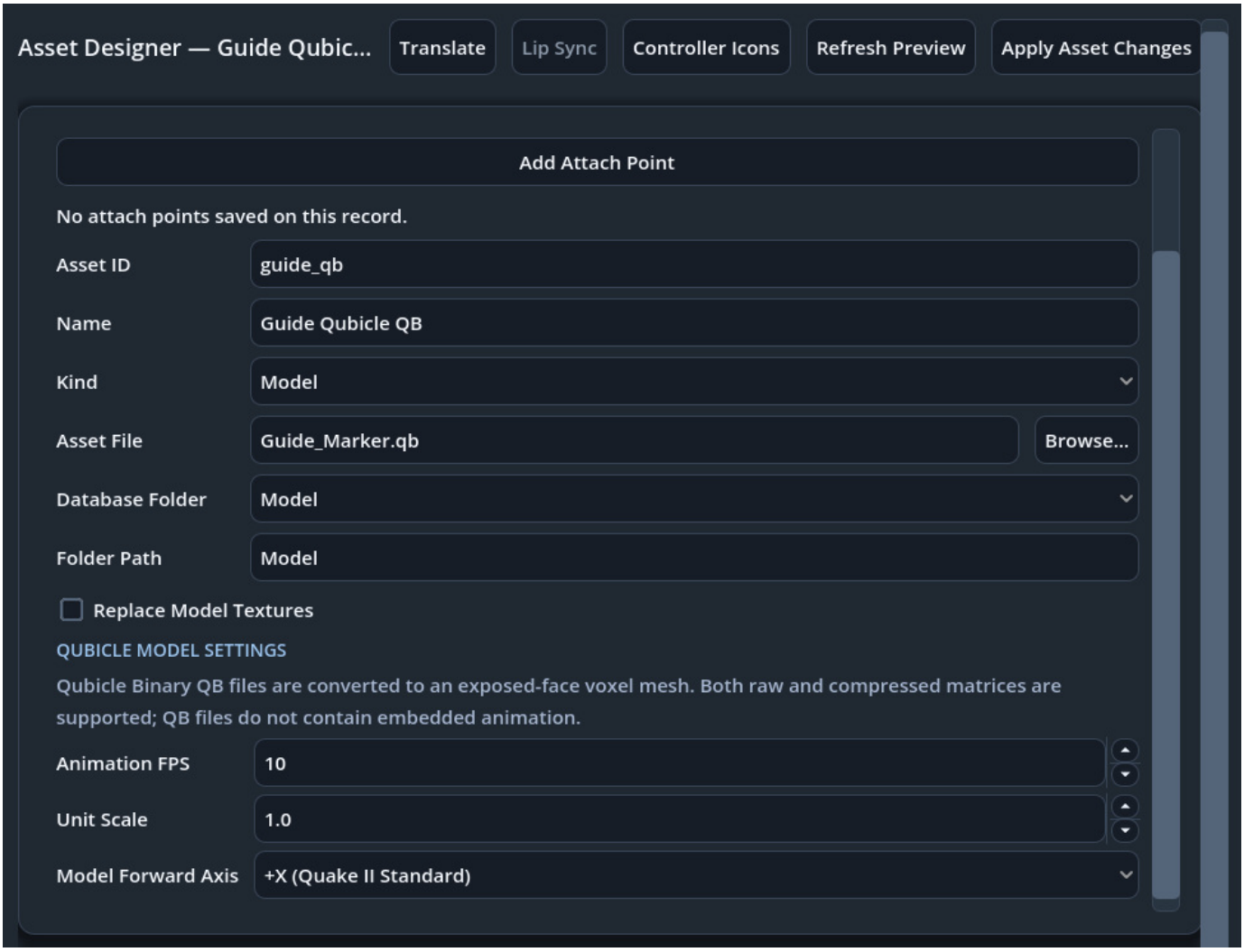


Figure 23. The Qubicle settings. A common control can appear even when the source itself does not contain animation.

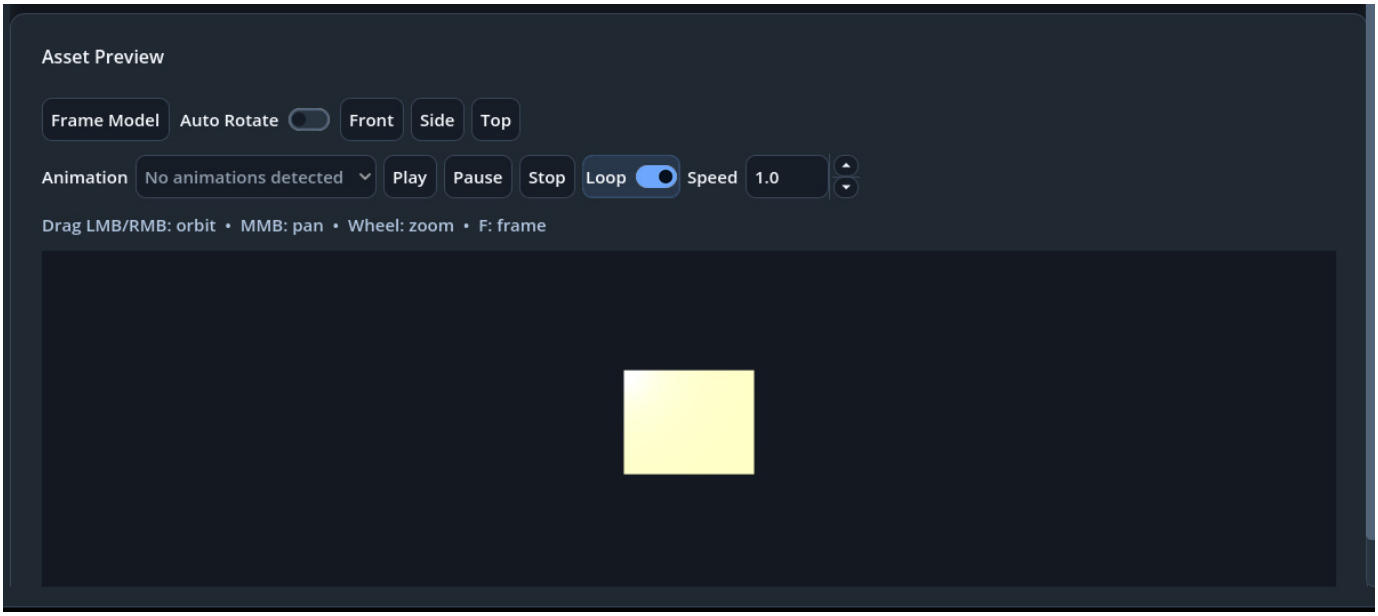


Figure 24. The static QB example. No animation clips are expected for this file.

Changing Animation FPS cannot animate a QB. Use a supported animated source or animate the resulting object through the game systems if it needs to move.

13. LightWave and DirectX

LightWave .lwo

The LWO route reads supported LWOB and LWO2 model data, including model layers, polygons and supported UV or morph information. It is a model import, not a complete LightWave Layout scene importer.

1. Create a Model asset and choose `Source-Models/Guide_Marker.lwo`.
2. Start with Unit Scale 1 and the expected forward axis.
3. Inspect the geometry and material result.
4. Apply and save.

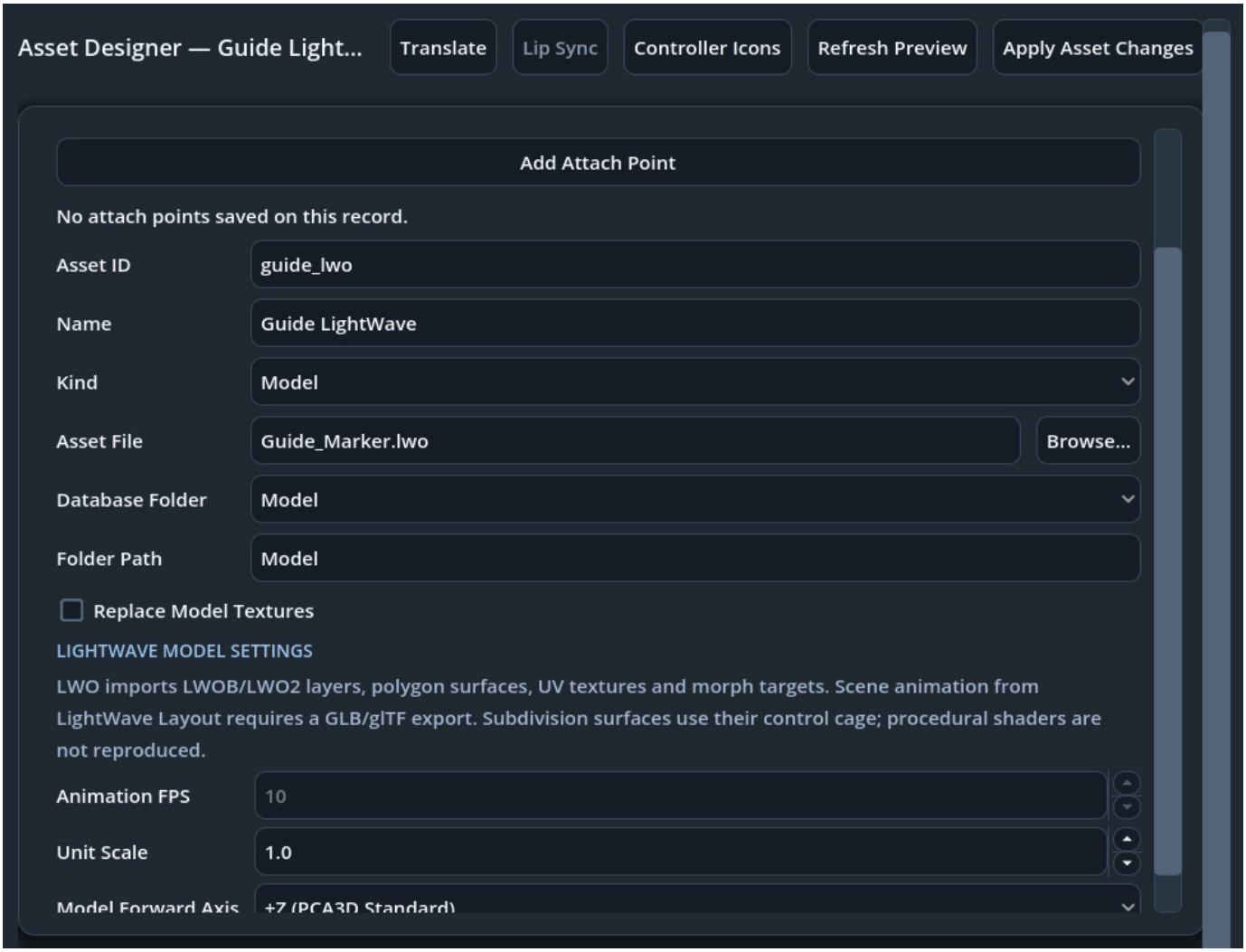


Figure 25. LightWave model settings. Animation FPS is disabled because this route does not import a Layout scene timeline.

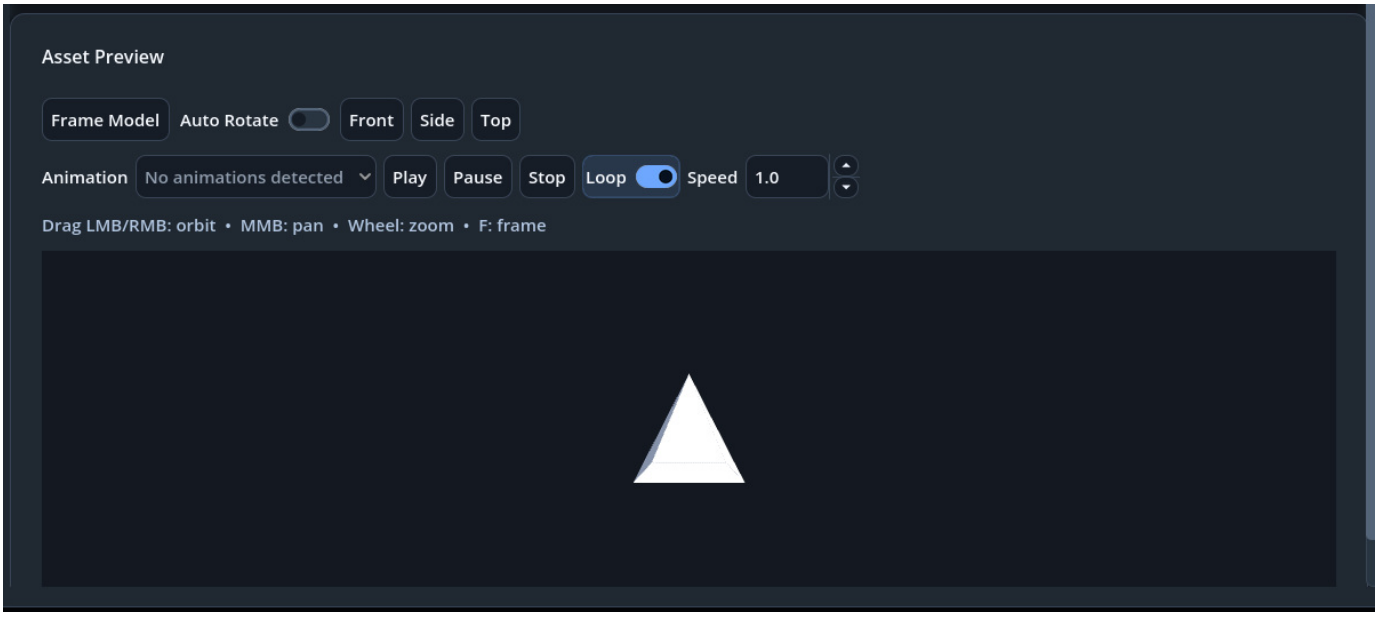


Figure 26. The small LightWave example after import.

Subdivision geometry is represented by the available control-cage model rather than a promise of identical authoring-application subdivision rendering. Procedural LightWave shaders also need conversion or baking. For a finished animated Layout scene, export a suitable GLB/glTF result from your authoring workflow.

DirectX .x

The X importer supports relevant text, binary and MSZIP data, with ordinary frame or skeletal AnimationSet animation. Try `Source-Models/Guide_Marker.x`; its short Bob clip should appear in the animation menu.

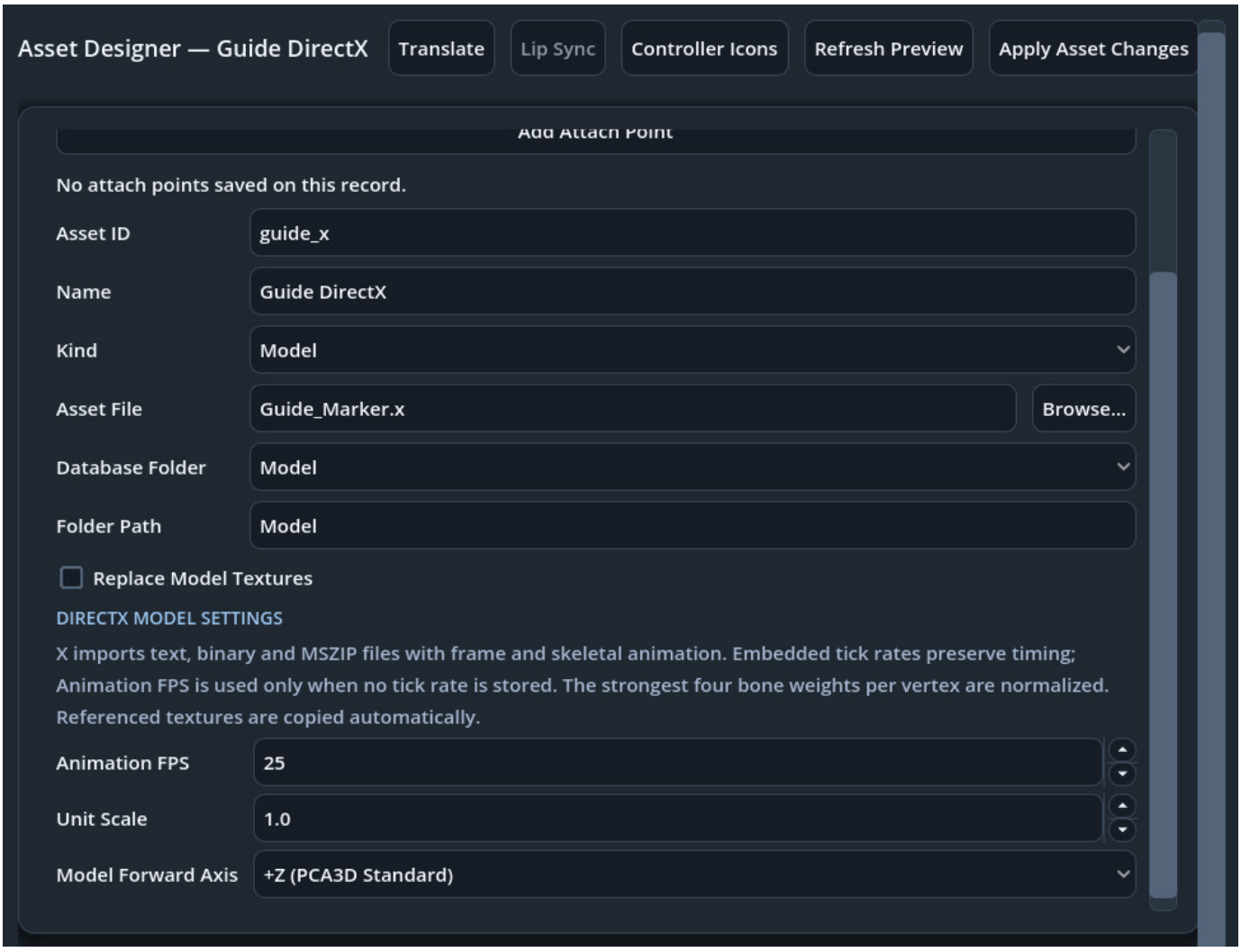


Figure 27. DirectX model settings. File timing information can govern a clip even when an FPS fallback is visible.

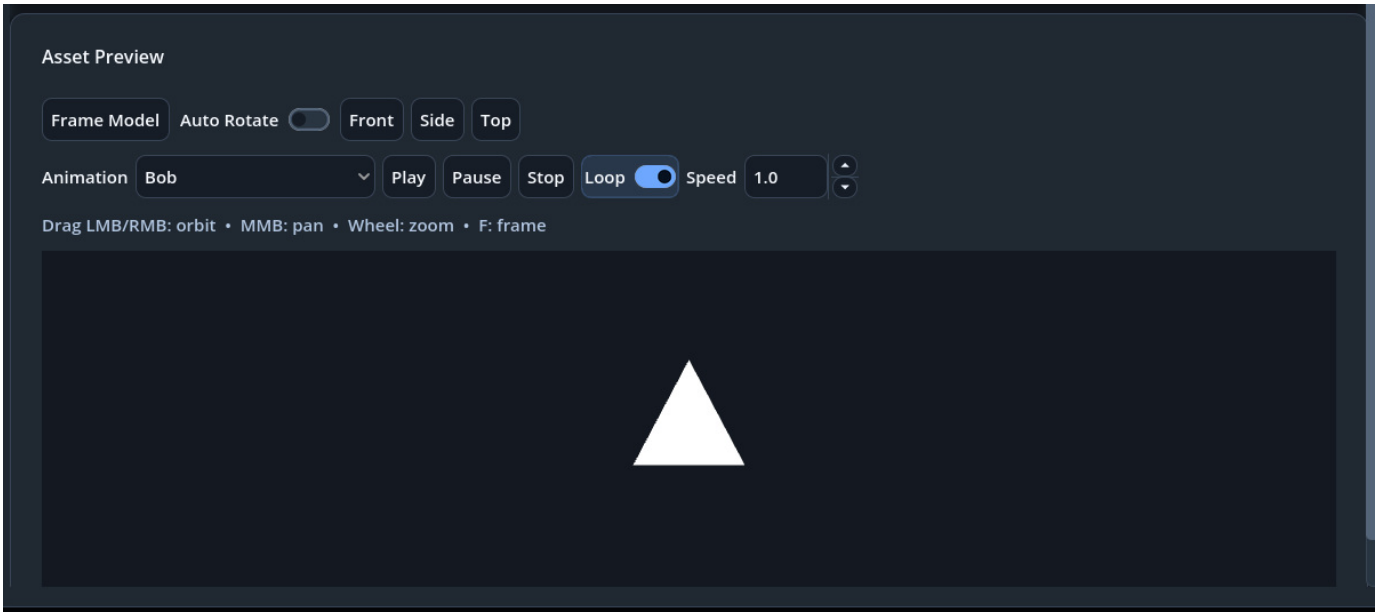


Figure 28. The DirectX practice marker. Play Bob to check that animation reached the imported model.

The file's tick rate is used where supplied. Skinning retains and normalises the strongest four weights per vertex. A source depending on more influences may therefore need adjustment before export. CompressedAnimationSet data is a different case from an ordinary AnimationSet and is not supported by this route.

If the mesh loads but an animation does not, inspect the source's animation type rather than repeatedly changing Unit Scale.

14. Blockbench

Native `.bbmodel` files are imported as Model assets. The current reader targets Blockbench format 4.x and 5.x, including supported cubes, polygon meshes, groups, UVs, embedded PNG textures and numeric transform animation.

1. Add a Model asset and choose `Source-Models/Guide_Marker.bbmodel`.
2. Leave Unit Scale at 0.0625 for the first test.
3. Choose the Sway clip and inspect it at Speed 1.
4. Check the material, then Apply and Save.

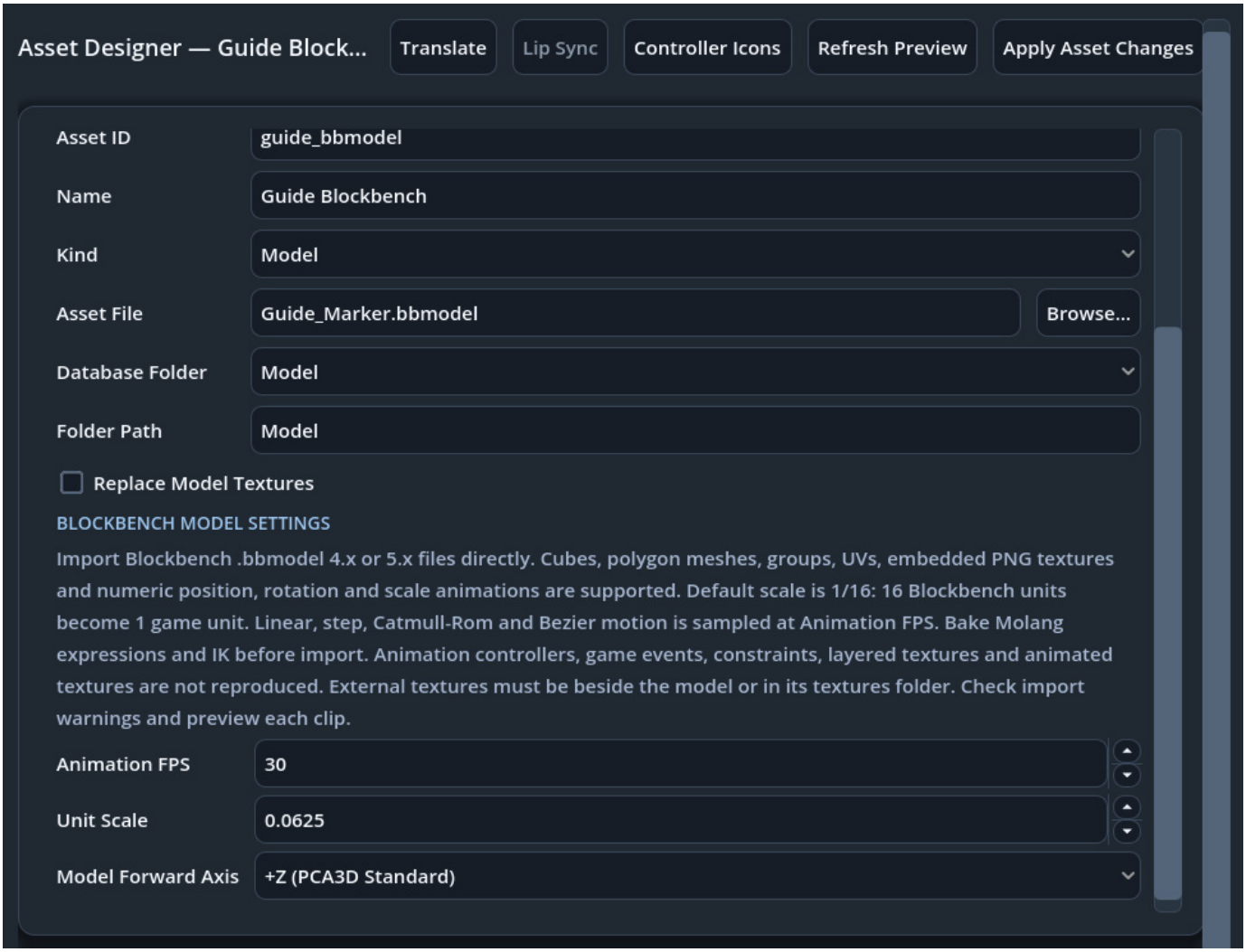


Figure 29. The Blockbench import controls. Sixteen source units normally become one game unit.

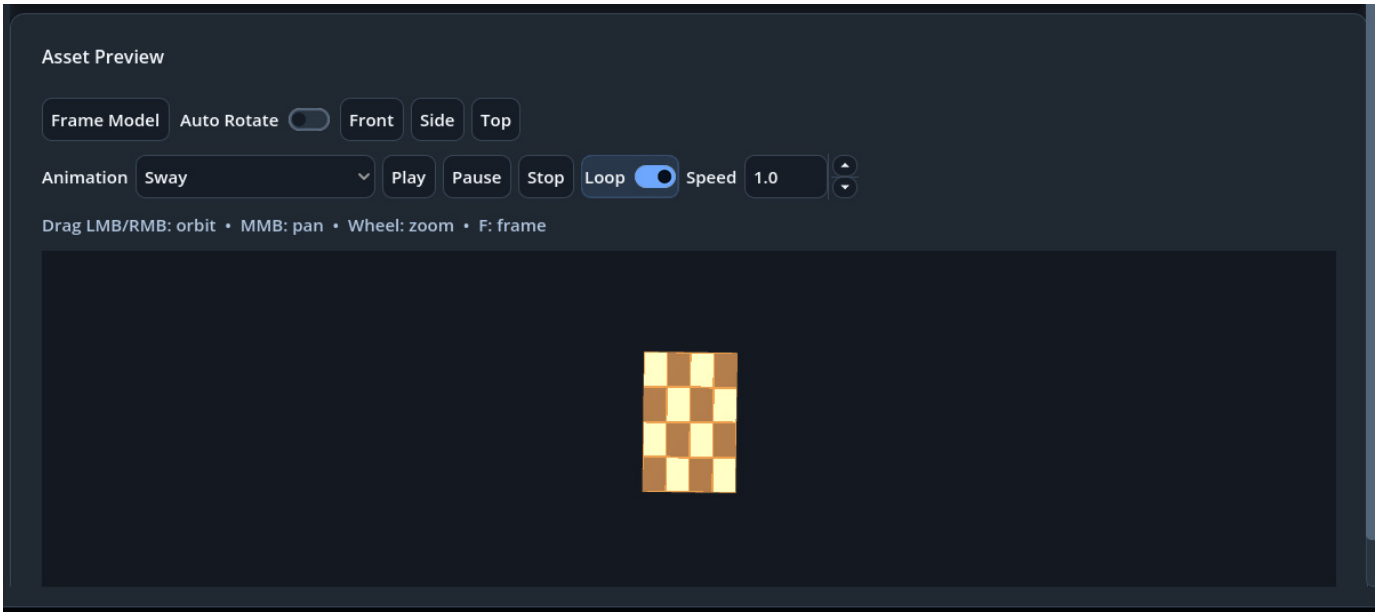


Figure 30. The native Blockbench example with its imported animation.

What needs preparation in Blockbench

Linear, step, Catmull-Rom and Bezier movement is sampled at Animation FPS. Bake Molang expressions and inverse-kinematics motion before importing. Animation controllers, game events, constraints, layered textures and animated textures are not reproduced as a complete Blockbench game runtime.

Keep external texture images beside the model or in its textures folder. Embedded PNG textures are convenient because they travel with the source. Read warnings about unsupported elements; a preview containing only some of the source is not a reason to ignore those warnings.

The Asset Designer imports the model and its clips. It does not decide which imported animation is the character's idle, walk or talk state. Make those assignments in the Character Designer after the asset is checked.

15. Sprite sheets, from picture to frames

A sprite sheet is one picture containing many frames. The importer needs to know how to cut that picture into equal cells and how fast to play them.

The practice `Signal_Lamp_Sheet.png` is 256 pixels wide and 128 pixels high. It has four columns, two rows and 64-by-64 cells. Seven cells contain a lamp frame; the last cell is deliberately empty.

The seven-frame lesson

1. Add an asset with ID `guide_sheet`, Name **Signal Lamp Sheet**, and Kind **spritesheet**.
2. Browse for `Signal_Lamp_Sheet.png`.
3. Choose **Grid (Rows/Columns)** in Split Method.
4. Set Columns to **4** and Rows to **2**.
5. Set Last Row Count to **3**. That keeps three frames from the final row instead of including the empty eighth cell.
6. Set the upper Animation FPS to **8**.
7. Apply Asset Changes. The preview should report **7 frames**.
8. Save Project.

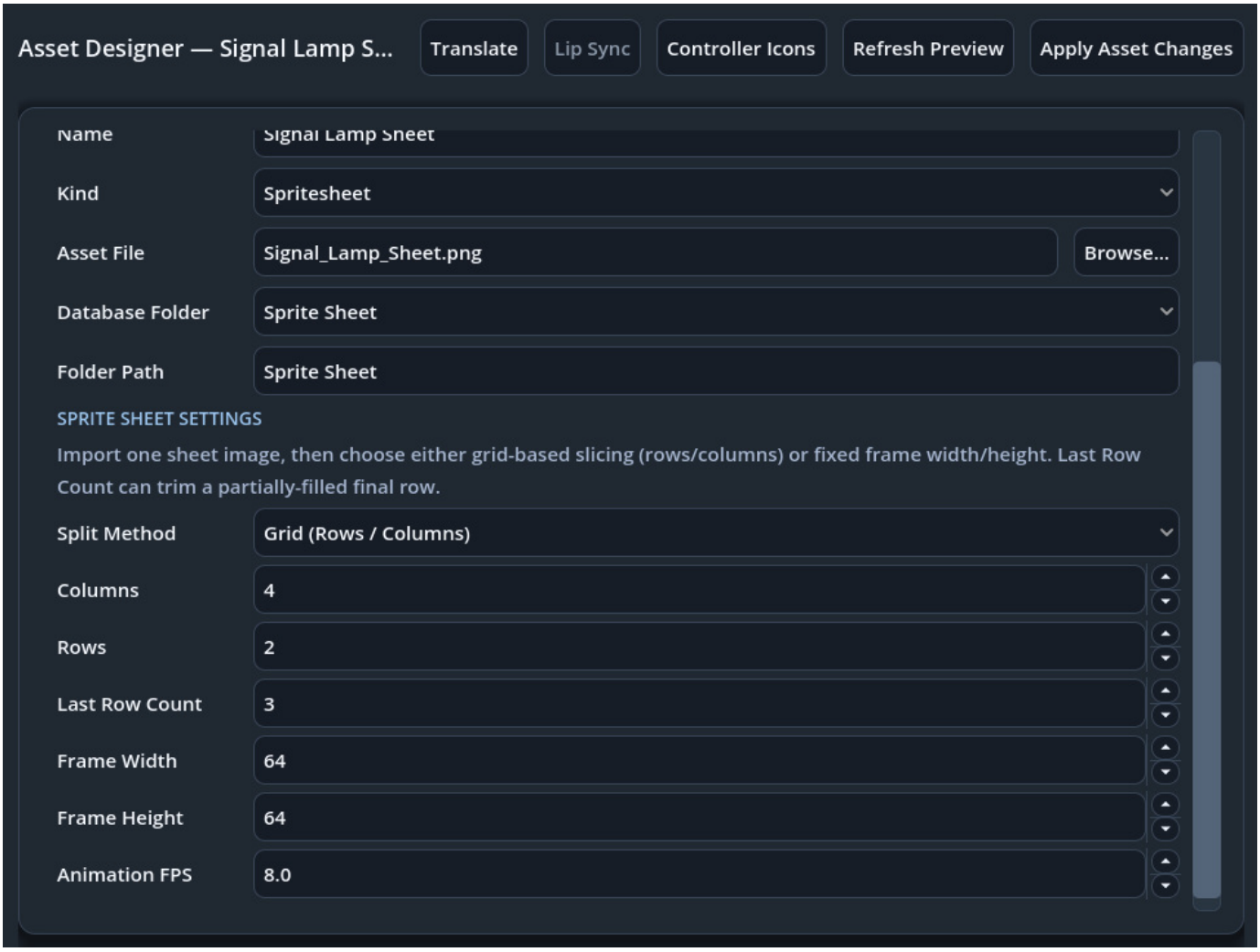


Figure 31. The practice sheet set to four columns, two rows and three frames in the final row. The result is seven frames.

Every slicing field

Field	Range	Meaning
Split Method	Grid or Frame Size	Choose whether you know the number of cells or their pixel dimensions.
Columns	1-512	Number of equal cells across the picture in Grid mode.
Rows	1-512	Number of equal cells down the picture in Grid mode.
Last Row Count	0-512	How many frames to keep in the final row. Zero means a full final row. The useful value cannot exceed Columns.
Frame Width	1-4096 pixels	Width of one cell in Frame Size mode.
Frame Height	1-4096 pixels	Height of one cell in Frame Size mode.
Animation FPS	0.1-240	Saved playback rate for the asset when applied.

Frames are read from left to right across the first row, then left to right across the next row. Last Row Count trims only the final row. It does not remove arbitrary gaps earlier in the sheet.

The alternative: Frame Size

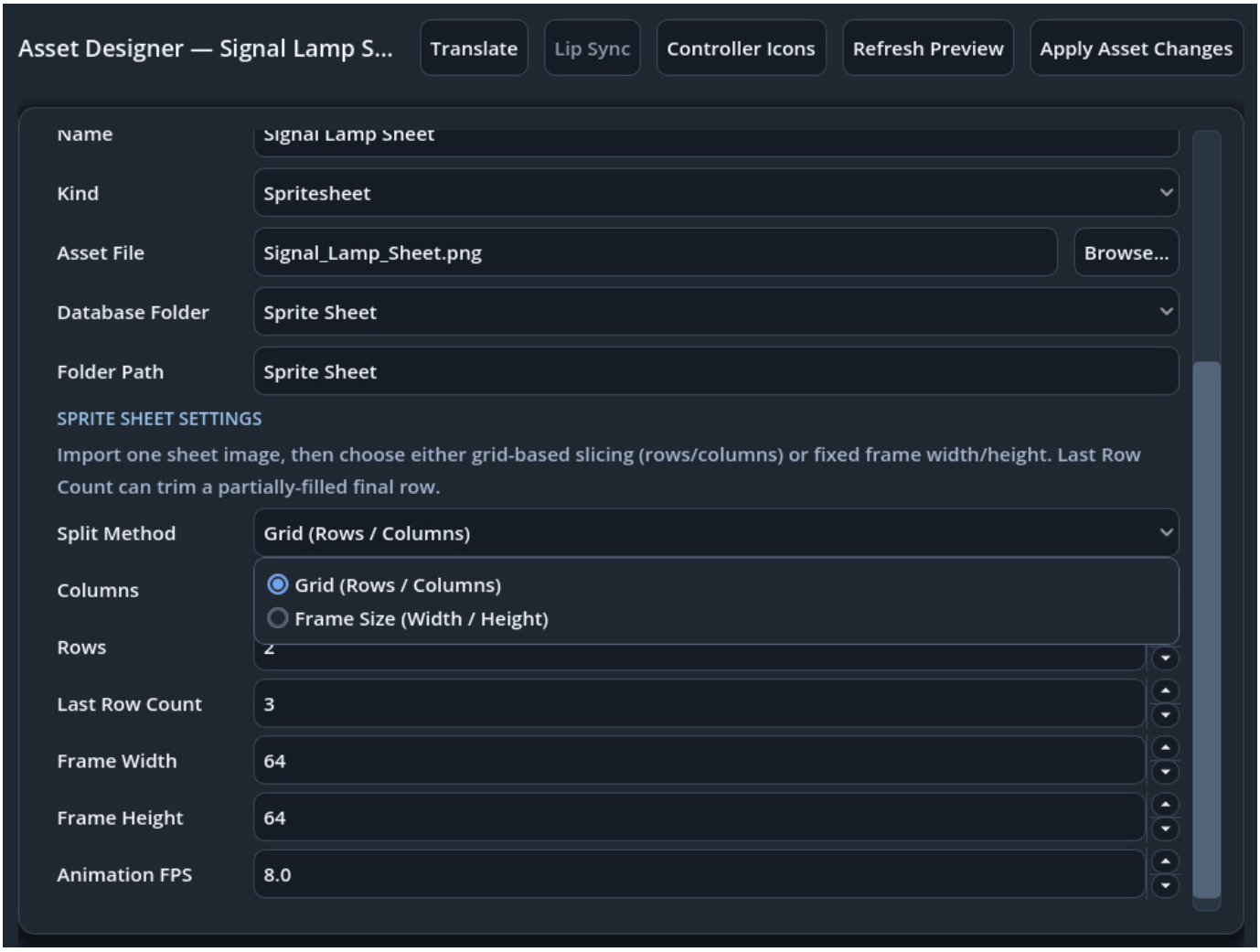


Figure 32. The Split Method menu. Use Grid when you know the row and column counts; use Frame Size when you know each cell's dimensions.

Choose **Frame Size (Width/Height)** and enter Width **64**, Height **64** for the same practice image. This calculates four columns and two rows from the picture size. Keep Last Row Count at 3 to omit the empty final cell.

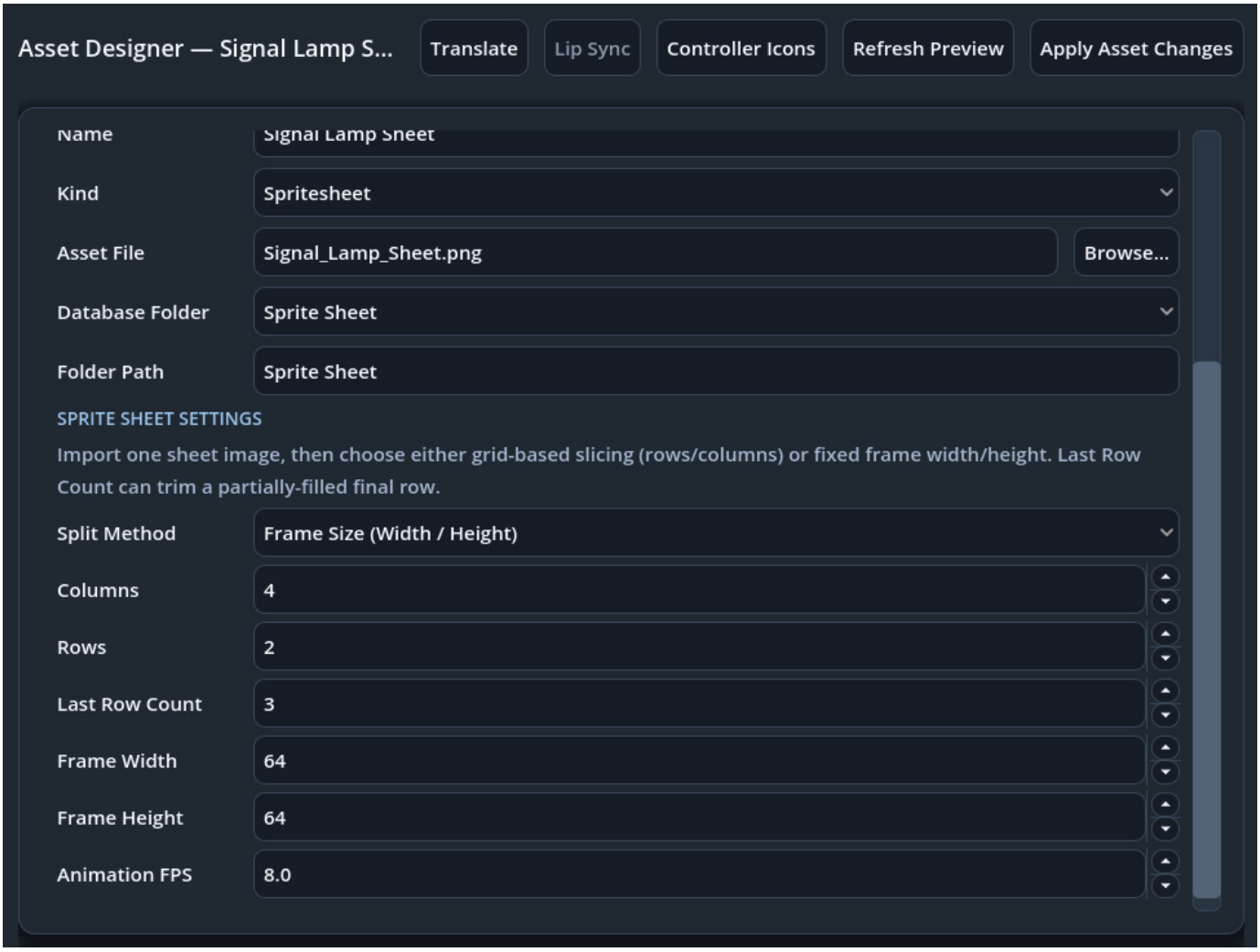


Figure 33. The same sheet sliced by 64-by-64 cell dimensions instead of a row/column count.

Both methods use whole-pixel cells. In Grid mode, cell dimensions are calculated from the picture and counts; in Frame Size mode, incomplete cells at the far edge do not become full frames. Prepare a sheet that divides cleanly.

There are no margin, gutter or arbitrary-rectangle slicing controls in this form. A sheet with spacing between cells should be repacked without that spacing, or exported as separate frame images.

Frame count and duration

For this example, the first row contributes 4 frames and the last contributes 3: **7 total**. At 8 FPS, one full pass lasts 7 divided by 8, or **0.875 seconds**. The preview's enlarged display does not change those source cell dimensions.

> Check your work. Pause the sequence, step through all seven frames and confirm that the empty final cell never appears. If it does, check Last Row Count before changing the artwork.

16. The 2D animation preview

Sprite sheets and separate-image animations share a sequence player. It is a useful final check for slicing, ordering and timing.

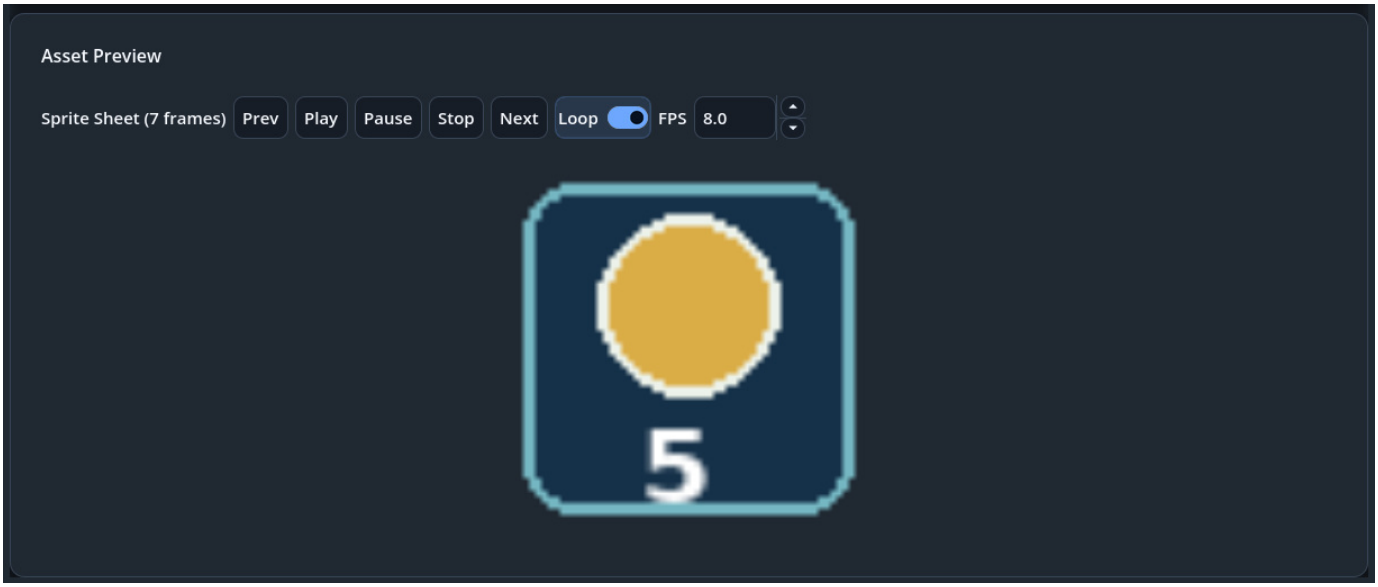


Figure 34. The enlarged sprite preview. Give this area more room when checking individual frames.

Control	Result
Prev	Goes to the previous frame, wrapping around the sequence.
Play	Starts or resumes playback.
Pause	Holds the current frame.
Stop	Stops playback and returns to the first frame.
Next	Goes to the next frame, wrapping around the sequence.
Loop	Repeats the sequence. With it off, playback ends at the last frame.
FPS beside the player	Changes the current preview rate. Use the upper Animation FPS field to save the asset's rate.

Pause before using Prev or Next for a frame-by-frame inspection. Those stepping buttons do not automatically turn off the running timer, so a playing sequence can move again immediately after you step.

The player normally starts automatically when there is more than one frame. A single-frame result is often a slicing or selection problem, but it can also be a legitimate one-frame animation asset.

Look for a consistent canvas and baseline. If an actor jumps around inside the preview, some frames may have different dimensions or inconsistent placement. Fix that in the source frames; the FPS control cannot correct registration.

17. Separate numbered images

Choose Kind **animation** when each frame is a separate image. Browse allows multiple selections for this kind.

1. Add `guide_frames`, named **Numbered Lamp Frames**.
2. Choose Kind **animation**.
3. Browse into the examples' `Numbered-Frames` folder.
4. Select `Lamp_1.png`, `Lamp_2.png` and `Lamp_10.png` together. Use your file picker's multiple-selection gesture.
5. Confirm that the form reports three selected frames.
6. Set Animation FPS, apply and save.
7. Pause the preview and step through the sequence. The intended order is 1, 2, 10.

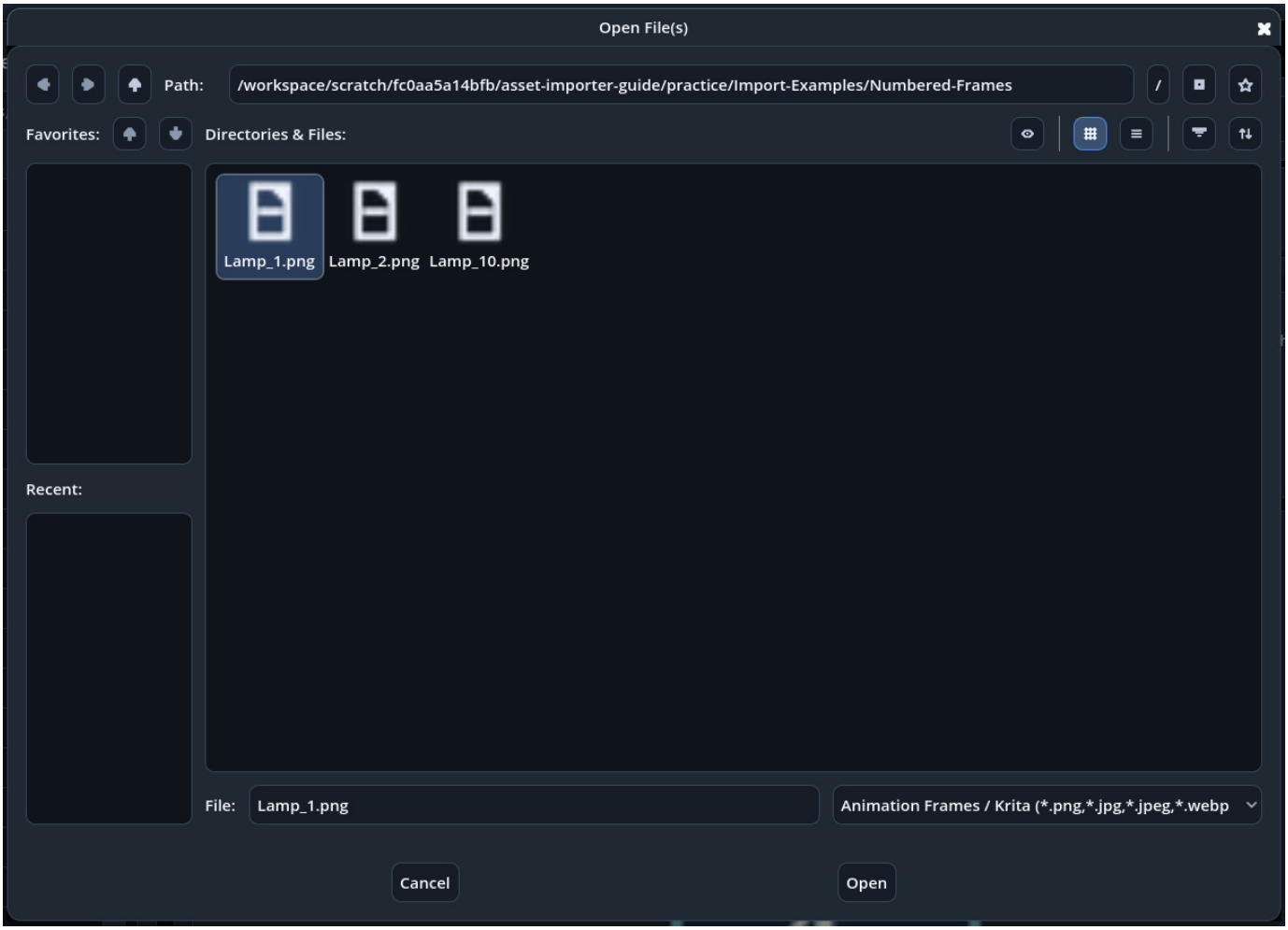


Figure 35. The frame picker allows a set of images to be selected together. Extract the examples before opening this folder.

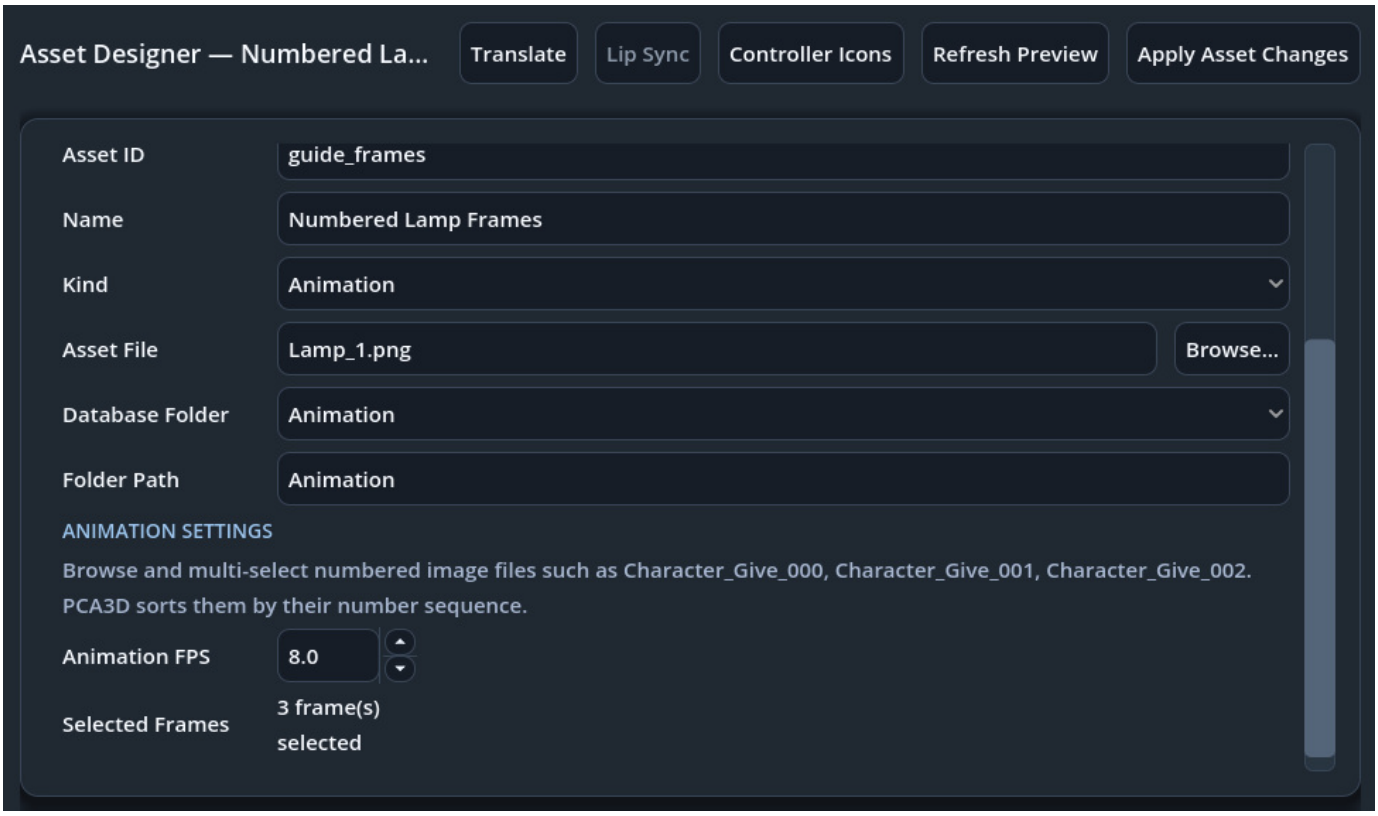


Figure 36. The separate-image animation after importing. Asset File shows the first file, while the selected-frame count represents the whole sequence.

How ordering works

The importer uses the final numeric suffix in the filename. This avoids the usual alphabetical mistake of placing frame 10 before frame 2. Names with a common prefix work best: `Walk_001.png`, `Walk_002.png`, `Walk_003.png`.

Prefixes also affect ordering. Do not mix `Walk_001`, `Jump_001` and `Talk_001` into one asset expecting the importer to invent three independent clips. Make a separate Animation record for each action, or use the supported Krita clip workflow.

The selected files are copied into an animation folder associated with the asset ID. Asset File displaying one filename does not mean the other frames were discarded. The frame list and selected-frame count are the important checks.

The Animation picker accepts PNG, JPG/JPEG, WebP, BMP and TGA frame images, plus a single Krita file through the special Krita route. Do not mix a `.kra` with ordinary frame images in the same selection.

> Check your work. Change the upper Animation FPS, Apply, select another asset and return. Confirm the saved setting is still there; then compare it with the lower player's temporary FPS control.

18. Krita drawings and animation clips

A Krita file can be imported as a saved picture or as a paint animation. Choose the asset Kind according to what you want to use.

Kind chosen before browsing	What the .kra supplies
Image	The saved merged drawing.
Sprite Sheet	The saved merged drawing, which you then slice using the normal sheet controls.
Animation	Supported paint-animation frames and named clips.

The importer reads the archive and creates usable image files. It does not require Krita to be installed merely to decode this supported content, and an exported game does not need Krita running.

Import the still first

Create an Image asset named **Krita Lamp Still** and Browse for `Signal_Lamp.kra`. Apply and save. You should see the saved merged lamp image.

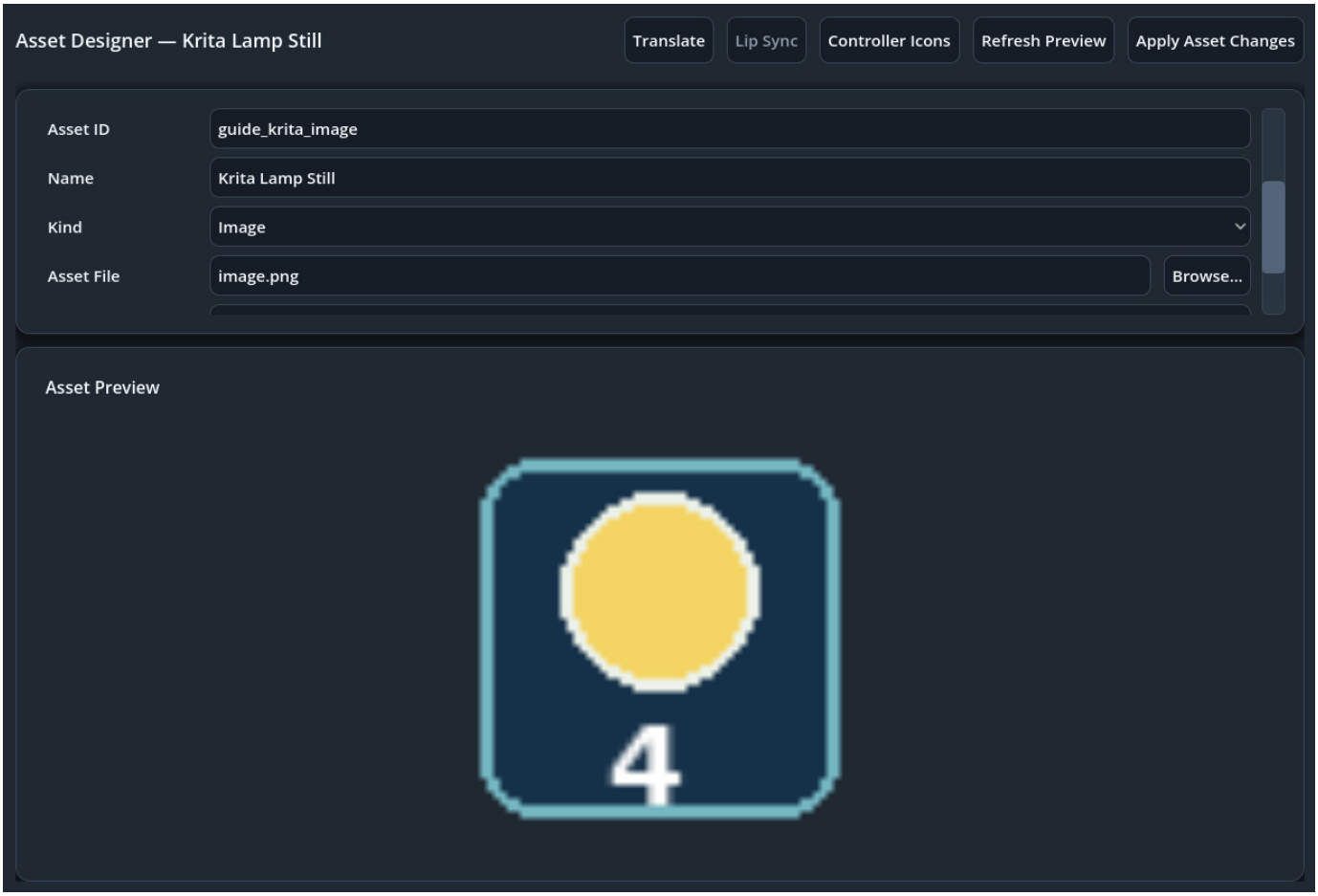


Figure 37. A Krita file imported as an Image. This is the merged saved picture, not an animation player.

Import the animation

1. Add another record, `guide_krita_anim`, named **Krita Lamp Clips**.
2. Choose Kind **animation** before browsing.
3. Select only `Signal_Lamp.kra`.
4. Wait for its frames to be prepared.
5. Open the Krita clip menu and choose **Idle** or **Pulse**.
6. Check the selected-frame count, source FPS and preview.
7. Apply Asset Changes with the clip you intend this asset to use, then save.

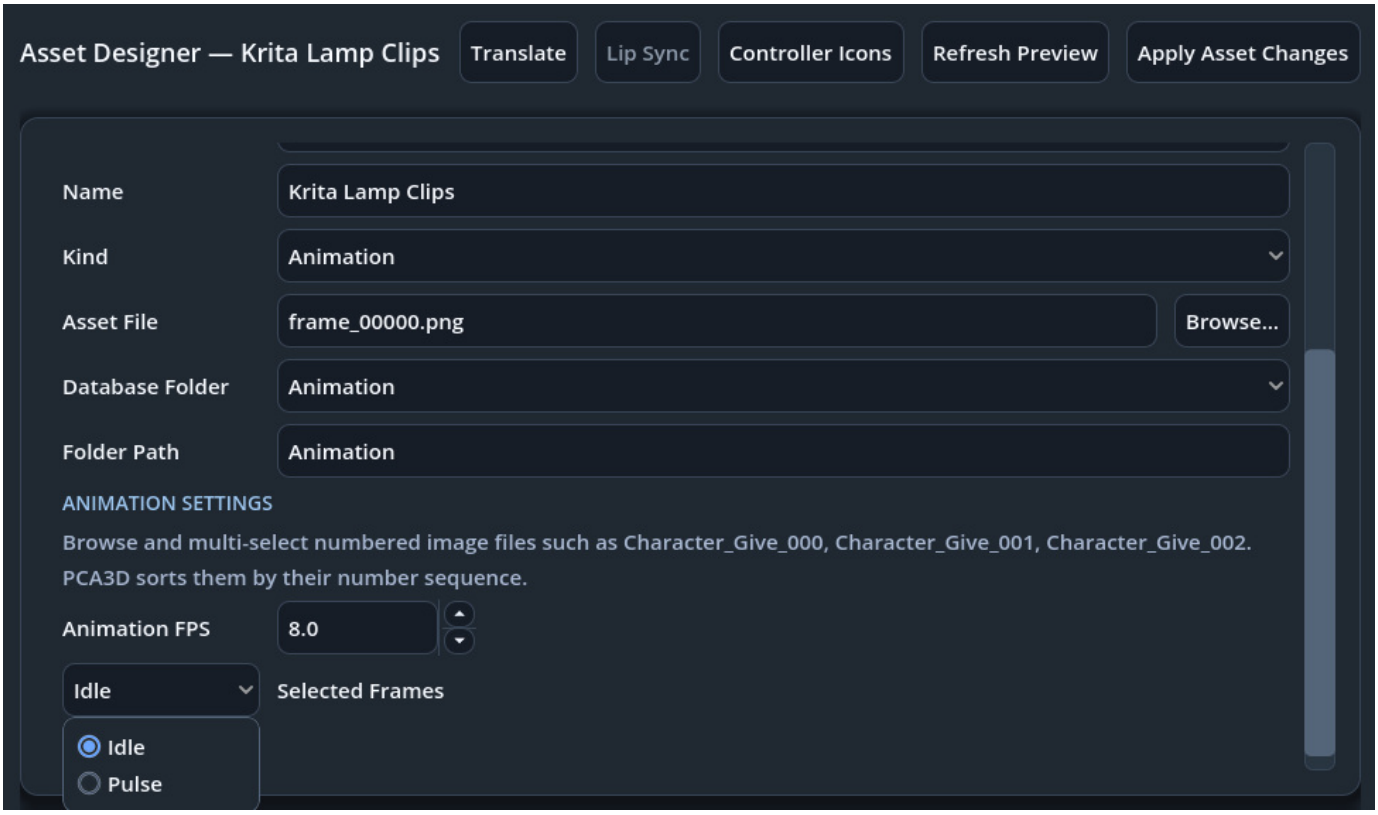


Figure 38. The practice Krita file exposes Idle and Pulse as separate clip choices.

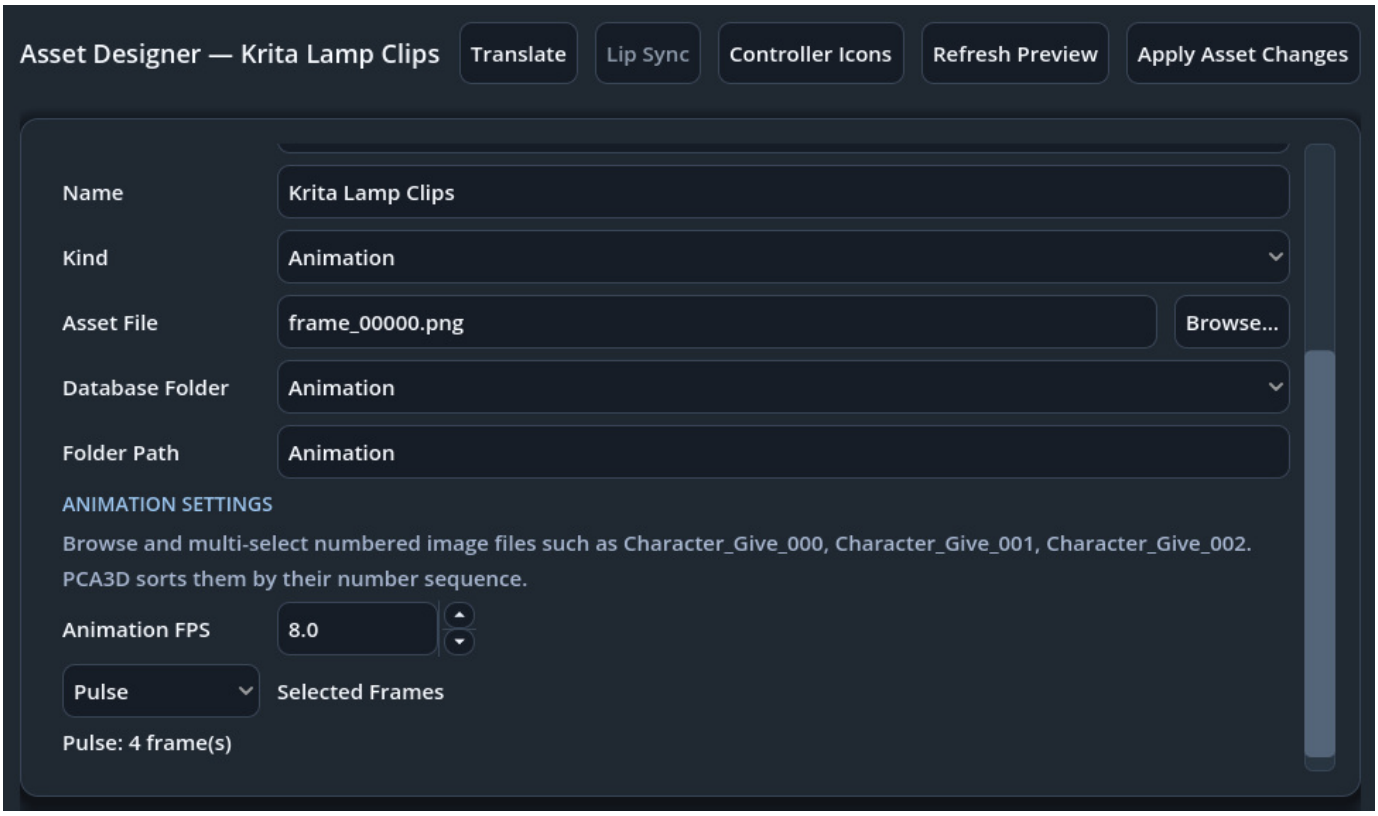


Figure 39. Pulse selected for the animation asset. Selecting a clip changes the frame list and source timing used by this record.

Preparing a Krita document

Supported visible top-level groups become named clips. Put a walk animation in a group named `Walk`, for example. Top-level groups beginning with an underscore are excluded from that clip list. Ungrouped shared artwork can be included across clips.

Held frames remain held; the importer does not invent in-between drawings. Keep the document saved before importing so its merged image and timeline data represent the version you want.

If you need several clips as independent game assets, duplicate the asset record, choose another clip on the duplicate, then apply it. Use different IDs and clear names. The records can share the same imported source package.

Limits and a dependable fallback

The current paint-animation route is intended for supported 8-bit RGBA paint layers. Its limits include a 16,777,216-pixel canvas and 1,000 output timeline frames. Animated transforms, masks, special blending or other unsupported features may not match Krita's display.

If a complex file differs, export a PNG sequence from Krita and use the separate-image Animation route. That gives you a straightforward set of finished frames to inspect. There is no automatic live update while painting; save the source, reimport it, then check the result.

19. Live2D Cubism bundles

Choose **cubism** for a Live2D Cubism model export. Browse for the `.model3.json`, with its referenced model and texture files available in their expected relative locations.

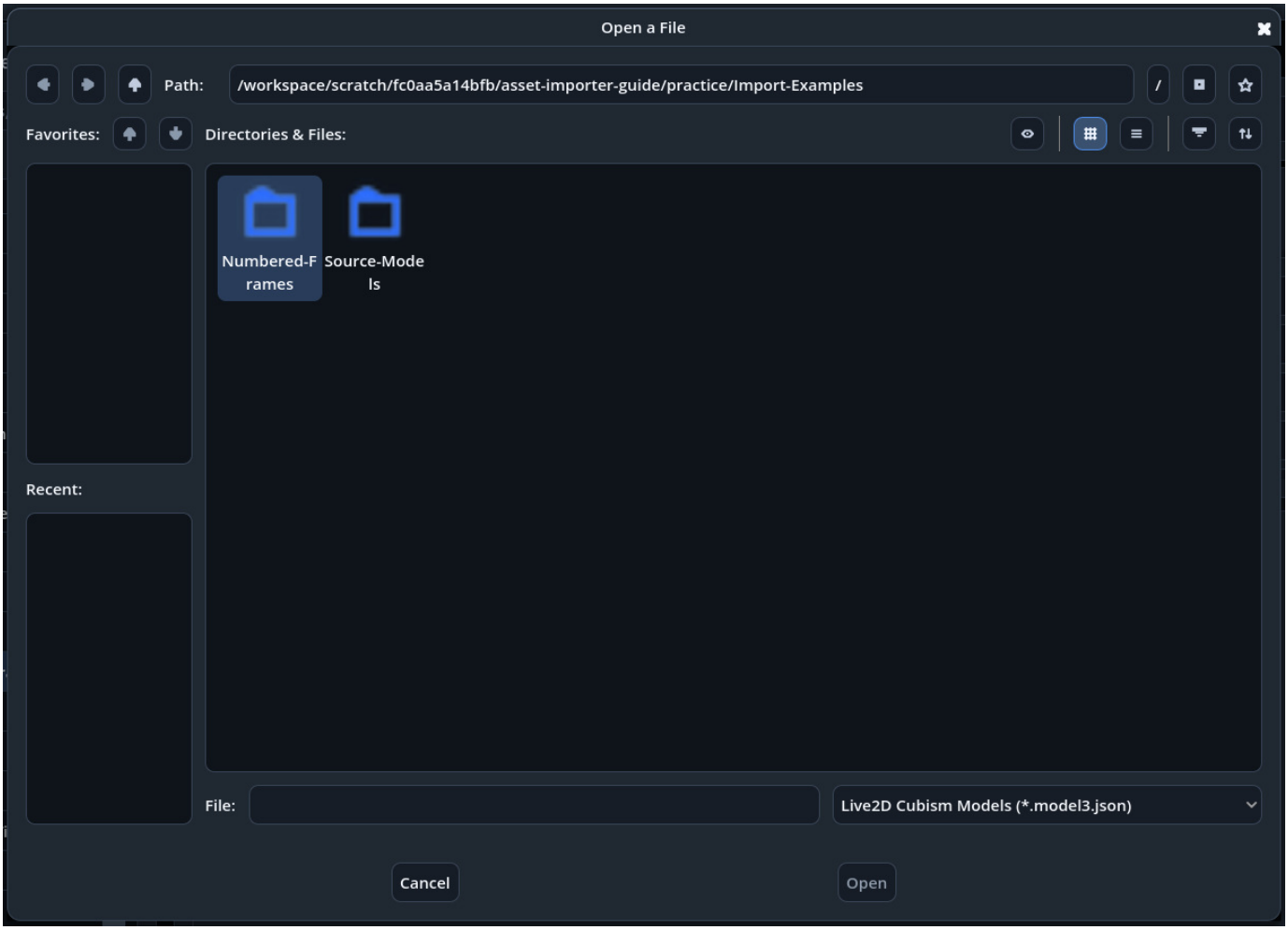


Figure 40. The Cubism file picker expects a .model3.json export. This shows the import route; it is not a screenshot of a functioning Live2D rig.

The importer inspects the bundle and copies its required companions. Its built-in asset preview can show a model texture, but **animated playback requires a compatible Live2D Cubism extension**. Importing the JSON alone does not install such an extension.

1. Export a complete model bundle from the Live2D workflow.
2. Keep the `.model3.json`, `.moc3`, textures and referenced companion files together.
3. Create a Cubism asset and Browse for the `.model3.json`.
4. Read the import status and check any available texture preview.
5. Apply and save.
6. Test the required extension and character use in the actual target build.

Do not browse for a loose texture and call that a Cubism model. The bundle's model data and runtime support are separate requirements. This guide's practice archive does not include a third-party Live2D character or pretend that a texture preview proves animated playback.

20. Audio files

Import sound effects, music and voice recordings as Kind **audio**. The picker offers Ogg, WAV and MP3. Use a clear name for each recording, especially when it is going to be chosen from dialogue or action lists.

1. Add `guide_chime`, named **Guide Chime**.
2. Choose Kind audio and Browse for `guide-chime.wav`.
3. Click **Play Audio** to hear it, then **Stop**.
4. Try the seek slider with a longer recording. Seeking begins playback at the chosen position.
5. Apply and Save Project.

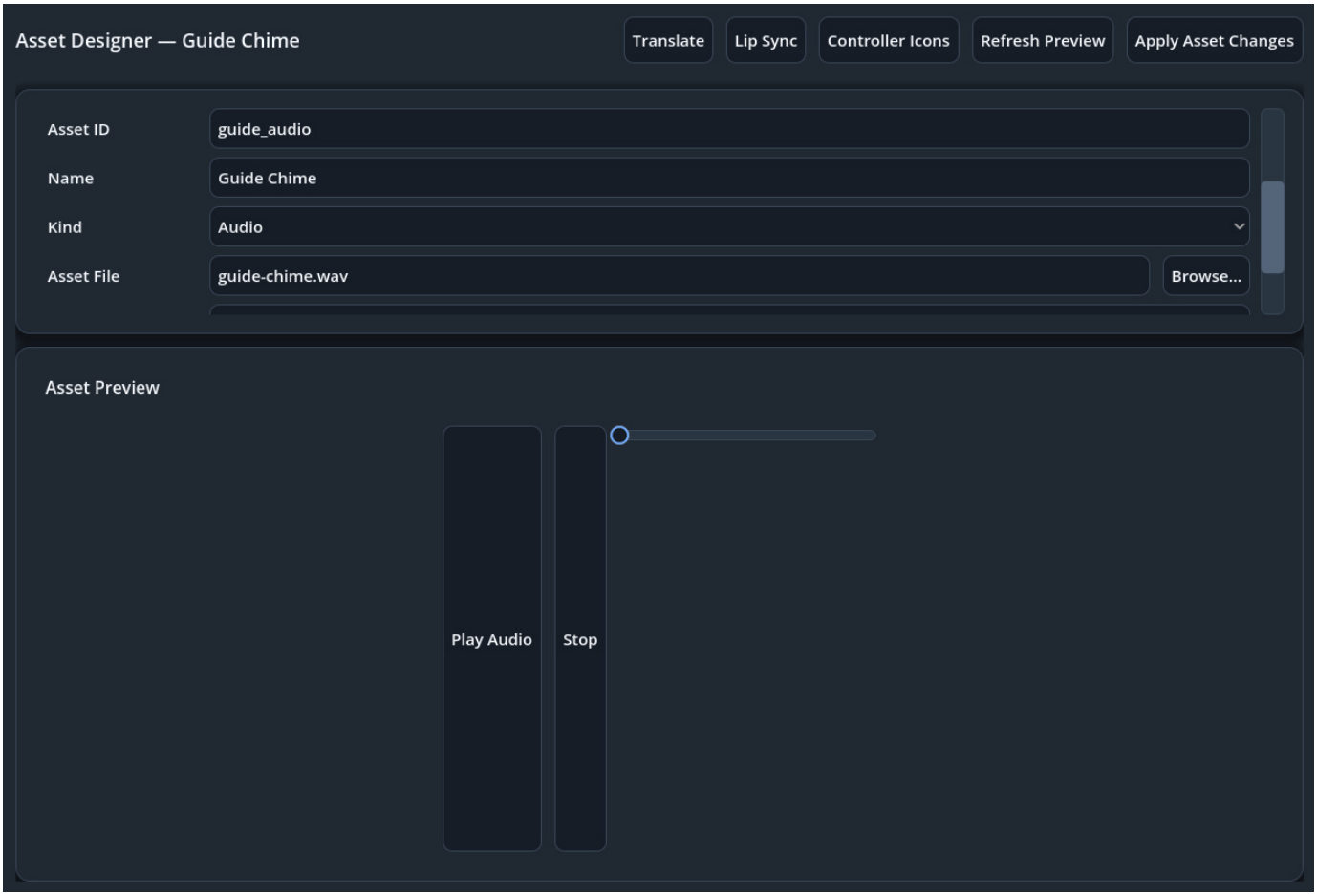


Figure 41. The audio preview has Play Audio, Stop and a seek bar. The practice file is a short chime.

This form is not the game's mixer. It has no general volume-bus editor or saved music-loop setup. Choose volume, buses, looping and playback behaviour in the game settings, sound action or other system using the asset.

For voice work, one line per recording is usually easier to manage. Use names such as `ivo_greeting_en.wav` rather than a folder full of `audio1.wav` and `audio2.wav`. Keep a written record of who speaks each line and which language it contains.

An imported Audio asset enables the Lip Sync tool once the record is applied. A sound effect can be a valid Audio asset without needing mouth cues. The next chapter is for spoken recordings.

21. Lip sync for imported recordings

Lip sync stores timed mouth cues on a particular audio recording. A separate character or portrait mapping tells PCA 3D how that mouth cue should affect the visible character. The track does not create a facial rig that is missing from the model.

The screenshots use an existing Clockmaker voice recording so that the tool shows real speech data. Do not use the practice chime as a test of speech recognition.

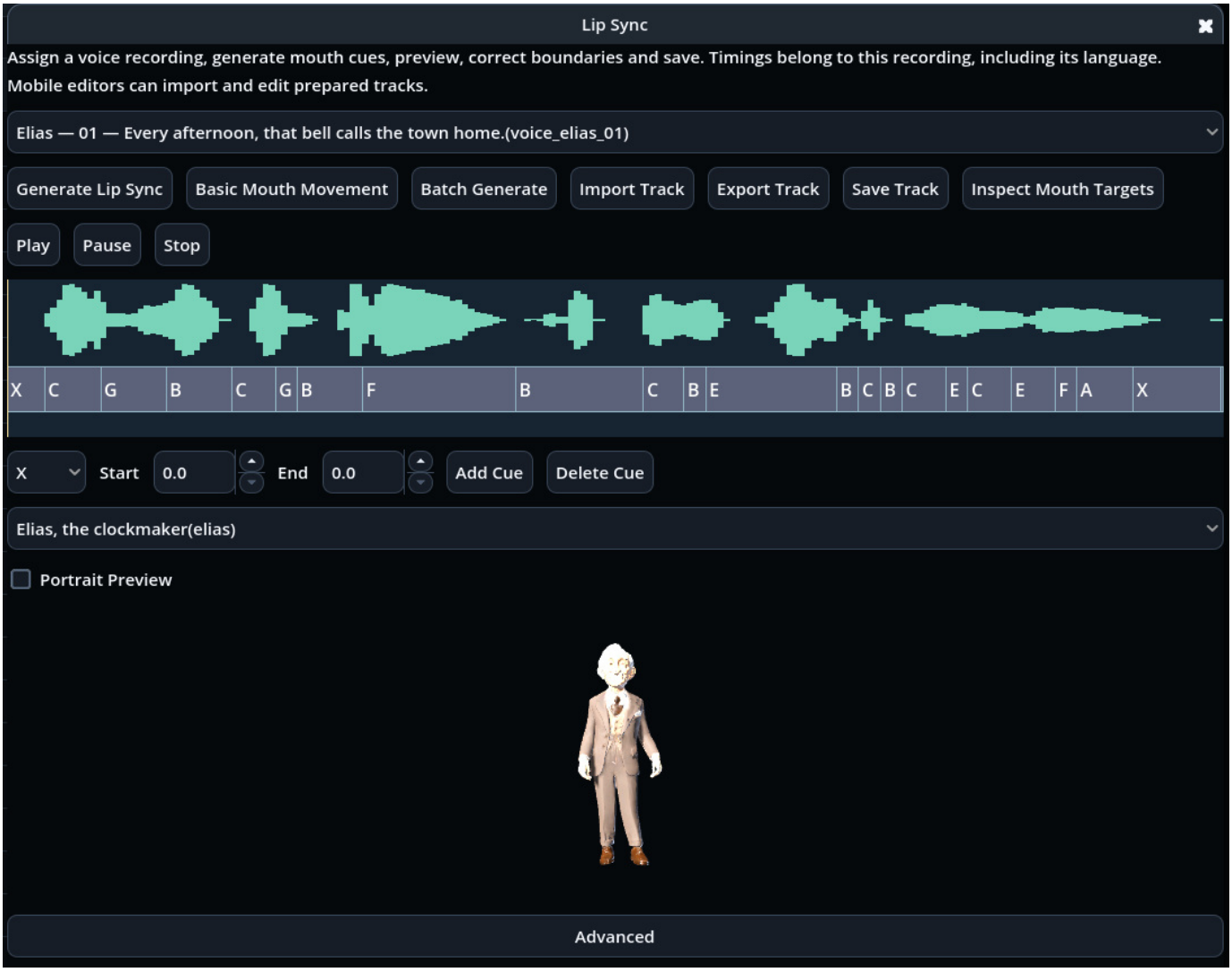


Figure 42. The Lip Sync window opened from a Clockmaker voice asset. The waveform and cues belong to the chosen recording.

The usual workflow

1. Import and apply a spoken recording as Audio.
2. Click **Lip Sync**.
3. Check the recording selector at the top.
4. Open Advanced and set **Recording Language** to the language actually spoken in the file.
5. Click **Generate Lip Sync**, or use **Basic Mouth Movement** for a simpler opening/closing result.
6. Choose a character with configured mouth targets and play the result.
7. Correct cue shapes or boundaries where needed.
8. Click **Save Track**, then save the project.

Automatic generation uses the desktop recogniser while authoring. Exported games use the prepared track and do not need to run that recogniser. Mobile editing can use imported prepared tracks where generation is unavailable.

The buttons and selectors

Control	Purpose
Recording selector	Switches to another imported Audio asset. Check unsaved edits before switching.
Generate Lip Sync	Analyses the recording and creates mouth-shape cues.
Basic Mouth Movement	Makes a simpler amplitude-based mouth-movement result.
Batch Generate	Selects several recordings for generation. It can replace their existing cues.
Import Track / Export Track	Reads or writes supported lip-sync JSON, including compatible Rhubarb-style cues.
Save Track	Applies the edited track to this recording's asset data.
Inspect Mouth Targets	Helps inspect the character's available mouth targets.
Play / Pause / Stop	Auditions the recording and its mapped mouth movement.
Waveform and cue timeline	Shows timing, supports seeking and selecting cues.
Shape, Start and End	Edits the selected cue. Timing is in seconds.
Add Cue / Delete Cue	Inserts a cue into available time or removes the selected cue.
Character selector	Chooses a character for preview, not a new owner for the audio file.
Portrait Preview and portrait selector	Checks the portrait's mouth mapping rather than the full scene character.

Editing a cue

Pause near the sound you want to correct, select the cue and change its shape or Start/End values. The controls use hundredths of a second. Keep boundaries in time order and avoid overlaps or backwards ranges. If the editor rejects a value, fix that timing rather than assuming the rejection was ignored.

Cue	General mouth shape
X	Rest or closed resting mouth.
A	Closed lips, such as M, B or P.
B	Teeth close together, a relatively narrow shape.
C	Open mouth.
D	Wider open mouth.
E	Rounded mouth.
F	Puckered mouth.
G	Lower lip against upper teeth.
H	Raised tongue shape, such as L.

These labels are targets for your mapping. A model's actual blend-shape or bone names can be completely different. Configure the correspondence in the character or portrait's lip-sync settings.

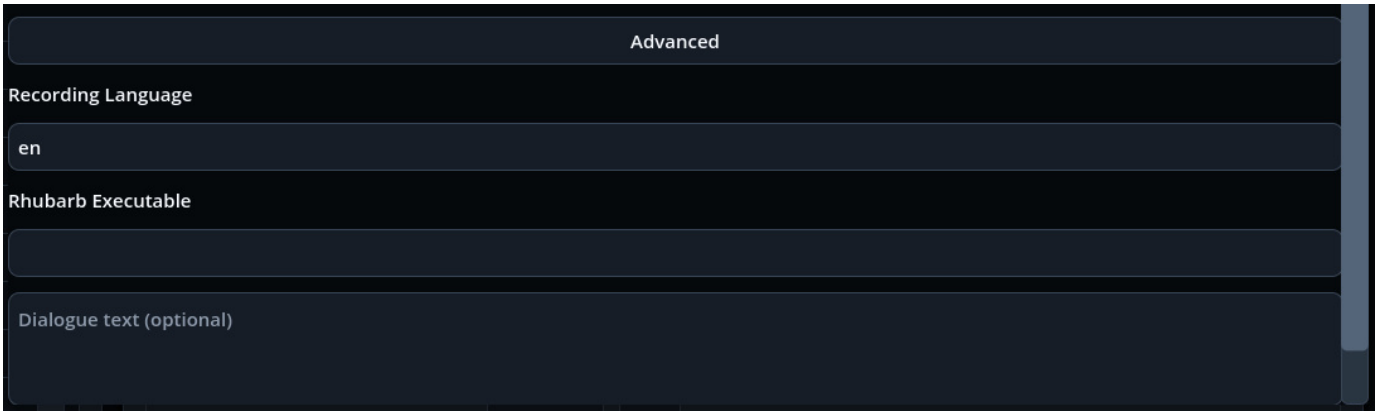


Figure 43. The Advanced section contains recording language, the recogniser path and optional dialogue text.

Recording Language describes the recording, not the language of a subtitle currently on screen. **Rhubarb Executable** can remain empty to use the bundled desktop recogniser. A custom executable must support the expected JSON output. **Dialogue text** can supply an optional transcript.

Batch work and replacement audio

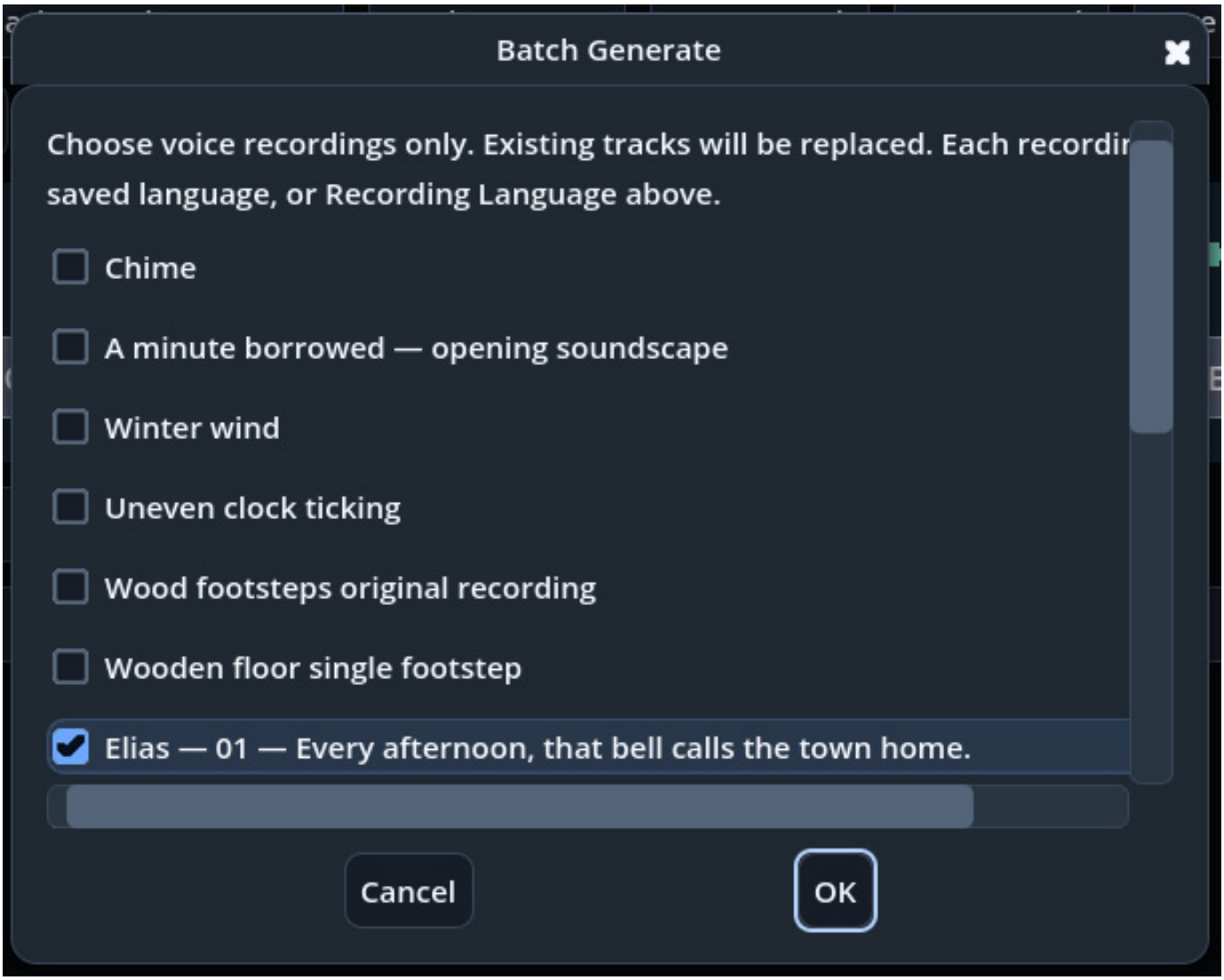


Figure 44. Batch Generate lets you choose the recordings to process. Review existing corrected tracks before replacing them.

Select only the recordings you intend to regenerate. Generation is a background job and locks relevant controls while it runs. The recording-analysis workflow is limited to ten minutes per recording; individual dialogue lines are easier to review than long combined sessions.

If you replace a recording, regenerate or carefully recheck its cues. A line with the same words can have different timing in a new take. Each translated recording needs its own timing too. Saving the old English track onto a longer translated performance will not keep the mouth aligned.

> Check your work. Play the whole spoken line after saving. Check the rest position before and after speech, and try Portrait Preview if the dialogue uses a portrait.

22. Fonts, modern and bitmap

Import a font as Kind **font**. The preview shows sample text so you can check that the file opens. Assigning the font to a GUI or other text system is a later step; import does not replace every font in the game.

Modern fonts

The picker offers TTF, OTF, WOFF and WOFF2. Start with `GuideFont.ttf` for a simple test. Apply the asset and inspect the sample at the sizes you expect to use in the game.

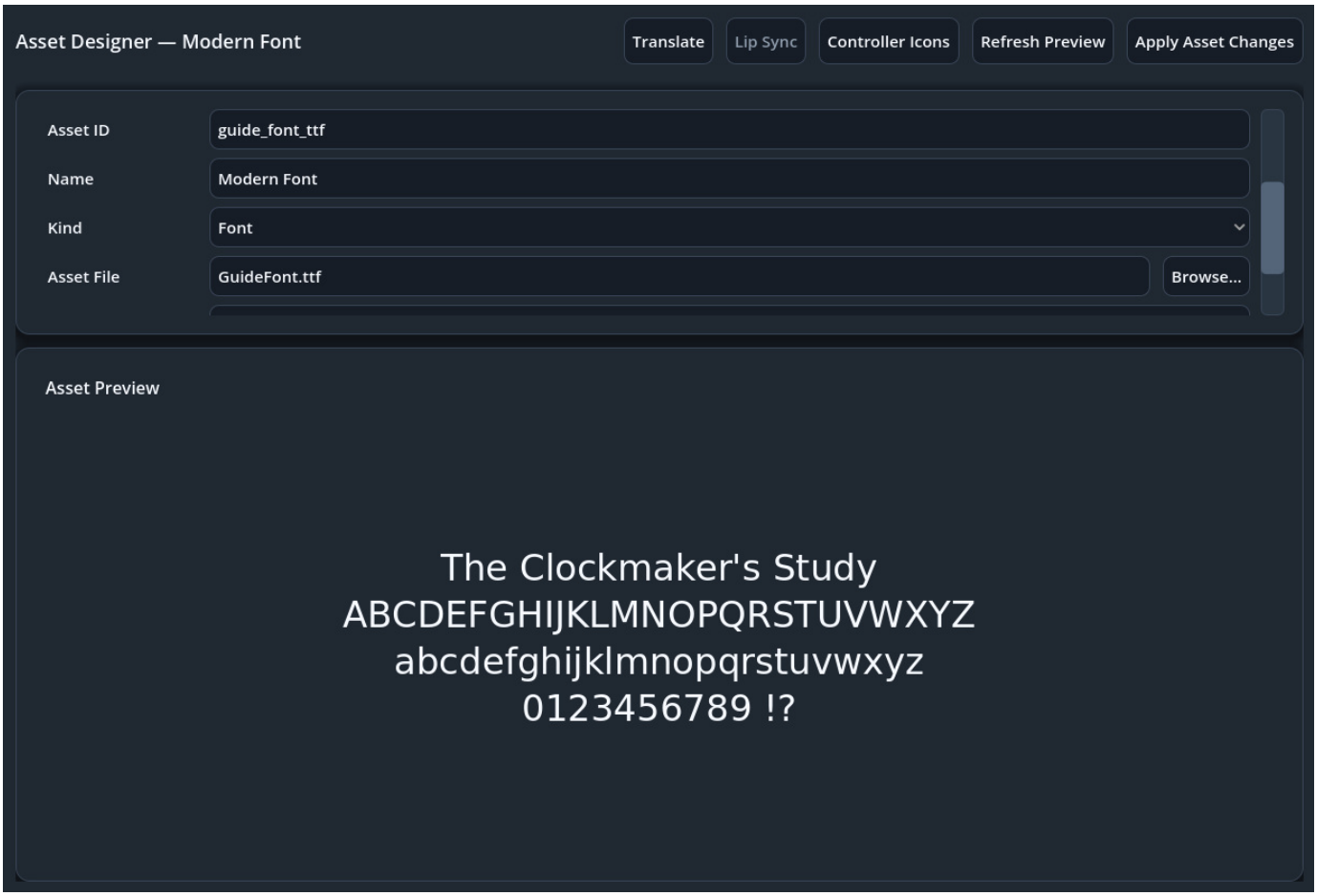


Figure 45. A modern font in the importer. The Latin sample proves that those displayed glyphs work, not that every language is covered.

Check the actual translated text, including accented letters and any non-Latin scripts. A clean English sample does not prove that a font contains the glyphs needed by every language.

Sierra AGI and SCI

The importer offers AGI binary font and SCI font routes. A standard AGI 8-by-8 set has 256 glyphs in 2,048 bytes. SCI resources have their own supported structure; an arbitrary file ending in `.fon` is not automatically a valid SCI font.

These supported legacy fonts are converted into a bitmap-font description and image atlas, while retaining the source. Keep that generated set together when moving the project.

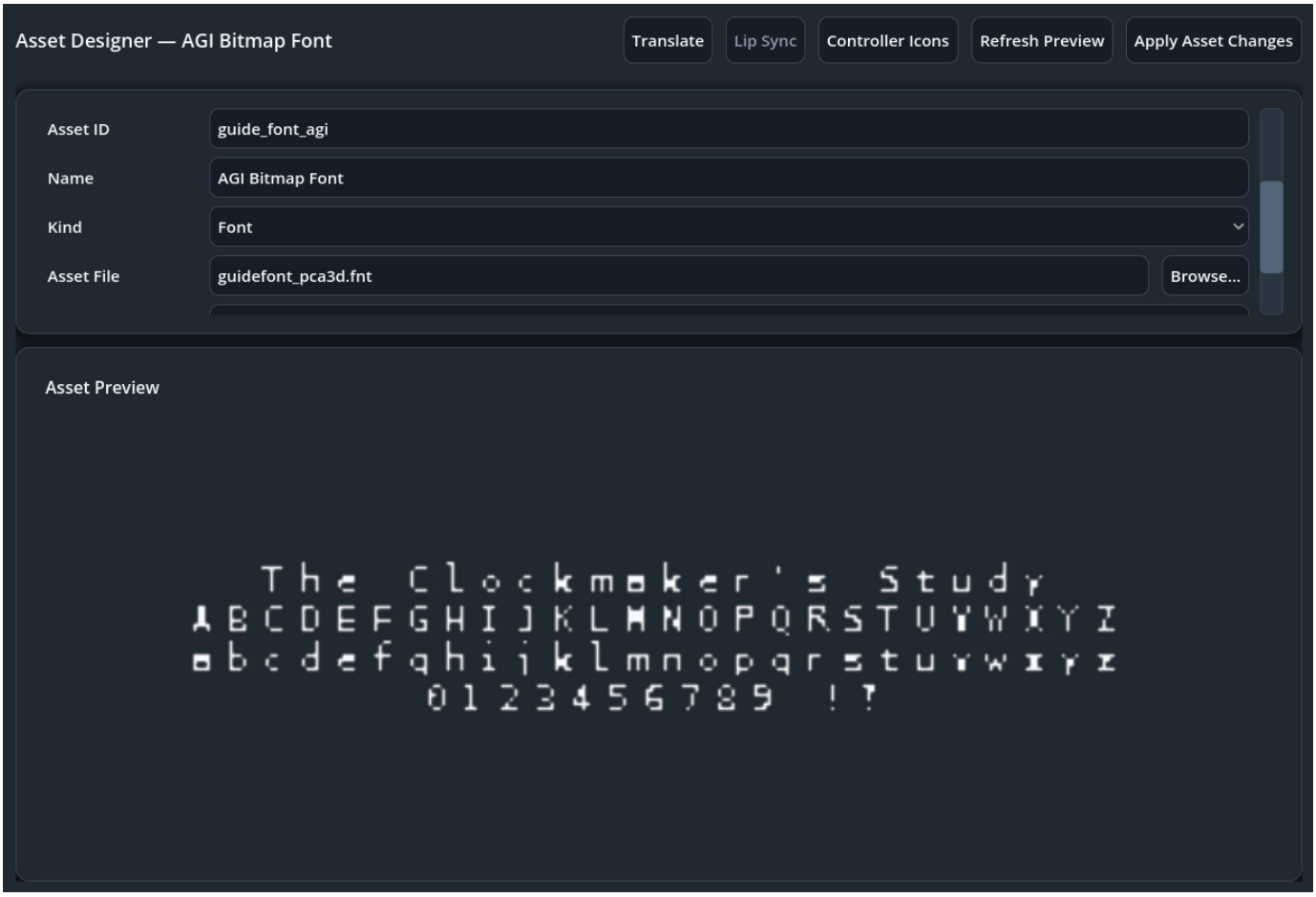


Figure 46. The AGI example after conversion. The source's small bitmap design is part of its appearance.

SCUMM charsets

Supported SCUMM charset files can be imported through the SCUMM font filter. The reader handles supported 1-, 2-, 4- and 8-bit data. Try `GuideFont.charset`, apply it and inspect the text preview.

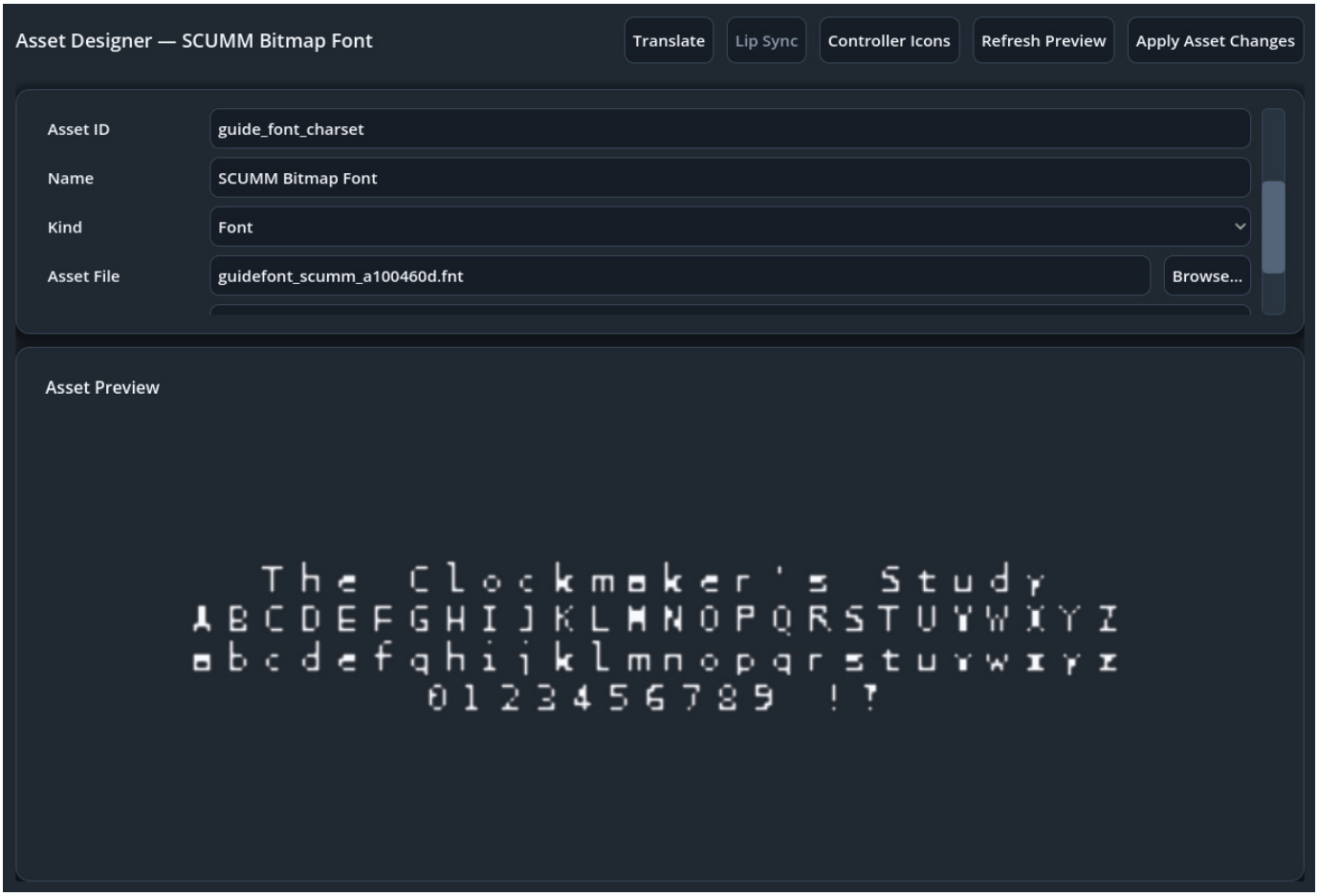


Figure 47. The original practice SCUMM charset converted for use as a font.

AngelCode bitmap fonts

The `.fnt` or `.font` description usually refers to one or more atlas pictures. Those images are part of the font. The standard file-copy route does not guarantee that an arbitrary external font's companions will all be collected, so check the atlas references and the resulting preview before relying on it.

The extension `.fnt` can also appear on other font formats. Use the correct filter and a genuine supported file; renaming an unrelated font is not conversion. Keep any required atlas images at the paths expected by the description, or prepare a self-consistent font package in the source project.

Keep the font's redistribution terms. The practice modern font is supplied with its DejaVu notices. Those notices belong with copies of the font, including a release that distributes it.

23. Video

Use Kind **video** for a movie file. The picker includes MP4, M4V, AVI, MPEG/MPG, WebM, MOV, MKV, FLV and OGV. A container extension does not tell you every codec inside it, so a file appearing in the picker is not proof that its particular encoding can play.

1. Add `guide_video`, named **Guide Test Clip**.
2. Choose video and Browse for `Guide_Test_Clip.ogv`.
3. Click **Play Video** and check the short moving test pattern.
4. Use Stop or the seek slider as needed.
5. Apply and save.

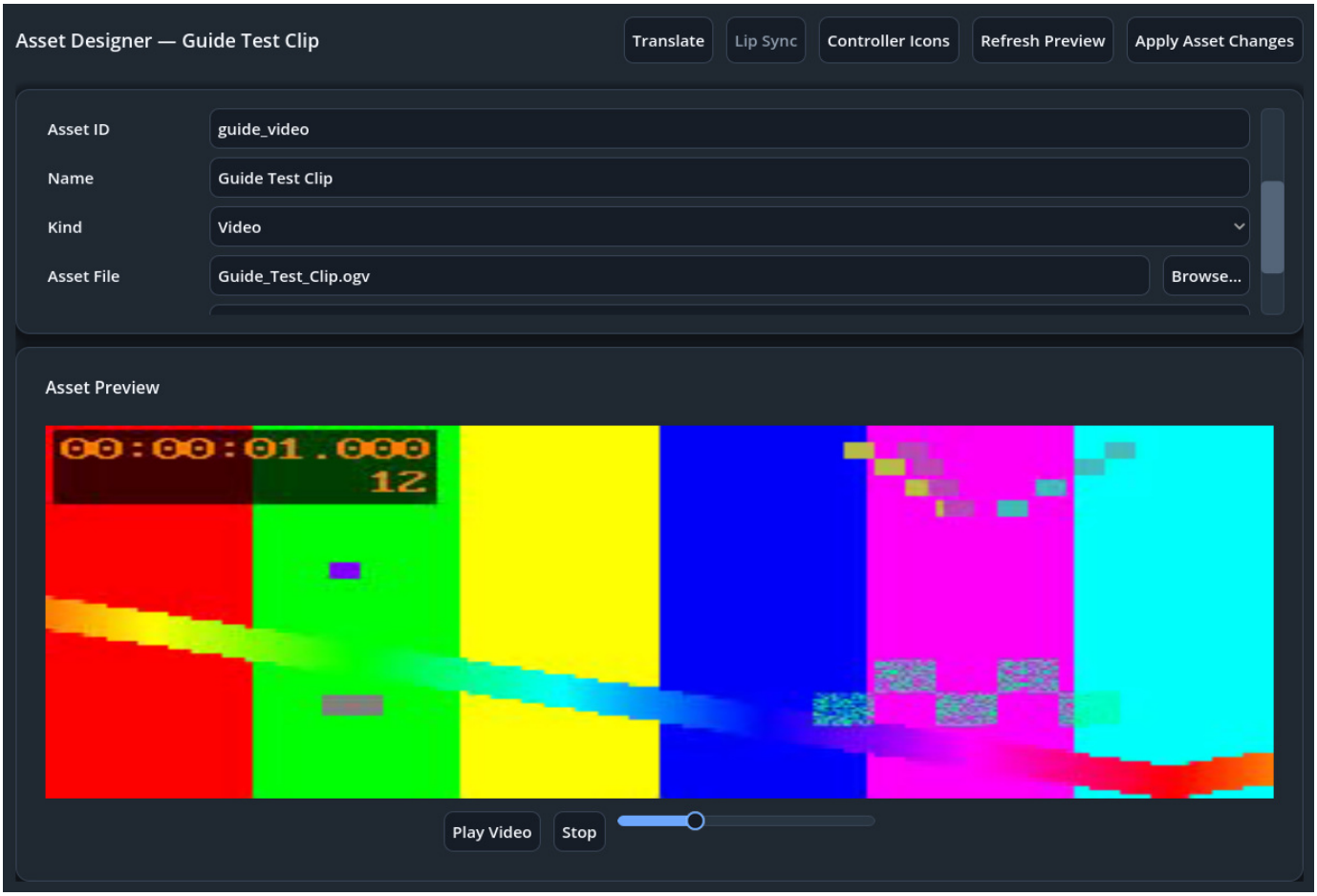


Figure 48. The practice Ogg Theora clip playing in Asset Preview.

The native Ogg Theora route provides a useful baseline. Other supported containers depend on the available decoder and compatible codecs. If a file fails, try a known-good encoding rather than only changing the extension. The editor's guidance includes H.264/AAC for MP4 and suitable MPEG-4 Part 2 or H.264 with MP3/AAC for AVI.

Seeking changes the stream position; it does not have the same forced-play behaviour as the audio seek slider. Preview controls also do not construct a cutscene. Assign the asset in the FMV or cutscene workflow to decide when it starts, how it can be skipped, and what happens afterwards.

Test the actual distributed game on each target platform. A decoder present in an authoring installation and a video filter listing are not substitutes for that test.

24. Godot and GLSL shaders

Choose Kind **shader** to import a native Godot shader or a supported fragment-shader subset. Importing stores a reusable asset; it does not apply the effect to every model or to the screen.

Native .gdshader

The supported native input starts with either `shader_type spatial;` for 3D materials or `shader_type canvas_item;` for 2D/GUI use. In the current importer, keep that declaration at the start of the file. A leading comment before it can prevent the simple initial check from recognising it.

Try `guide-material.gdshader` as a Shader asset. Its spatial shader is intended for a 3D material, not a screen overlay.

Convertible GLSL fragments

The picker offers `.glsl`, `.frag` and `.fs`. This is a converter for a limited fragment subset, not a complete GLSL engine. It expects an ordinary fragment `void main()` and supported declarations.

The practice `guide-tint.frag` contains the marker `// pca3d:canvas`, which asks for a 2D canvas shader. Without that marker the conversion defaults to a spatial shader. The converter writes a `.pca.gdshader` result and retains the imported source.

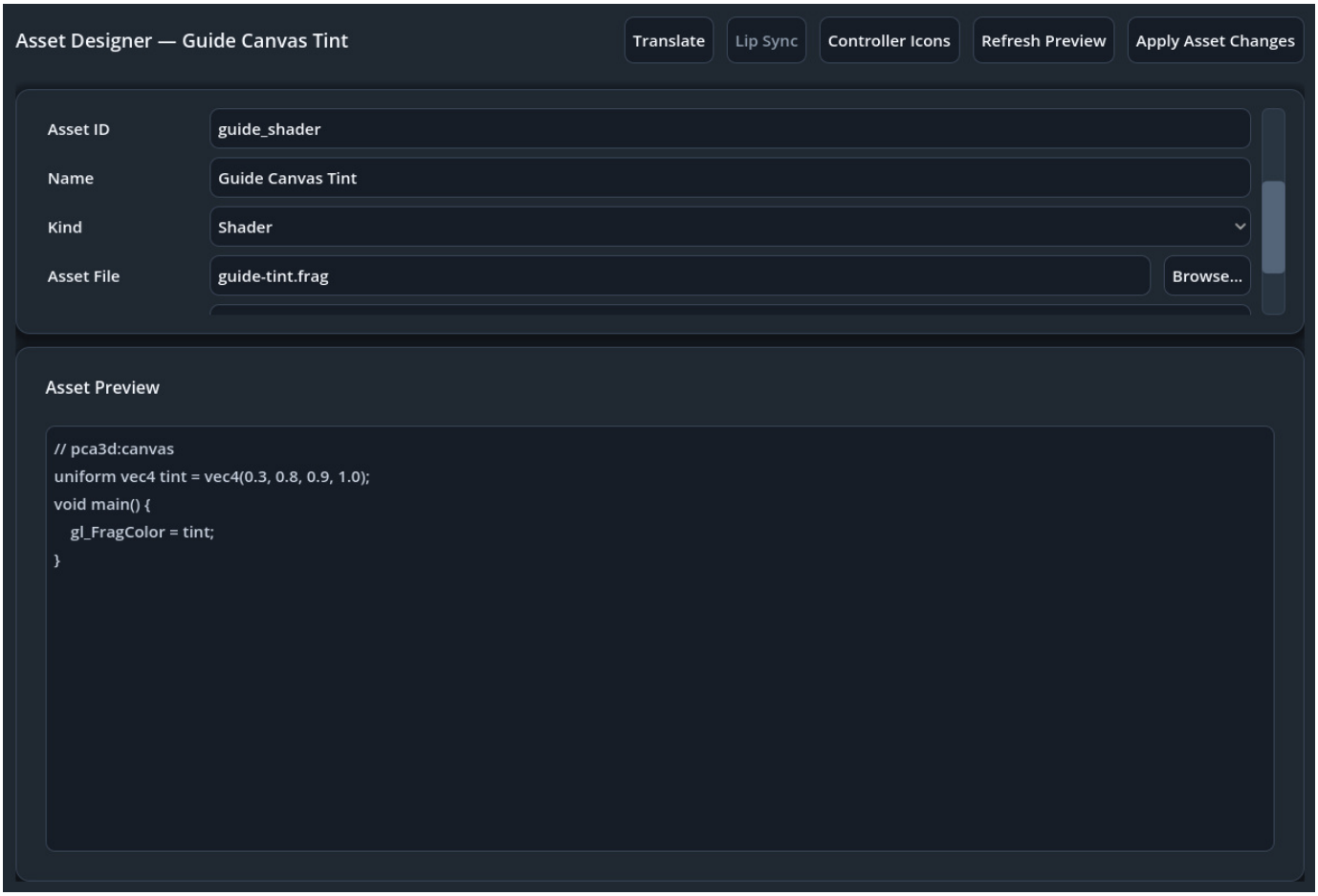


Figure 49. The Shader preview displays source text. It is not a live shader sphere or proof of how an effect looks on a particular object.

What the conversion does

Supported fragment UV input is mapped to the Godot UV input; supported fragment output is adapted to Godot's output path. Familiar fragment texture calls and fragment coordinates are translated where supported. Version and precision directives can be ignored as part of that conversion.

Vertex-shader code using `gl_Position`, includes, Shadertoy `mainImage`, unsupported declarations, `samplerCube` and the converter's unsupported `discard` case need a proper rewrite rather than a different file suffix.

Source situation	Best next step
A small compatible fragment effect	Import and inspect the converted shader, then assign it to a suitable target.
A native Godot 3D material shader	Use the spatial shader route and apply it through the object's material/shader settings.
A 2D or GUI shader	Use canvas_item or the converter's canvas marker, then choose it in the matching 2D/GUI system.

Source situation	Best next step
A complete multi-pass shader package	Rebuild it through the supported effect/material workflow; one imported fragment is not the whole pipeline.
An unsupported GLSL error	Read the exact message and simplify or translate the unsupported feature.

Check both compilation and appearance after assignment. A text preview alone cannot tell you whether the material looks correct, and standard texture overrides do not take over the internal texture setup of every custom shader material.

25. Data and Other files

Data

Use **data** for JSON, TXT, CSV, XML, CFG or INI content. The intended preview is a read-only text view. There is a preview limitation in this release: a plain CSV or other non-Godot text resource can be copied successfully, yet the preview reports that Godot cannot load the selected resource. The screenshot below records that result rather than hiding it.

1. Add `guide_data`, named **Guide Data Table**.
2. Choose data and Browse for `guide-data.csv`.
3. Read the status. In 0.86.3.4 this CSV can show the resource-load message described above.
4. Apply and save, then check the copied source and the consumer that will read it. Use a normal text editor to inspect the CSV while this preview issue remains.

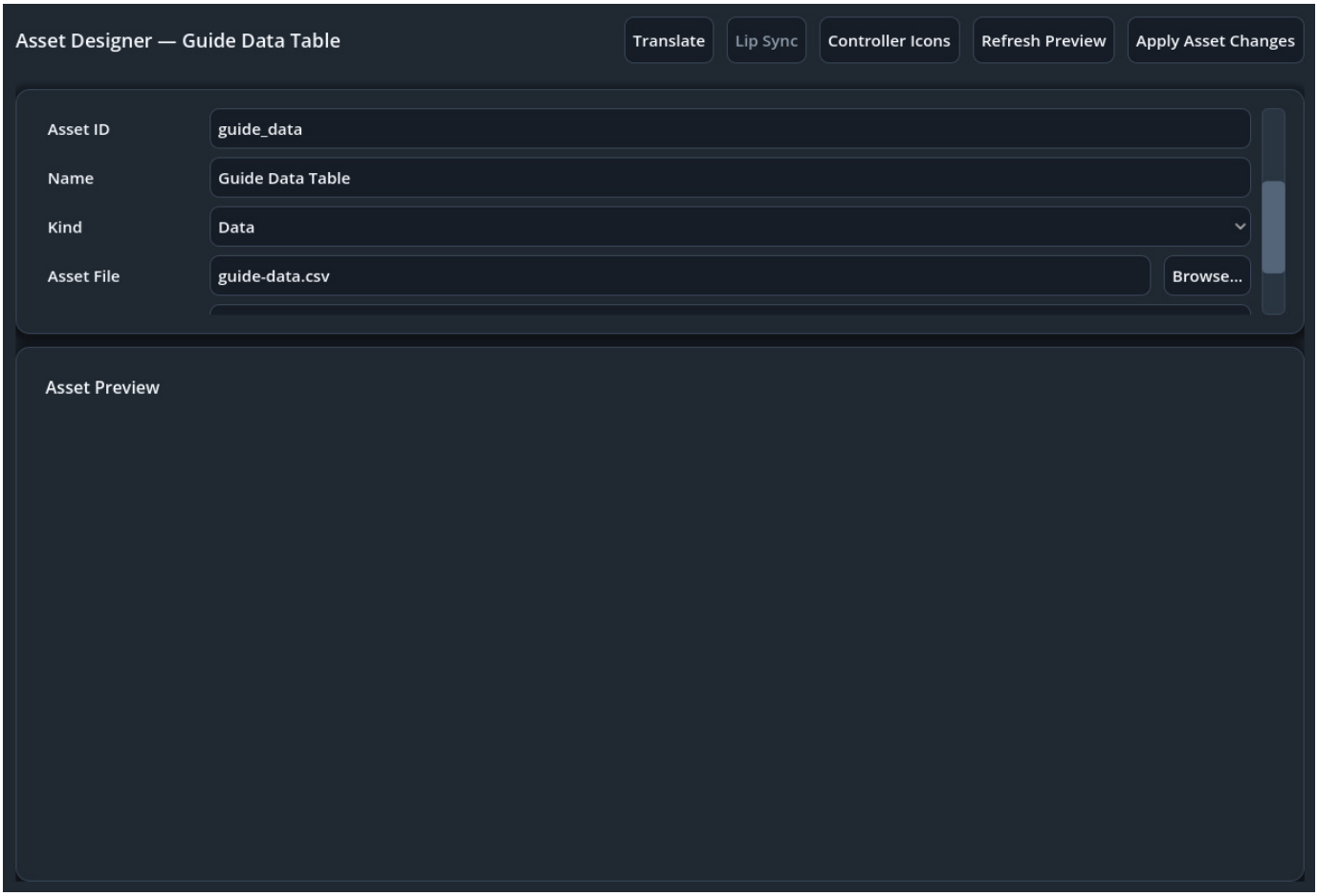


Figure 50. The CSV was copied, but its preview stays empty in this release because a resource-load check fails before the text view. The status line can also be too short to display its message.

When the text-preview route is reached, a very large text file stops displaying after 200,000 characters and marks the preview as truncated. That limit applies to what the preview displays, not to the source file's actual contents.

The importer does not execute a CSV, run a JSON file as a script, or automatically interpret every column. A game system or your script needs to read the file and decide what its values mean.

Other

Choose **other** when you need a file kept as an asset but there is no specialised importer for its contents. Try `guide-package.bin` for the example.

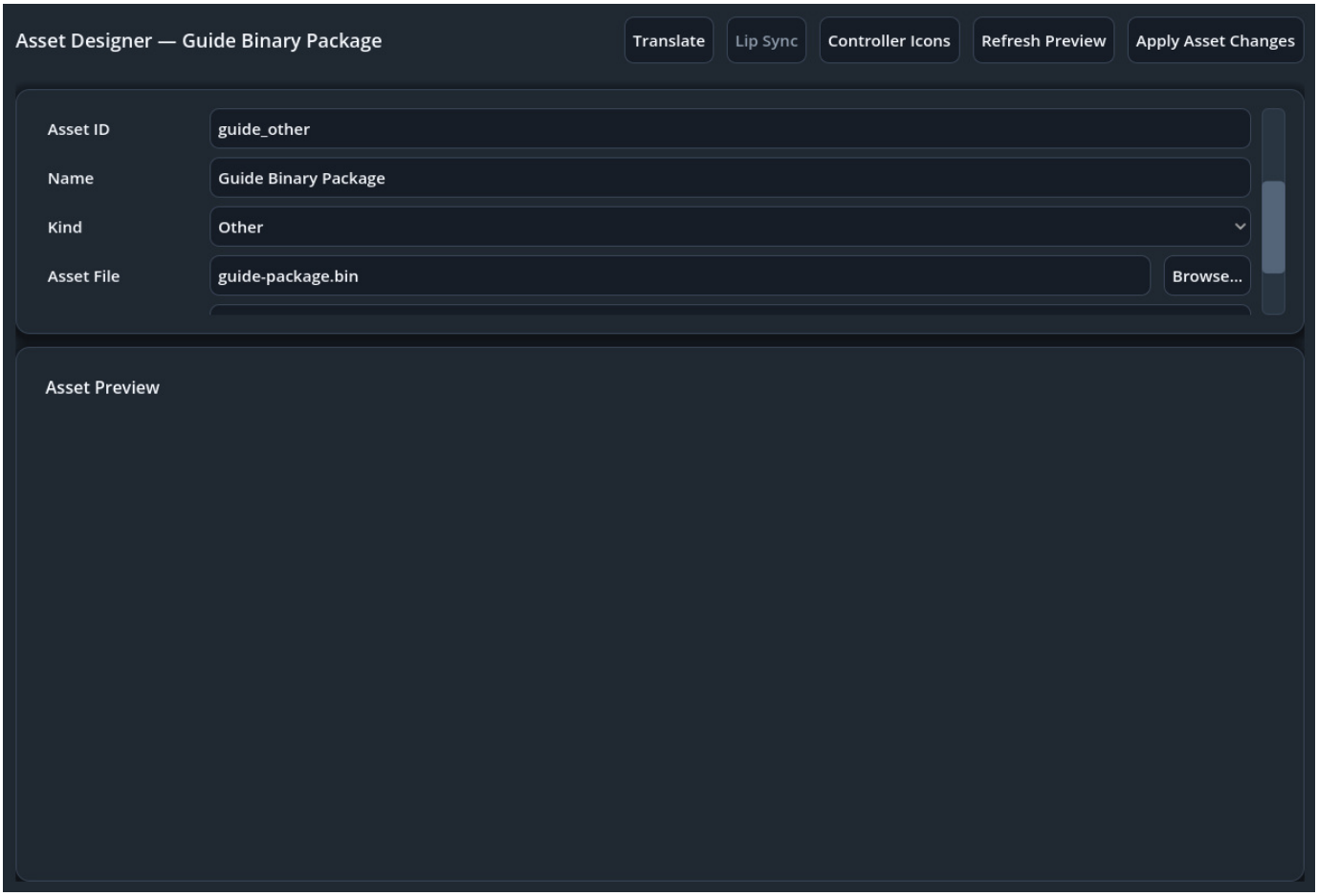


Figure 51. The generic binary file is stored, but the same resource-load check leaves its preview empty in this release.

A generic file can show the same resource-load message even after a successful copy. Check the imported file and its intended reader. Other is storage and reference, not a universal decoder. It does not add support for an unknown model format or install a plugin that can play a proprietary file. If the game needs to understand that format, a compatible reader is still required.

26. Controller and keyboard icons

The Controller Icons button opens a ready-made image browser. This is a convenient way to add pictures of buttons, sticks and keys to instructions or GUI prompts.

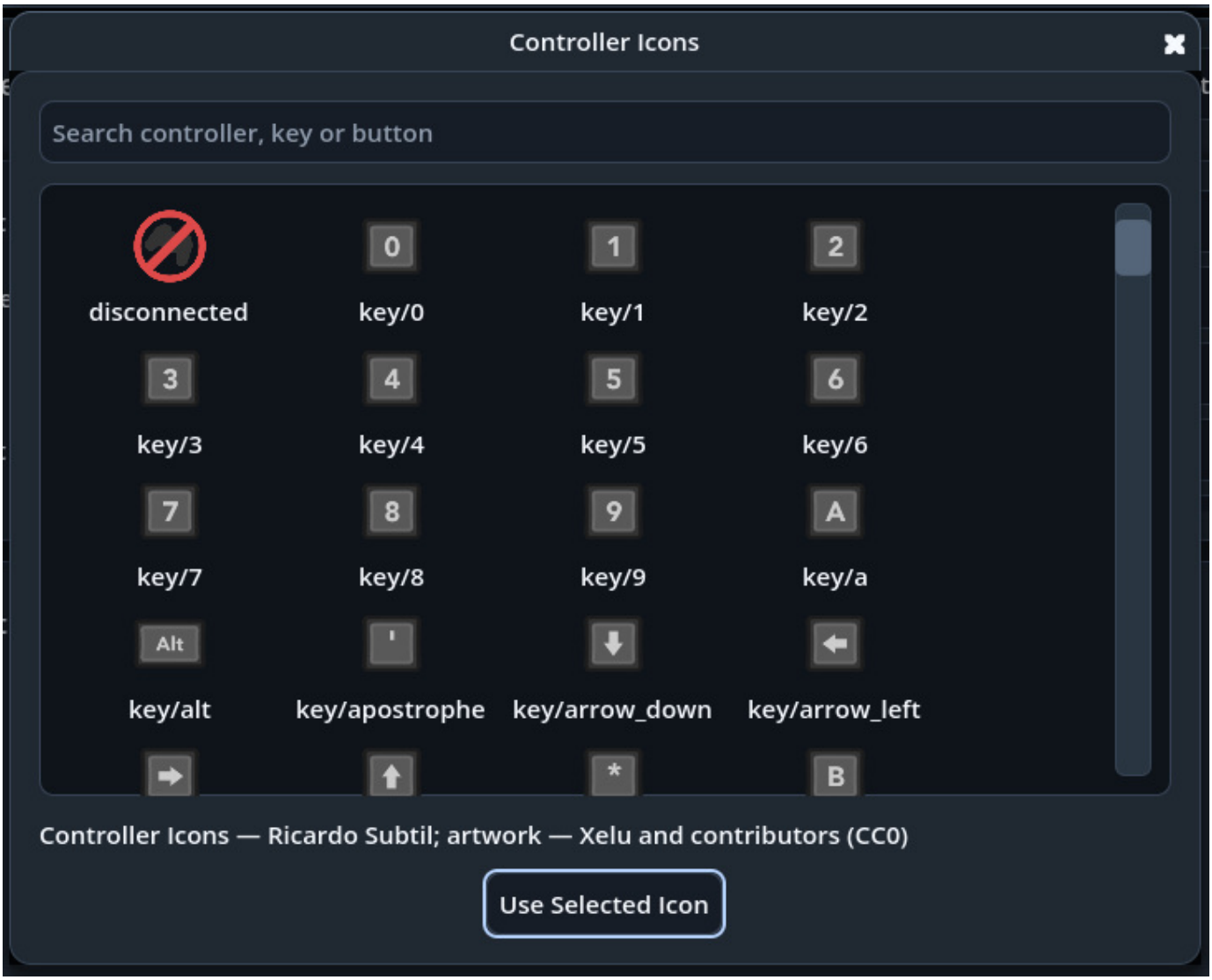


Figure 52. The built-in icon gallery. Icons are organised by their family and relative filename.

1. Click **Controller Icons** in the Asset Designer.
2. Type a family, key or button term in Search, for example xbox.
3. Select an icon and click **Use Selected Icon**, or double-click it.
4. The selection is imported through the Image route. Give the asset a clear ID and name.
5. Apply Asset Changes and save.
6. Choose that Image asset in the GUI element or prompt that should display it.

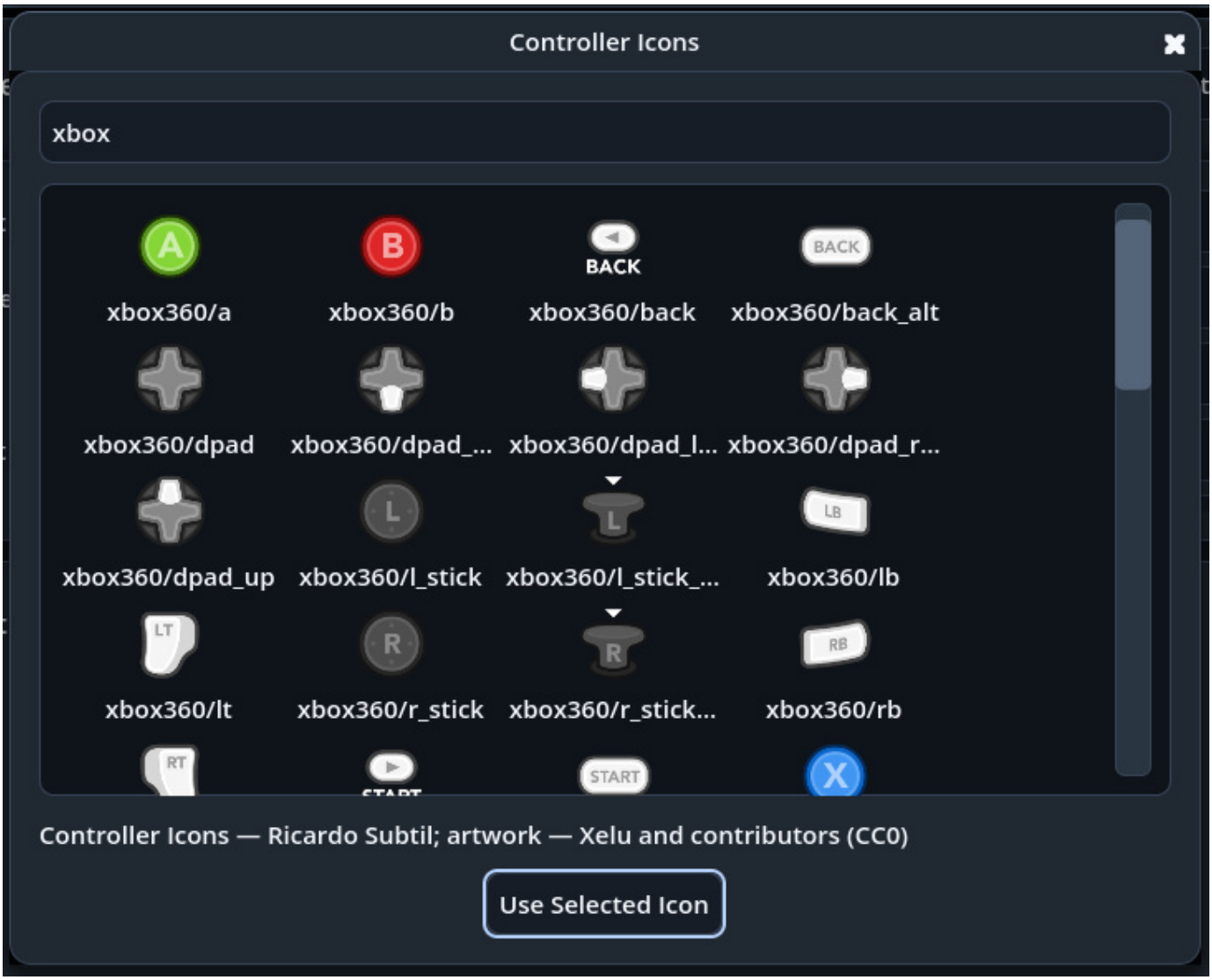


Figure 53. Searching for xbox narrows the gallery. A prompt icon is an ordinary reusable Image asset after selection.

Search is a case-insensitive text match against the icon entries. Try the controller family or a shorter button name if a very specific phrase finds nothing.

Importing a button image does not create an input binding, remap an action or turn on automatic controller-family switching. Configure the game input and adaptive prompt behaviour separately. The picture tells the player which button to use; the input system decides what that button does.

The bundled icon workflow credits Ricardo Subtil's Controller Icons project and Xelu with contributing artists for the underlying CC0 prompt artwork. Keep the distribution's notices when releasing the supplied assets.

27. Language-specific asset replacements

Translate is for the game's content. It is separate from choosing the language used by the editor's menus.

Select an asset and click **Translate** to open game translations with that asset as the starting context. A translated replacement might be a sign image with different lettering or a voice recording in another language.

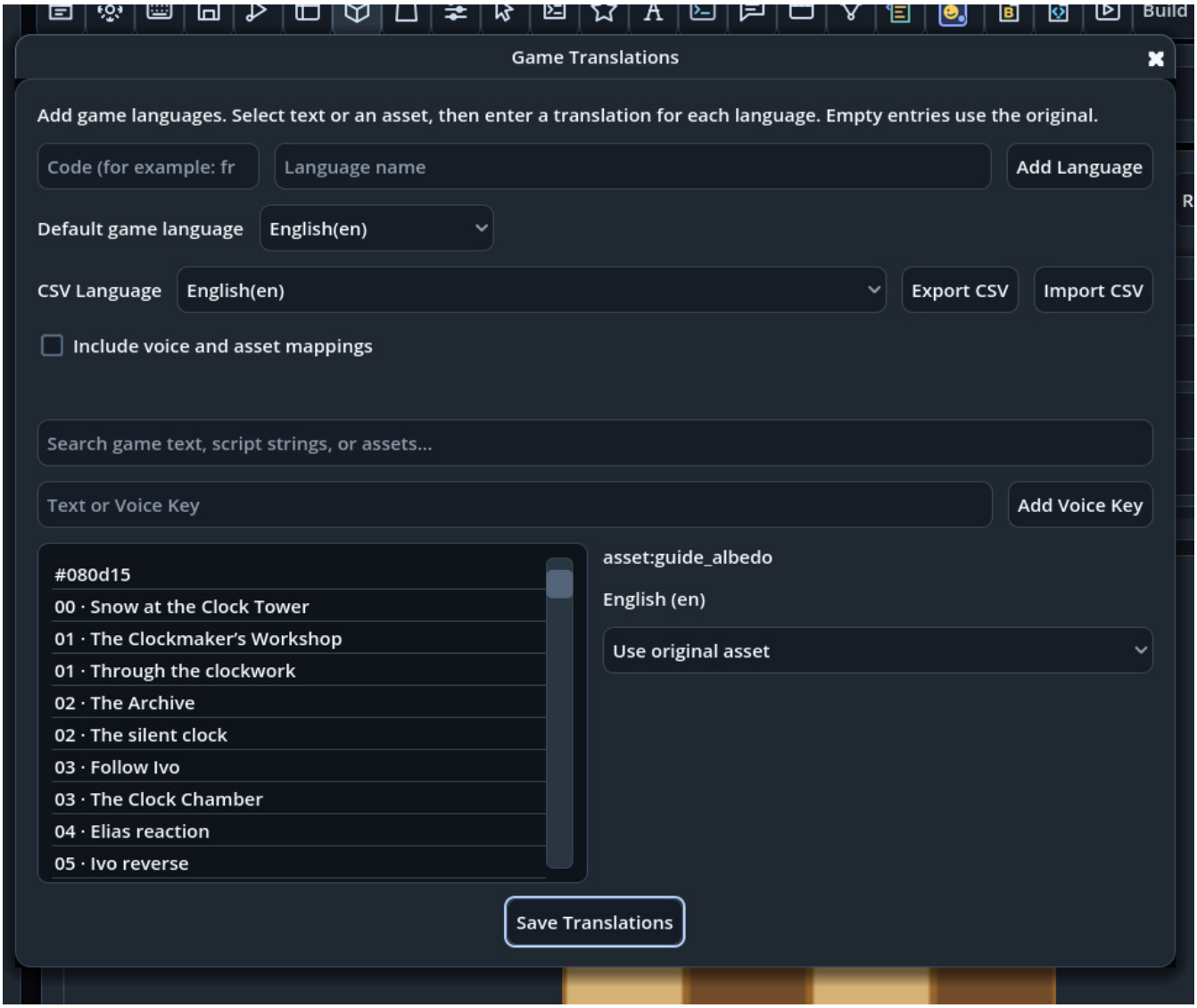


Figure 54. Game Translations opened for an asset. The precise replacement should suit the same job as the original.

1. Make sure the project has the intended game languages configured.
2. Import the replacement media as an appropriate asset.
3. Open Translate for the original asset and choose the target game language.

4. Assign the replacement using the translation editor's asset controls.
5. Click **Save Translations**, then save the project.
6. Run the game in that language and check the actual result.

Use a compatible replacement. A sprite sheet should retain the layout its consumer expects. A translated sign should fit its GUI or model surface. A voice replacement needs its own lip-sync timing, even when its subtitle is a translation of the same sentence.

An empty translation normally lets the original content act as the fallback. Test that fallback rather than assuming a half-finished language will always be visually identical. CSV translation exchange can help a team coordinate text and supported mappings, but exchanging a CSV does not send all of the actual voice or image files to another translator.

The translation controls

Control	What to use it for
Language code, language name and Add language	Add a game language using its code and a readable name.
Default game language	Choose the original/default language of the game.
Search	Find source text, script strings or asset entries in the catalogue.
Text or Voice Key and Add Voice Key	Add a key for text/voice mapping when the workflow needs one. This does not import a recording.
Catalogue list	Choose the source text or asset to translate.
Per-language asset selector	Choose an imported asset of the same Kind as the source.
Use original asset	Remove that language's replacement and use the original asset.
Default Voice Audio / Voice Audio	For a text entry, assign the default or language-specific recording.
Save Translations	Commit the window's translation work to the open project. Save Project afterwards.

Sharing translation work by CSV

Choose the intended language in **CSV Language** before exporting or importing. Turn on **Include voice and asset mappings** when the exchange needs those references as well as text. Leave it off for a simpler text-only handoff.

1. Select a target language and click **Export CSV**. Keep an untouched copy of the exported file.
2. Give a working copy to the translator. Preserve the headings and source identifiers; edit the translation values. Use a spreadsheet's proper CSV/TSV save operation so commas and line breaks are quoted correctly.
3. Back in the editor, choose the matching CSV Language and click **Import CSV**.

4. In **Preview Translation Import**, compare Source, Current Translation, Imported Translation and Import Status. Read the counts for changes, skipped rows, conflicts and invalid rows.
5. Leave **Overwrite conflicting translations** off unless the incoming work should replace existing conflicting values. Review the changed plan if you turn it on.
6. Click **Apply Changes** in that preview when the proposed changes are the ones you want.
7. Click **Save Translations**, then Save Project. Test the language in the game.

Different people can work on different language exports. For two people editing the same language, divide the work clearly and review conflicts on import. A mapping refers to an asset ID already available in the project; the CSV does not contain the actual image, model or recording. Import new media before trying to assign its ID through a mapping.

28. Putting the imported asset to work

Once an import is checked and saved, choose it in the designer that needs it. This is where the difference between importing media and setting up behaviour becomes important.

Imported asset	Where it is commonly used	What still needs setting up
Static 3D model	Scene Designer object	Placement, scale, material choices, collider and interactions.
Animated character model	Character Designer, then a scene character	Model assignment, state-to-clip mappings, movement, mouth targets and scene placement.
Image	GUI Designer, inventory or another image field	Fit, size, state images and any related gameplay assignment.
Sprite Sheet or Animation	Compatible 2D character, object or GUI source	Which animation/state uses it and when it plays.
Audio	Sound/music action, dialogue or scene audio	Playback timing, volume, looping and the correct speaker.
Font	Compatible text/font selector	Actual size, layout and language coverage in the consuming GUI.
Video	FMV or cutscene workflow	Start, skip, resume and completion behaviour.
Shader	Matching material, GUI or screen-effect system	Correct shader mode, target, parameters and rendering compatibility.
Data / Other	A compatible game system or script	Parsing and using the file's contents.

A model is visible in the importer but not on a character

Check the Character definition's model asset assignment first. Then check that the scene instance is using that definition rather than a placeholder. Set the animation states to real imported clips. Inspect visual scale, orientation and position. A good import does not make these assignments for you.

A reflection looks like a shiny highlight

A roughness or metallic map changes material response. A character visible in a mirror needs an appropriate reflection system and scene setup. Do not try to solve a missing mirror image merely by importing the same roughness texture again.

An attach point is not a collider

Moving a grip or changing Contact Radius does not shrink a door's walking obstacle. Configure the scene object's collider/blocking dimensions in the relevant scene settings. Asset import, event volumes, attach points and navigation obstacles each have separate jobs.

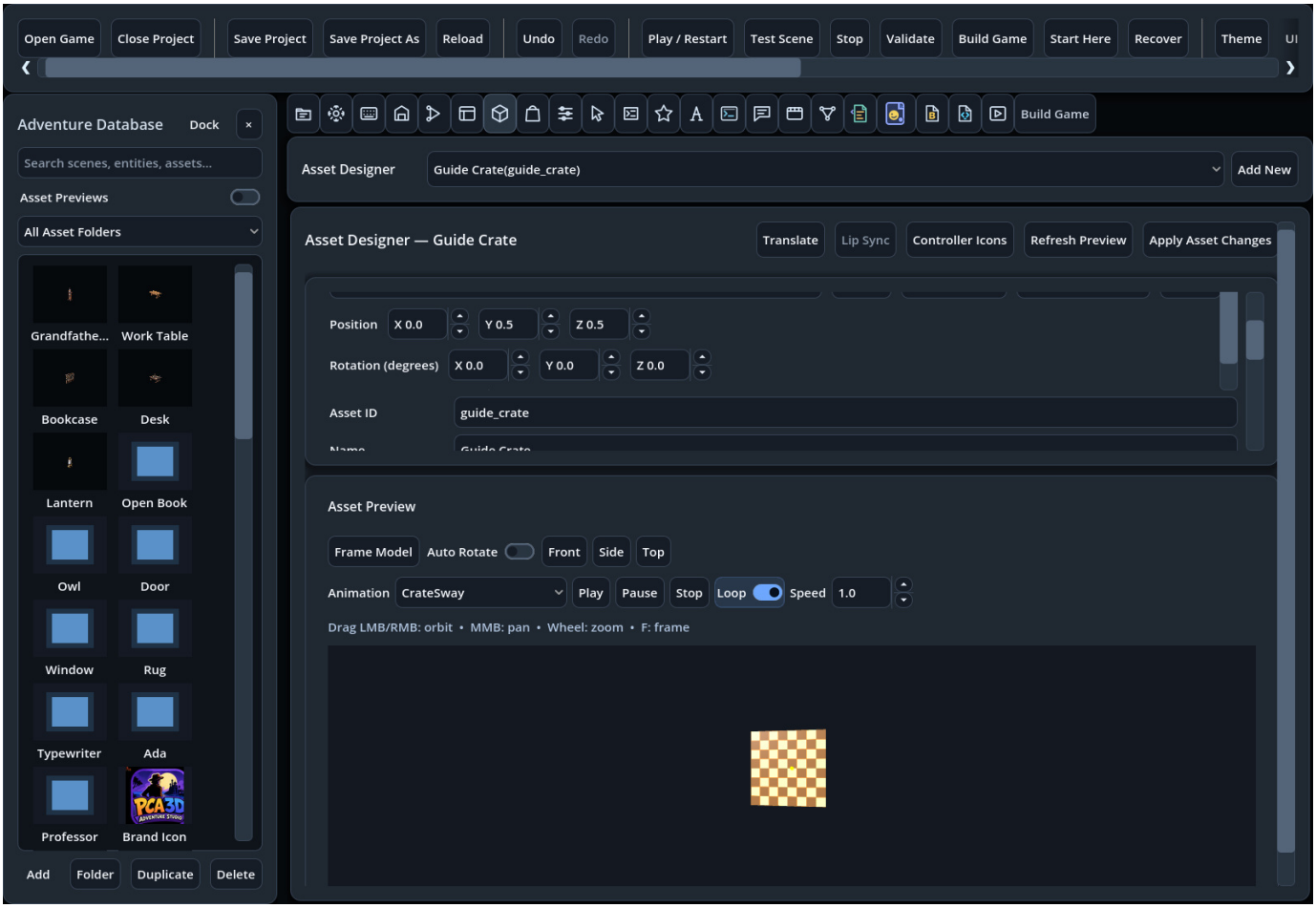


Figure 55. The asset gallery provides another way to browse imported content. Always check the selected record before replacing or deleting a shared file.

29. Reimporting and keeping projects organised

A dependable replacement routine

1. Save a checkpoint before replacing a model or a large set of frames.
2. Keep the asset ID if existing records should continue to use the replacement.
3. Browse for the revised source through the correct Kind.
4. Wait for conversion or dependency import to finish.
5. Check geometry, textures, clip names, scale and forward direction again.
6. Recheck attach points, character states and lip sync if the source affects them.
7. Apply Asset Changes, save, and test one actual gameplay use.

Reimporting can change clip names or the order of materials. A replacement that looks similar may still invalidate a state mapping, a texture override or a bone-associated attach point. The later check is part of the job.

Duplicate a record for a variant

Duplicate is useful when two asset records need different settings, such as two Krita clips or two material treatments of the same model. Give the duplicate a different ID and name, then apply it.

A duplicated record can share its original file paths. If you need the underlying source to be independently editable, make a separately named source copy and import that copy. Do not assume that replacing one shared physical image cannot affect another record pointing to it.

Keep source names useful

Use source filenames with meaning: `harbour_sign_en.png`, `harbour_sign_it.png`, `ivo_greeting_take02.wav`. Keep a complete author's source bundle outside the release output. Database folders help navigation, but they do not replace that source-file organisation.

Avoid casually changing IDs after scripts and records refer to them. Changing a visible Name is easier to understand than turning a working ID into an unrelated identifier. After an intentional reference change, validate and test the affected game path.

Backups and builds

Back up the working project document and its imported/custom media. Keep the supplied demos unchanged and work from copies. Keep built games in a separate output folder so that a cleanup does not remove your source.

Finally, build a small test game and run it from a clean output folder. Check representative model, animation, image, sound, video, font and shader assets. A preview inside a source installation can see files that an incomplete distributable might be missing.

30. When something goes wrong

Use the symptom to choose the next check. Changing every setting at once usually makes the original problem harder to find.

Symptom	Check first	Then try
File is missing from the picker	The selected Kind and current filter.	Choose the correct Kind. Use All Files only when you know the real supported format.
Preview says it is waiting for import	Current operation/status and whether preparation has finished.	Wait, then Refresh Preview. Repeated Browse clicks are not a speed boost.
Model is grey or textures are missing	Source companions and relative paths.	Reimport the complete bundle; compare with a self-contained GLB.
Model is tiny in the room but large in preview	Frame Model has fitted it to the preview.	Check Unit Scale or the scene transform against a known-size object.
Character faces sideways while walking	Model Forward Axis and character visual rotation.	Correct the source orientation, apply, and test the character again.
No animation clips	Whether the source contains exported animation.	Check export Actions/NLA or supported source format; Refresh Preview after conversion.
Blender cannot be found	Blender installation or executable path.	Configure the full executable path, or import an already exported GLB.
OBJ lost multiple materials	The single-surface OBJ conversion route.	Use a suitable GLB for a richer material or animated scene.
MDL is rejected	The real MDL format and version.	Use supported Quake/GoldSrc data or export another supported format.
Animation jumps or flashes blank	Frame registration, slicing and frame count.	Pause and step; check canvas sizes and Last Row Count.
Frames are in the wrong order	Filenames and mixed prefixes.	Use one consistent numeric sequence per asset and reimport.
KRA is still instead of animated	Kind chosen before Browse.	Use Animation, select one KRA, and choose its clip.
KRA differs from Krita	Supported paint layers, masks and special effects.	Export finished PNG frames and use the image-sequence route.
Texture dropdown has no new picture	Image Kind and whether that record was applied.	Apply the Image asset, return to the model, and inspect the material selector.

Symptom	Check first	Then try
Texture override seems ignored	Master checkbox, selected material and custom shader use.	Enable the override and choose the right standard material; custom shaders keep their setup.
Lip Sync button is disabled	The saved asset Kind.	Apply an Audio record first.
Mouth does not move in preview	Character mouth mapping and selected preview character.	Inspect Mouth Targets and configure the real model/portrait mapping.
New voice take drifts out of sync	Old cues belong to the previous recording.	Generate or edit the replacement recording's own track.
Font shows boxes for translated text	Glyph coverage and source encoding.	Use a font with the required glyphs and test actual localised strings.
Bitmap font has no glyph images	Required atlas files and their references.	Keep or repair the complete supported font package.
Video opens but will not play	Container, codec and decoder support.	Test the OGV example or transcode to a known supported encoding.
Shader fails conversion	Shader mode, required first declaration or unsupported GLSL.	Read the exact error and prepare a supported native/fragment shader.
Data preview is cut short	The 200,000-character preview limit.	Read the original file elsewhere; the preview limit did not trim the source.
Data/Other says Godot cannot load the resource	The current preview gate can reject plain text or binary files after copying them.	Verify the copied source and reader. This guide does not change the engine to hide the issue.
Generic file has no visible player	Other has no specialised decoder.	Use a compatible reader or convert to an appropriate supported format.
Works in editor, missing in built game	The packaged files and target runtime support.	Test a clean build and include the required imported media/dependencies.

If you need to report an import problem, include the PCA 3D version, operating system, chosen Kind, actual file format/version, exact status/error text and a small reproducible example you can legally share. A screenshot helps show the settings, but a picture of an error cannot reveal missing companion files by itself.

R1. File-format desk reference

This table describes the current importer routes. A listed extension is not a promise that every possible variant, exporter or codec using it is supported.

Kind	Picker formats	Important detail
Model, modern	GLB, glTF, FBX, OBJ, DAE, BLEND	Keep dependencies; BLEND needs Blender for conversion; OBJ becomes a single-surface mesh.
Model, source	MD2, MDL, MD3, VOX, QB, LWO, X, BBMODEL	Format/version restrictions and extra scale/FPS/axis settings apply.
Image	PNG, JPG/JPEG, WEBP, SVG, KRA	KRA supplies its saved merged picture.
Sprite Sheet	PNG, JPG/JPEG, WEBP, SVG, KRA	Requires equal-cell slicing information.
Animation	PNG, JPG/JPEG, WEBP, BMP, TGA; one KRA	Multi-select image frames, or use the separate Krita clip route.
Cubism	MODEL3.JSON	A complete bundle and compatible animation extension are separate requirements.
Audio	OGG, WAV, MP3	Use Audio for voice before opening Lip Sync.
Font, modern	TTF, OTF, WOFF, WOFF2	Check actual language glyphs.
Font, bitmap	AGIFONT/AGI/AGIFONT/BIN, FONT.* / SCI / FON, FNT/FONT, CHAR/CHARSET/SCUMM	The actual encoding matters. Preserve converted or required atlas files.
Video	MP4, M4V, AVI, MPEG, MPG, WEBM, MOV, MKV, FLV, OGV	Containers depend on compatible codecs/decoder availability.
Shader	GLSL, FRAG, FS, GDSHADER	Limited fragment conversion or supported native Godot shader.
Data	JSON, TXT, CSV, XML, CFG, INI; All Files	Files can import despite the current plain-text preview gate. A consumer interprets data.
Other	All Files	Generic file reference, without an automatic player.

R2. Settings and destination reference

Use this section to look up a range or understand the imported copies. Do not manually rearrange managed files while the editor is importing them.

Numeric controls

Control	Allowed range or unit	Reminder
Source-model Animation FPS	1-120 FPS	Source fallback/sampling; disabled for LWO.
Source-model Unit Scale	0.0001-100	Compare actual room size; preview framing hides size differences.
Texture Scale	0.01-100	Repetition, not physical model size.
Model preview Speed	0.05-8 times	Preview audition.
Sheet Columns / Rows	1-512	Grid count.
Last Row Count	0-512	Zero means a full final row; useful count is capped by Columns.
Sheet Frame Width / Height	1-4096 pixels	Used by Frame Size splitting.
Saved 2D Animation FPS	0.1-240 FPS	Upper form; Apply and Save.
Preview sequence FPS	0.1-240 FPS	Lower player; temporary audition.
Attach point rotation	Degrees	Point orientation, not scene placement.
Attach point position / rotation fields	-100,000 to 100,000; steps of 0.0001	Position is local; rotation is in degrees. Normalised image points normally stay within the picture.
Attach Contact Radius	0-100,000; steps of 0.001	3D game units or native 2D pixels, not collider size.
Lip-sync cue Start / End	Seconds, 0.01-second steps	Keep ordered, non-overlapping timings.
Audio/video seek	Seconds, 0.05-second steps	Player control; audio seeking starts playback.

Common managed locations

The paths below are relative to the source package's `res://` root. Model packaging and conversion can create a more specific per-asset subfolder.

Kind or special route	Typical location
Models	assets/models, usually a managed model subfolder with companions.

Kind or special route	Typical location
Image and Sprite Sheet	assets/images.
Separate frame Animation	assets/images/animations followed by a safe asset-ID folder.
Krita conversion	assets/krita followed by an asset/source-specific folder containing the saved source and baked pictures.
Live2D Cubism	assets/live2d with the copied bundle structure.
Audio	assets/audio.
Font	assets/fonts; converted legacy fonts also have generated description/atlas files.
Video	assets/video.
Shader	assets/shaders; converted result uses .pca.gdshader.
Data	assets/data.
Other	assets/other.
Imported controller icons	assets/controller_icons, preserving the icon-family path.

These locations explain why changing a database Folder Path does not move files on disk. They also explain why an adventure document and its imported media should be backed up together.

R3. Short recipes to keep beside the editor

These are reminders after the full lessons, not a replacement for checking the preview.

An inventory picture

Add an Image asset, Browse for the picture, apply and save. Choose that asset in the inventory item's image field. Check its transparency and appearance in the inventory GUI at the real display size.

A seven-frame sprite sheet

Choose Sprite Sheet. Browse for the practice lamp sheet. Grid: Columns 4, Rows 2, Last Row Count 3. Animation FPS 8. Apply. Pause and step through seven frames. Save.

A material variant of a model

Import replacement Image assets first. Duplicate the model record if you want a separate variant. Enable Replace Model Textures, select the material, choose the replacement channel and mapping, then apply and save. Inspect a room instance.

Two clips from one Krita source

Import the KRA as Animation. Choose Idle, apply and save. Duplicate the record with another ID, choose Pulse, apply and save. Check both records' names and selected clips before assigning them to states.

A spoken line with mouth movement

Import Audio, apply, open Lip Sync, choose the recording language, generate or import cues, select a mapped character, preview and correct, then Save Track and Save Project. Give each translated recording its own track.

A shared hand or grip target

Open the model asset. Add and name an attach point. Choose the correct animation scope and place it on the model or bone. Check position, rotation and contact radius. Apply and save. Later use the named point in the appropriate attachment/contact action.

R4. Glossary and credits

Use these terms when a status message or another designer refers back to an import setting.

Term	Plain meaning
Asset	A named project reference to imported media and its settings.
Asset ID	The stable identifier used to refer to that record.
Companion or dependency	Another file required by the selected source, such as a texture or glTF binary.
Atlas	One image containing many smaller pictures, such as font glyphs or sprite frames.
Clip	A named section of animation.
FPS	Frames per second; how quickly frames are sampled or played.
UV	The model's stored mapping from its surface to a 2D texture.
Triplanar	Texture projection from three directions.
Material	Surface rendering settings, including compatible textures or shader.
Bone	A named part of a skeleton used to animate a model.
Blend shape	A stored mesh deformation, often used for facial shapes.
Attach point	A named local position/orientation used by later attachment or contact logic.
Collider	Geometry used for physical or movement blocking; separate from attach-point contact radius.
Viseme / mouth cue	A timed mouth shape used while playing a voice recording.
Container and codec	The video file wrapper and the method used to encode its audio/picture data.
res://	The resource root of the source package.
Apply	Update the current record in the open working project.
Save Project	Write the adventure document after applying changes.

About the examples and screenshots

The screenshots are captures of PCA 3D Adventure Studio 0.86.3.4, including its supplied Clockmaker content and the small practice imports. They have been cropped or resized for the page; the controls shown are from the editor. No Live2D animation validation is claimed, and the practice archive contains no third-party Live2D model. The Data/Other preview limitation observed during capture is documented in Chapter 25; no engine code was changed for this guide.

The crate, lamp artwork, test models, sample chime and technical data examples were prepared for these lessons. The video is a technical test pattern. The TTF example uses DejaVu Sans and includes its redistribution notices. The bitmap examples are small training fixtures rather than files extracted from a commercial game.

The engine's controller-icon integration credits Ricardo Subtil and the Xelu/contributor CC0 artwork. Krita is the painting application's native file format used by that import route. Blender is used by the author-time .blend conversion. Rhubarb Lip Sync supplies the supported desktop recognition route. Consult the PCA 3D distribution's third-party notices for the software and media supplied with the engine.

This guide documents the controls and supported routes in the version named on the cover. When a later release changes a field or adds a format, use that release's notes alongside this book.