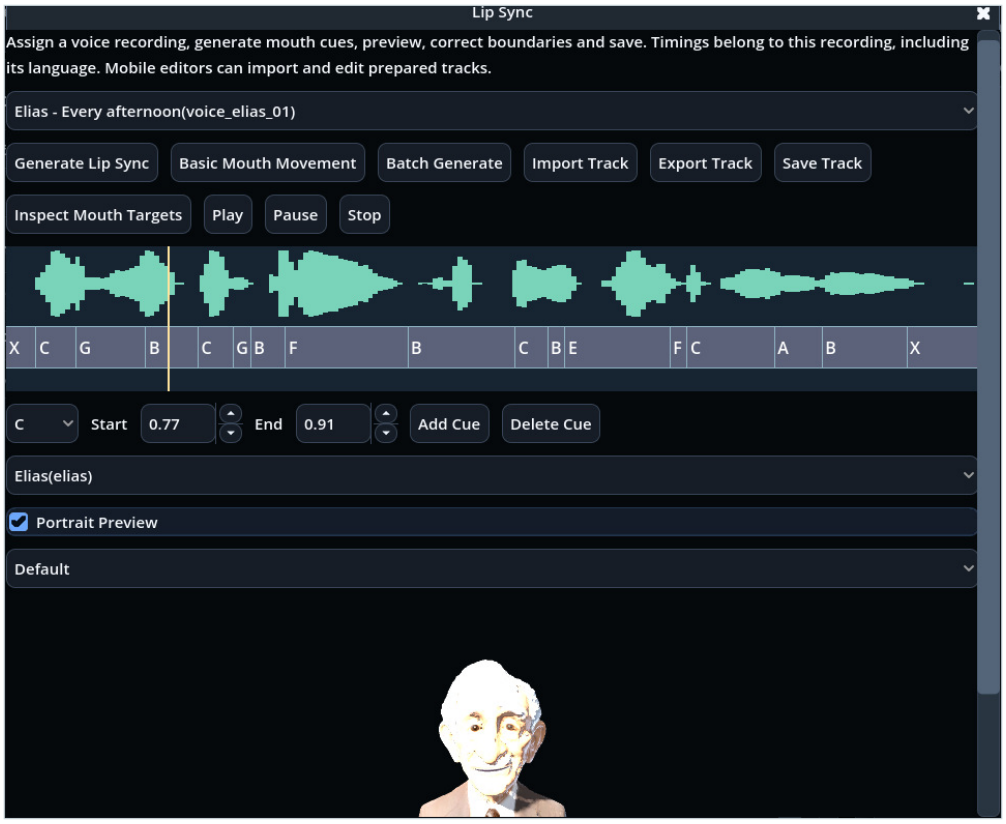
PCA 3D ADVENTURE STUDIO

The Character Designer

Portraits, animation and lip sync



From an imported model to a speaking character.

A practical, step-by-step guide to character definitions, models and sprites, animation states, player and NPC behavior, portraits, voice recordings, mouth mappings and timing tracks.

Version 0.86.3.4 - Windows, macOS and Linux

Actual editor screenshots. No practice files required.

How to use this guide

Work through the first character setup in order, then choose the portrait and mouth method that fits your artwork. You do not need to configure every method for every character.

Chapters	What you will do
1-4	Find the right records, assign artwork, and place a character.
5-11	Build animation states, movement, player/NPC behavior, and interactions.
12-16	Create portraits and connect the speaker, text, voice, and dialogue GUI.
17-20	Choose a timing mode and map a model, sprite, or portrait mouth.
21-22	Generate, correct, save, and organize recording-specific timing.
23-25	Use actions/scripts, test the result, and find settings quickly.

The pictures were captured from the running editor and game preview. Close-ups are cropped so you can read the controls. The examples use bundled models, sprite artwork, and one Clockmaker voice recording; the portrait stills were captured from the bundled model in the engine.

Use your own project and assets. Example IDs and numbers show how a setup fits together; they are not universal presets. Screenshots use Compatibility rendering, so materials and lighting can look different on another renderer.

Apply the section, save the project, then test. State and portrait editors have their own apply buttons. A recording also has a separate Save Track command. No companion project or practice files are included.

Contents

Click a chapter or page number to jump to it. The PDF bookmarks also include section headings.

1. What you are setting up . . . 4
2. Find your way around Character Designer . . . 5
3. Assign an actual 3D model . . . 7
4. Place the character and fit it to the scene . . . 8
5. Map idle, walk, and the other animation states . . . 12
6. Use sprites and image animation instead of a 3D model . . . 14
7. Set identity, movement, collision, and footsteps . . . 16
8. Give an NPC an activity or a companion to follow . . . 19
9. Choose the player and set personal interfaces . . . 20
10. Give the character something to say and do . . . 23
11. Finish the character's extra settings . . . 25
12. Add a still portrait . . . 28
13. Animated, video, and Live2D portraits . . . 30
14. Make a 3D model portrait . . . 33
15. Connect a line, its speaker, and its voice . . . 37
16. Put portraits in a dialogue GUI . . . 39
17. Choose how speech will drive the mouth . . . 40
18. Map a 3D face with blend shapes . . . 43
19. Map a drawn portrait or sprite face . . . 47
20. Jaw bones, mouth textures, and Live2D . . . 52
21. Generate, preview, and correct a timing track . . . 55
22. Manage many recordings and translated speech . . . 60
23. Use actions and scripts during play . . . 62
24. Test the complete character . . . 65
25. Character Designer reference . . . 68

1. What you are setting up

A character is more than the model standing in a room. It has a reusable definition, animation states, movement rules, interaction responses, and optional portraits and speech settings. Those pieces work together, but you do not have to set them all up at once.

This guide takes you through the Character Designer in PCA 3D Adventure Studio 0.86.3.4. It uses the engine's bundled character models and a recorded Clockmaker line for the pictures. Follow the same steps with your own artwork and recordings. No practice project or separate asset pack is required.

The instructions apply to Windows, macOS, and Linux. Your file picker or theme may look different. Read the selected character, section title, and field labels rather than trying to match every window border.

The order that keeps things manageable

1. Import the artwork, create the character, and assign its source.
2. Map idle and walk, place it in a usable room, and test movement.
3. Add a portrait and check its framing.
4. Connect a spoken line to its speaker and voice asset.
5. Prepare that recording's mouth timing and the character's face mapping.
6. Test the scene character and portrait together.

A still portrait does not acquire a moving mouth simply because a voice plays. The voice needs a timing source, and the visible face needs a usable mouth mapping. We will connect those two pieces in the second half of the guide.

Three records that are easy to confuse

Record	What belongs there
Asset	The imported model, picture, sprite sheet, animation, or sound file.
Reusable character	Identity, source artwork, states, movement, portraits, and lip-sync defaults.
Placed scene character	That character's position in a room and any scene-specific overrides.

Edit the reusable character when the change should follow it throughout the game. Edit the placed instance when the change belongs only to that room. A portrait is another view of the character; it is not another physical actor placed beside them.

Keep a saved working version. Apply the page, save the project, and test before adding the next feature.

2. Find your way around Character Designer

Open **Character Designer**, then choose a character in its record selector. The example uses **Ivo (player)** as the player and **Elias (elias)** as the workshop keeper. The text in parentheses is the ID used by references and scripts.

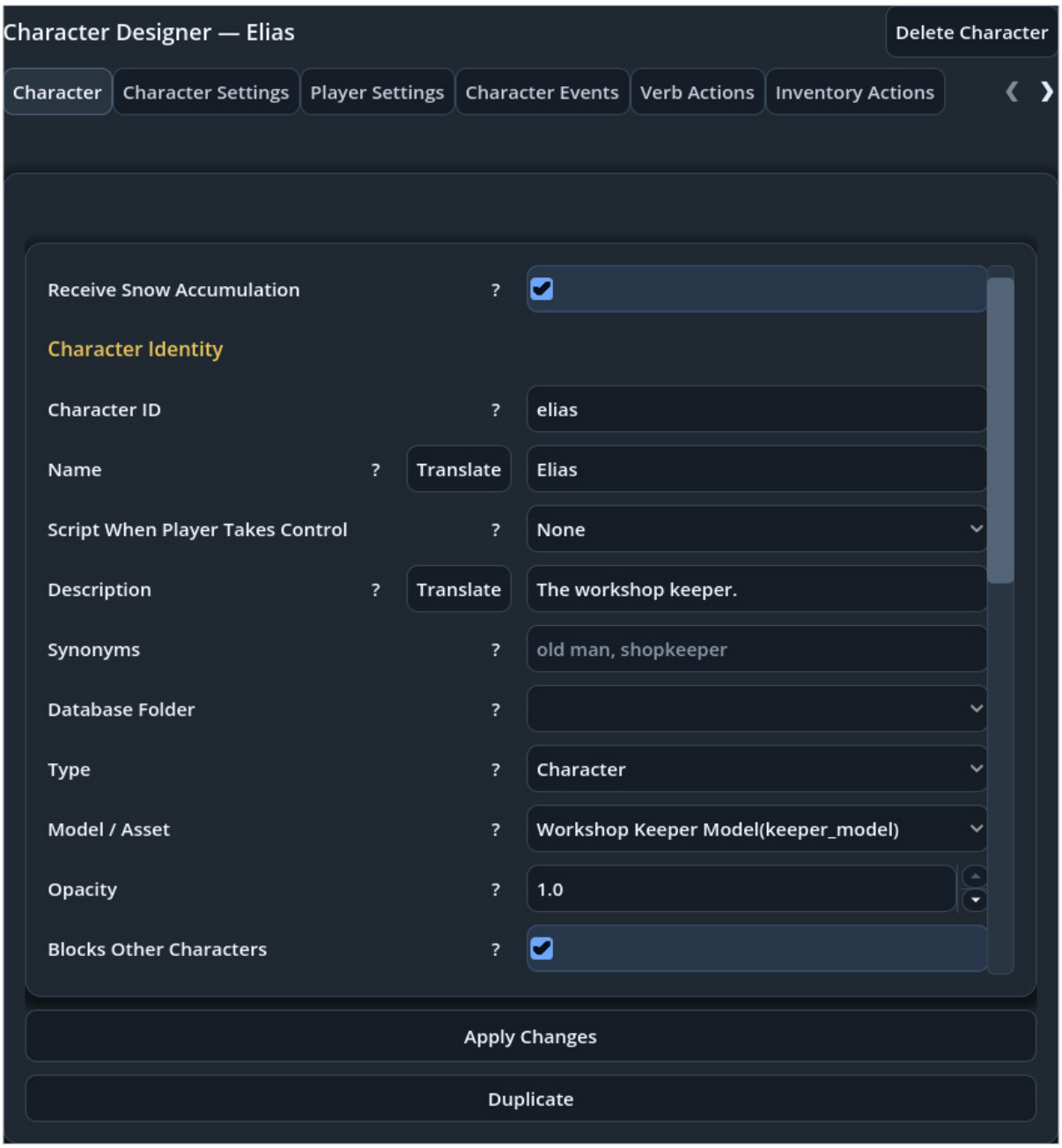


Figure 1. The Character Designer has its own section row. Choose the reusable character before editing its settings.

Section	What you use it for
Character	Identity, source model/asset, blocking, movement sound, event volume, and approach behavior.
Character Settings	Display name, role, speech color, movement, NPC activity, collision, and following.
Player Settings	Playable status, starting player, switching, personal GUIs, and starting inventory.
Character Events	Collision/entry, exit, and pointer event actions.
Verb Actions	Responses to Look, Talk, Use, and the other authored verbs.
Inventory Actions	Responses when an inventory item is used on the character.
2D / 2.5D	Sprite/billboard presentation, image fitting, and attachment data.
Custom Properties	Your own values stored on this character.
Raw JSON (Advanced)	The selected character's nested data.
Animation States	Which animation plays for each named state.
Portraits	The pictures, videos, or model views used in dialogue.
Lip Sync	Timing mode, targets, and mouth mappings.

Use the section row's arrows if a title is offscreen. The first section is called **Character** in this designer; an ordinary Inspector may call its equivalent **General**.

The reusable character deliberately omits scene-only appearance pages such as **Visual Adjustments** and **Pointer Feedback**. Select the placed character in Scene Designer for those pages. Their absence here is not evidence that the character cannot be moved or tinted.

Add, clone, or delete

Use **Add New** to create a character. Give it a readable name and a stable ID before other records start referring to it. **Clone Selected** is useful for a genuinely different character based on an existing setup. Check the clone's model, states, portraits, player flags, and actions afterward.

Delete Character removes the reusable definition. Check scene placements, dialogue speakers, actions, and project starting-player settings before deleting a character used by the game. Run Validate afterward. Do not delete the definition merely because you want to remove one scene placement.

Keep IDs boring and dependable: `elias`, `apprentice`, `guard_west`. Use names for readable labels. Renaming a person in the story does not always require changing their ID.

3. Assign an actual 3D model

Import the model through **Assets** first. A GLB with its materials and textures included is a convenient starting point. Check the asset preview before assigning it to a character. If the preview cannot load the mesh, Character Designer will not repair the file by selecting it again.

1. Add or select the reusable character.
2. Open **Character**.
3. Enter its name, ID, description, and optional synonyms.
4. In **Model / Asset**, select the imported **Model** asset.
5. Leave opacity at 1 for a normal visible character.
6. Apply the character page and save the project.

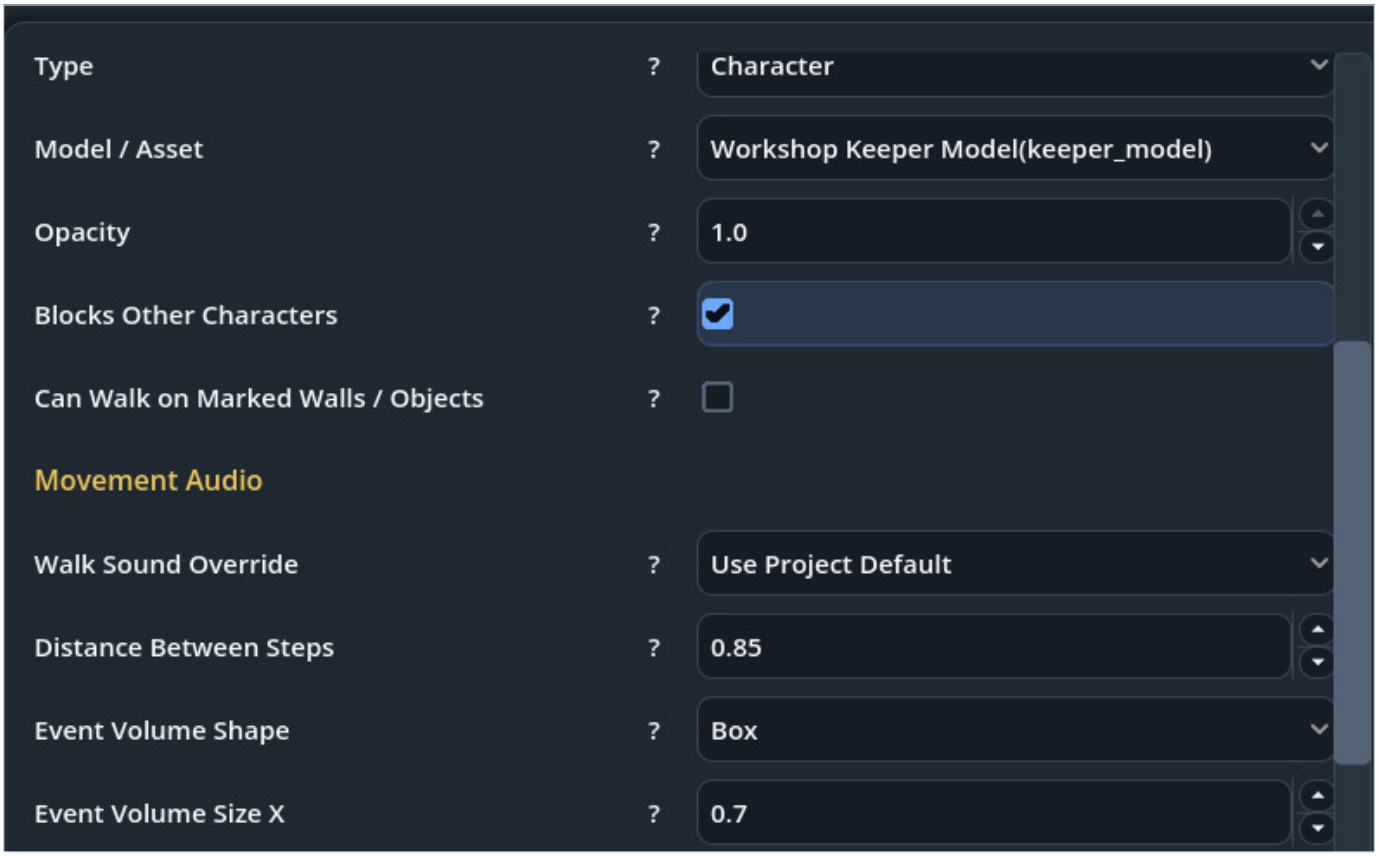


Figure 2. Model / Asset belongs to the reusable character definition. It points to an imported asset record.

The model and its animation states are separate assignments. Choosing a walk animation alone does not necessarily assign the model that should be visible. Start with the correct model asset, then map its states in Chapter 5.

The character's **Name** identifies the editor record. **Display Name** in Character Settings is the name used for character presentation. A portrait also has its own display name and ID. Give these sensible names without assuming they are interchangeable references.

If all you see is a simple polygon

Check these in order:

1. Does the imported model appear correctly in Assets?
2. Is it assigned to the reusable character's **Model / Asset** field?
3. Is the scene item actually a **Character** linked to that definition?
4. Did you apply and save the change?
5. Is the model visible at a sensible height, scale, and position?
6. Does Test Scene show the same result, or are you looking at an editing helper?

A primitive placeholder, an empty asset, and a character's event-volume helper can all look much simpler than the finished model. Confirm the selected record before changing animation settings to solve a geometry problem.

Missing materials or animations

A model can load while its external textures are missing. Keep the imported asset's related files together and fix the asset import first. Likewise, a static mesh can be a perfectly valid model without containing a walking skeleton or any animation clips.

For clips imported from a different file, the skeleton and animation must be compatible with the character. A dropdown containing a clip is not proof that it was made for the selected skeleton. Test the pose and movement before building dialogue around it.

4. Place the character and fit it to the scene

The definition gives you a reusable character. Scene Designer gives it somewhere to stand.

1. Open a usable room with navigation and a default camera.
2. Choose **Add to Scene > Character** and select the reusable character, or use the character-placement workflow offered by the editor.
3. Place it on clear walkable ground.
4. Select the placed character in Scene Objects.
5. Open **Visual Adjustments** and check its XYZ position, rotation, and scale.
6. Use Test Scene to check its visible feet, height, facing, and movement.



Figure 3. Select the placed character for Visual Adjustments. Scene alignment and the reusable character definition serve different jobs.

Use a room with a floor around Y = 0 for the first test. Avoid placing the character inside a table, wall, or doorway collider. You can return to the more difficult placement once the basic model works.

Model Height is a useful character-scale reference; 1.8 world units is a reasonable starting point for the pictured adult models. **Model Facing Offset** corrects the artwork's forward direction. **Lock Root Motion** helps keep an imported movement clip from dragging the visible mesh away from the engine-controlled actor.

The placed instance exposes model alignment and appearance controls that do not belong to the reusable definition's main page. Ordinary transform rotations use radians. Fields explicitly labeled degrees, including portrait rotation, use degrees.

Reuse the existing character when changing rooms

Keep one placement for the same reusable character. Room-changing and character-moving actions transfer that character to the destination. Do not place the same definition in several rooms to imitate the journey. Validation can reject duplicate placements.

For two genuinely different guards with the same artwork, clone the definition and give each guard its own ID. They can share the asset while keeping separate character identities and behavior.

Physical body, event volume, and approach

These are separate controls:

Control	What it affects
Collision Radius / blocking	The space needed for physical movement and clearance.
Event volume	The region that detects entry, exit, or pointer-related interaction.
Stop Distance	How close the player tries to get before interacting.
Visible model size	The appearance of the artwork.

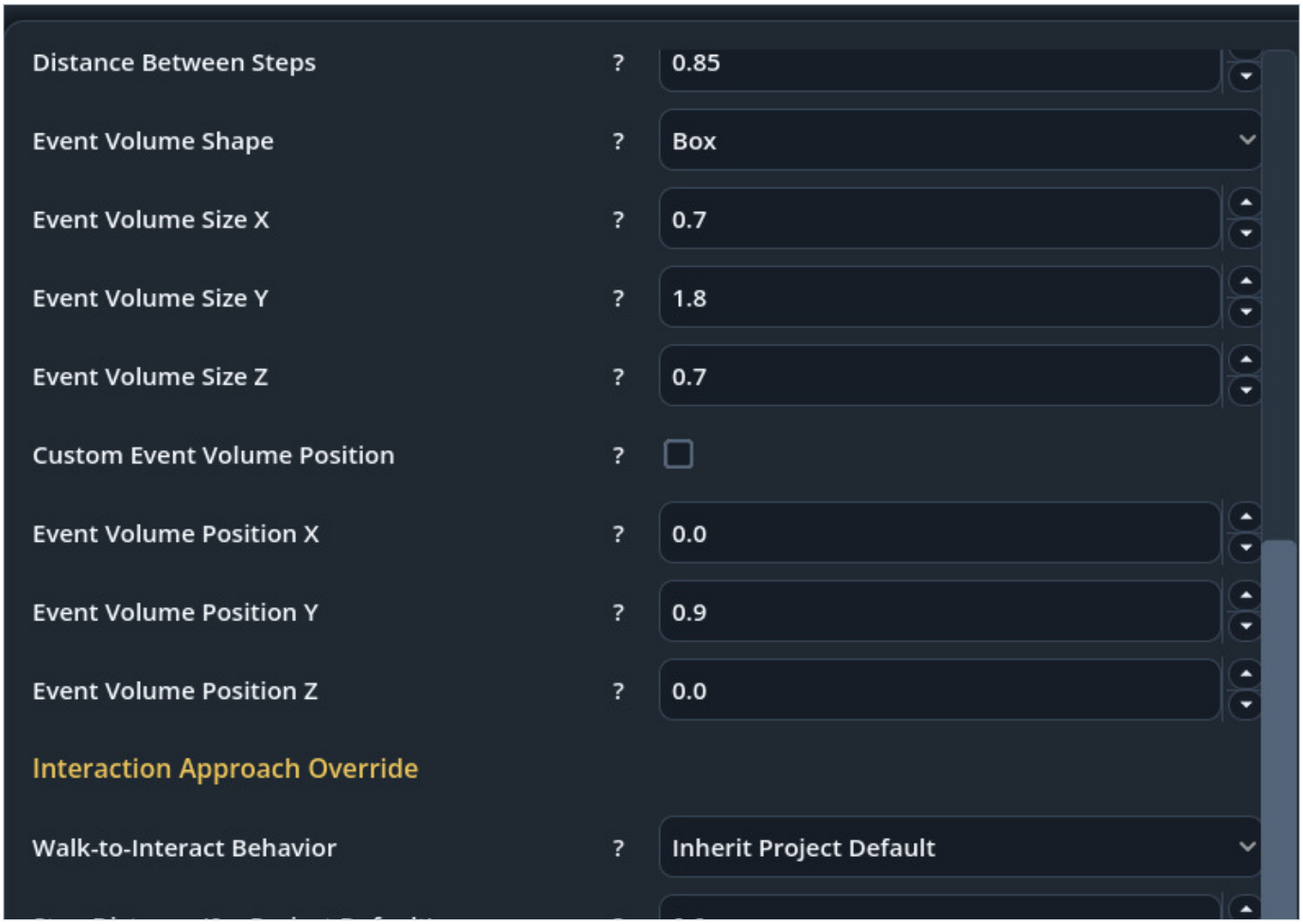


Figure 4. The character definition can supply event-volume defaults that a scene placement may override.



Figure 5. Approach behavior can inherit the project choice, always approach this character, or disable automatic approach.

A small event volume does not guarantee that two physical characters can stand shoulder to shoulder. Check their collision radii and the route's clearance as well. Conversely, a tall event volume does not mean the visible model should be stretched to match it.

5. Map idle, walk, and the other animation states

A **state** is the gameplay name, such as `walk`. A **clip** is the animation inside the imported file, which might be named `Walking`, `Armature|walk`, or something much less friendly. You connect the two in **Animation States**.

1. Select the reusable character and open **Animation States**.
2. Click **Add State** when you need a new state, or select an existing row.
3. Set **State ID** to `idle` and give it a readable display name.
4. Set **Animation Source** to **Model Clip**.
5. Choose the appropriate **Animation / Sprite Asset**, or use the current character model where that source permits it.
6. Choose the actual clip from **Known Clip**. Check **Clip Name** rather than guessing it from the English word Idle.
7. Enable **Loop** for a repeating idle.
8. Click **Validate and Apply State**.
9. Set the default state to the intended idle, then save the project.

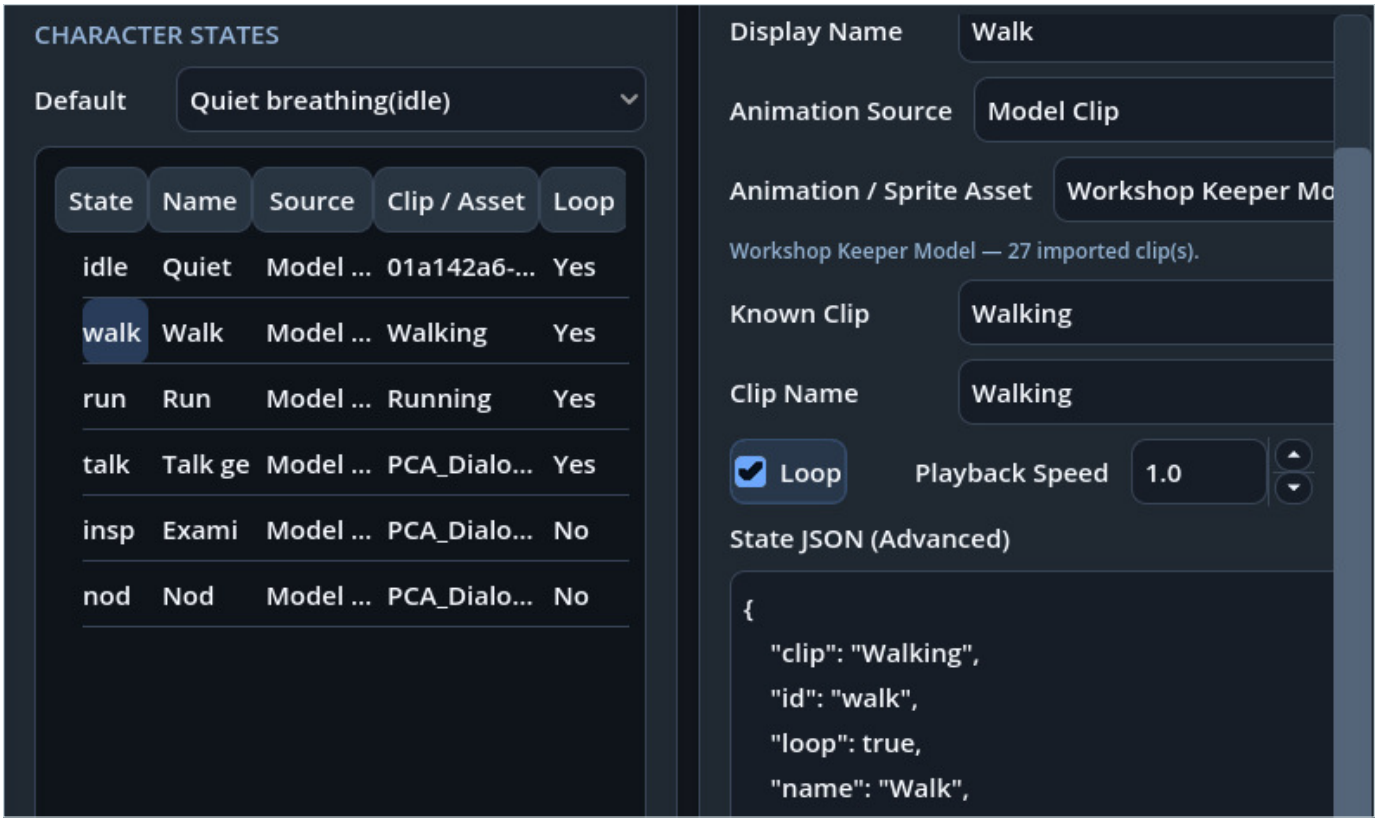


Figure 6. The state name and imported clip are separate. The pictured walk state points to the model's Walking clip.

Create `walk` the same way, using the walking clip and looping it. Add `run` only when the character has a useful running clip and the game can actually request running. Add gestures such as `inspect` or `nod` for actions that explicitly call them.

State	Typical use	Usually loops?
idle	Standing without moving	Yes
walk	Normal movement	Yes
run	Running movement	Yes
talk	Body gesture or talking pose	Often
inspect	A deliberate examination gesture	Usually no
nod	A brief response	Usually no

The pictured models happen to have clips named Walking, Running, and several PCA dialogue gestures. Your model may have different names or fewer clips. Map what actually exists.

Playback speed and frame rate

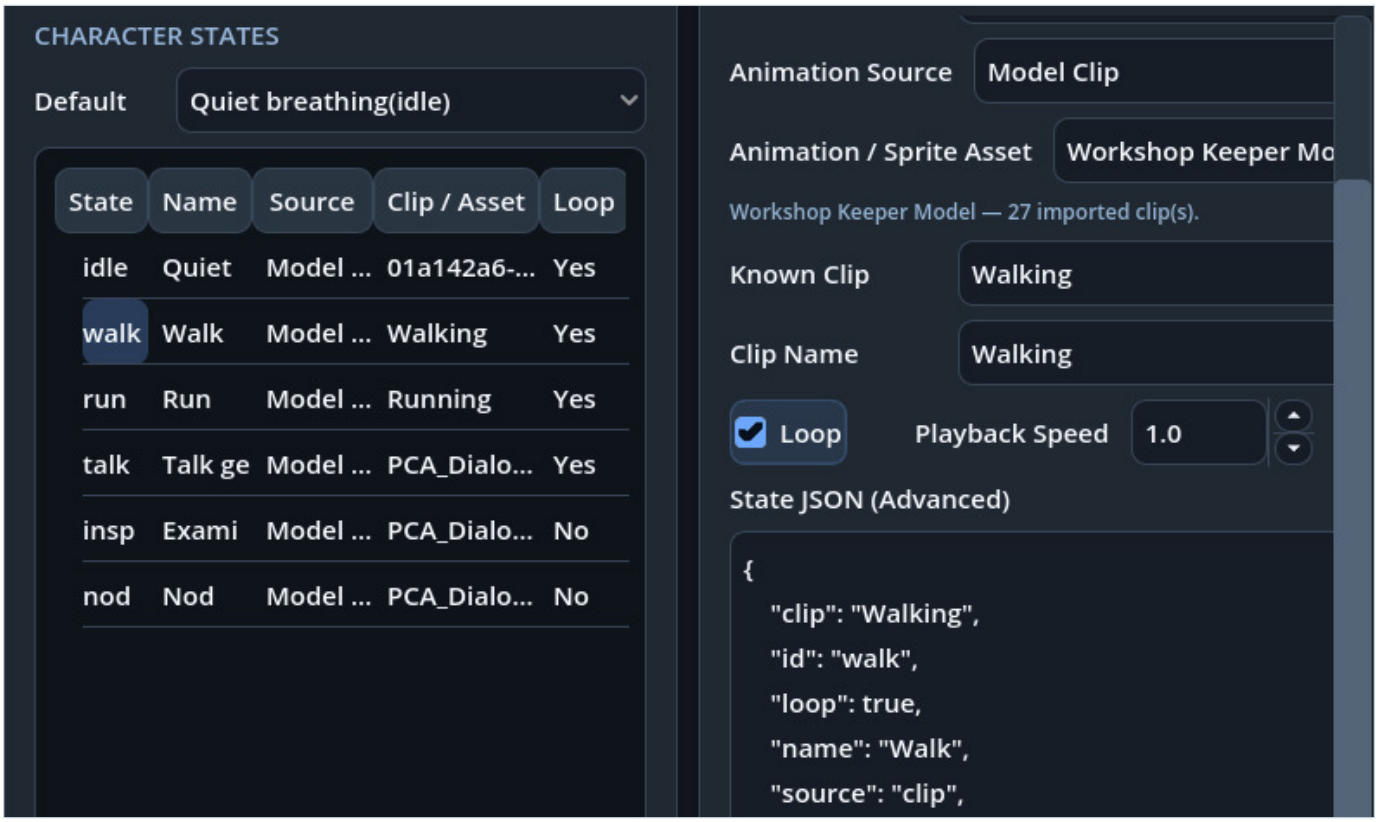


Figure 7. Loop, Playback Speed, and FPS have different jobs.

Playback Speed multiplies the clip speed. One is the imported speed; zero holds the first frame. It does not change the character's distance traveled per second. **FPS** is used by frame-based and generated sources, rather than being a universal fix for skeletal animation.

If the feet slide, compare the motion in the clip with **Walk Speed** in Character Settings. Start with a modest walk speed and normal playback, then adjust one at a time. If the model turns sideways while walking, correct its facing alignment rather than increasing the animation speed.

Preview State restarts the relevant runtime scene and plays the selected state. Save your setup and use a suitable scene placement. A character with no useful placement may not produce the preview you expect.

Duplicate, **Move Up**, **Move Down**, and **Delete** organize states. Deleting or renaming a state can break actions that call its ID. State order does not make every state play in sequence.

6. Use sprites and image animation instead of a 3D model

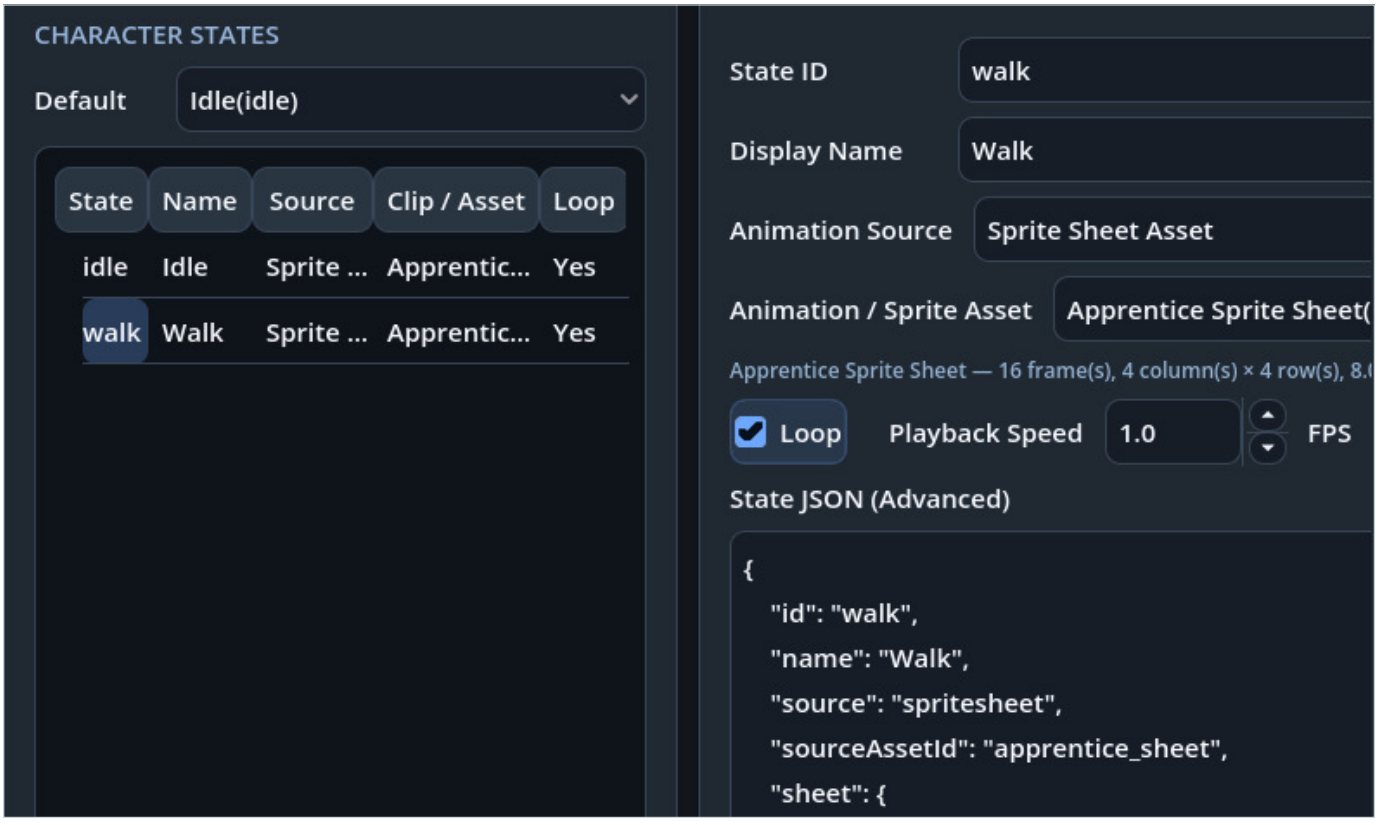


Figure 8. An image-based character uses the same state list. This walk state uses an authored frame range from the bundled Apprentice sheet.

Animation States can use more than imported skeletal clips. Choose the source that matches the actual asset.

Source	What you supply
Model Clip	A clip from a compatible model asset.
Sprite Sheet Asset	An imported sheet with its frame grid defined in Assets.
Image Animation Asset	A prepared sequence of images.
Manual Frame List	One image asset ID or supported path per line.
Generated Animation	The generated setup and state data supported by that source.

For a sprite sheet, import it as **Sprite Sheet**, define the grid and FPS in Assets, and confirm that its preview shows individual frames. Then select **Sprite Sheet Asset** in the state and choose that asset. A whole sheet displayed as a single picture is an asset/setup problem, not a character who needs to become sixteen times wider.

Use an idle asset for `idle` and a walking asset for `walk`. If one large sheet contains several actions or directions, prepare suitable animation assets or intentionally configure the advanced frame ranges. Do not assume the editor will identify the correct row from the state name alone.

For a Manual Frame List, put the frames in their intended order and use the state's FPS and Loop settings. For an Image Animation Asset, check the imported sequence first, then assign it to the state.

Fitting a sprite's visible body

Open **2D / 2.5D** for its presentation and image-fitting controls. **Fit Character to Visible Pixels** ignores transparent padding around frames. **Visible Character Height** and **Foot Anchor Offset** provide deliberate corrections when automatic fitting needs help.

Keep the visible feet on the scene position. A large transparent margin can otherwise make the character appear small or float above the floor. Use matching frame dimensions and consistent alignment across the sequence.

Flip With Direction is useful for artwork that should mirror horizontally. Check the result with asymmetrical clothing, equipment, and gestures. Flipping a picture also flips those details; it does not create a newly drawn rear view.

When image animation is assigned, Character Settings can expose **Animated Paper / Cardboard Cutout**. This keeps the image animation active while giving the flat character the supported room-lighting treatment. Tune its darkness and tint in the actual room lighting.

A single character can have states using different source types. That flexibility is useful, but prove the basic idle/walk pair first. Switching to an image state changes the visible renderer; it is not a promise to combine a full sprite over an already visible model.

7. Set identity, movement, collision, and footsteps

Open **Character Settings** for the character's shared behavior.

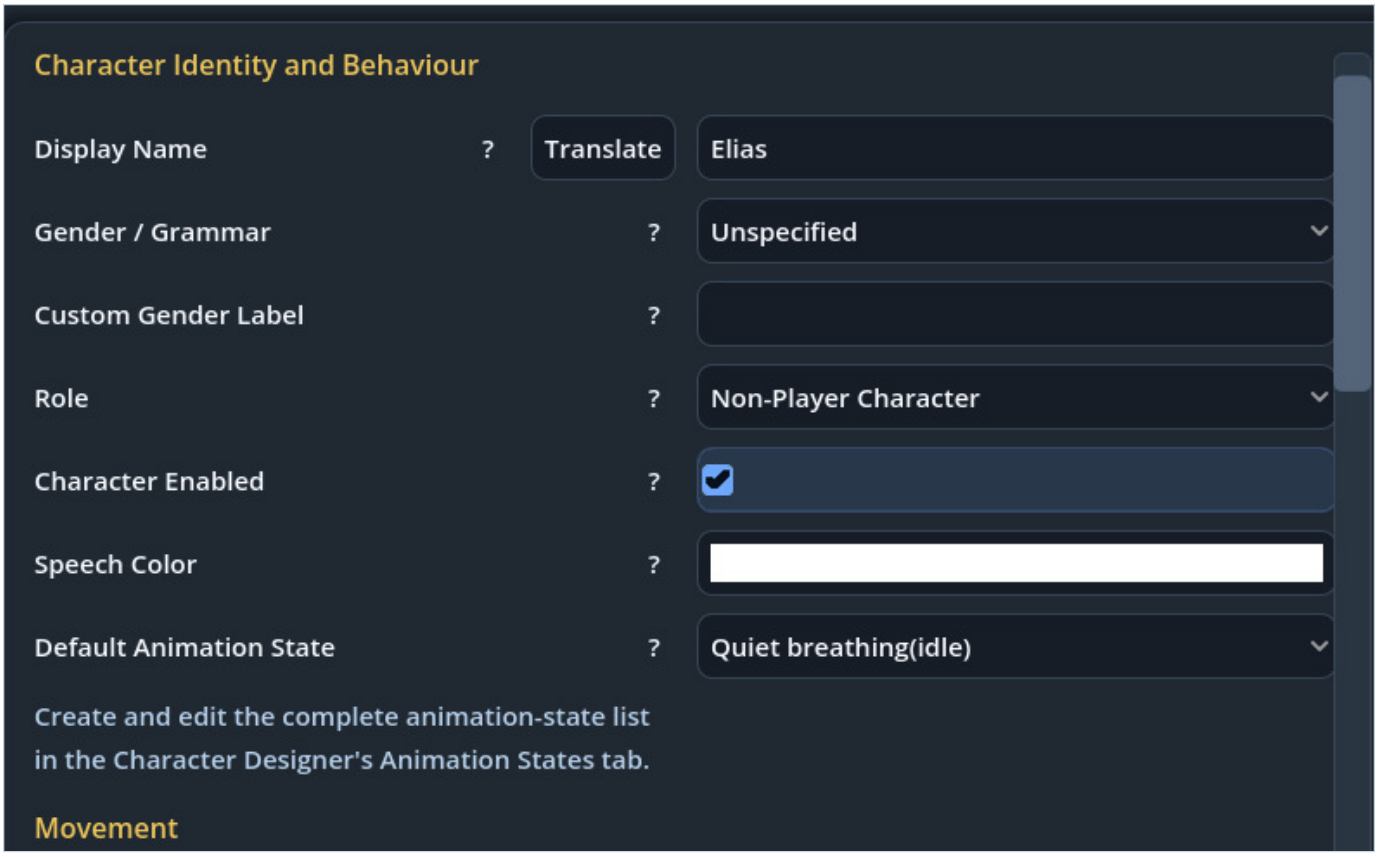


Figure 9. Display name, grammar metadata, role, speech color, and default animation are character-level choices.

Gender / Grammar and its custom label supply authored character information. **Role** distinguishes an NPC from a player-type character. **Speech Color** is the character's text color where the speech interface uses it. **Default Animation State** chooses the normal starting state.

Set a sensible movement speed

Enable **Use Character Walk Speed** to use this character's own value. Disable it to use the project's Default Walk Speed, subject to scene overrides. Set Walk Speed, Run Speed, and Turn Speed, then test the character on a straight route and around a corner.

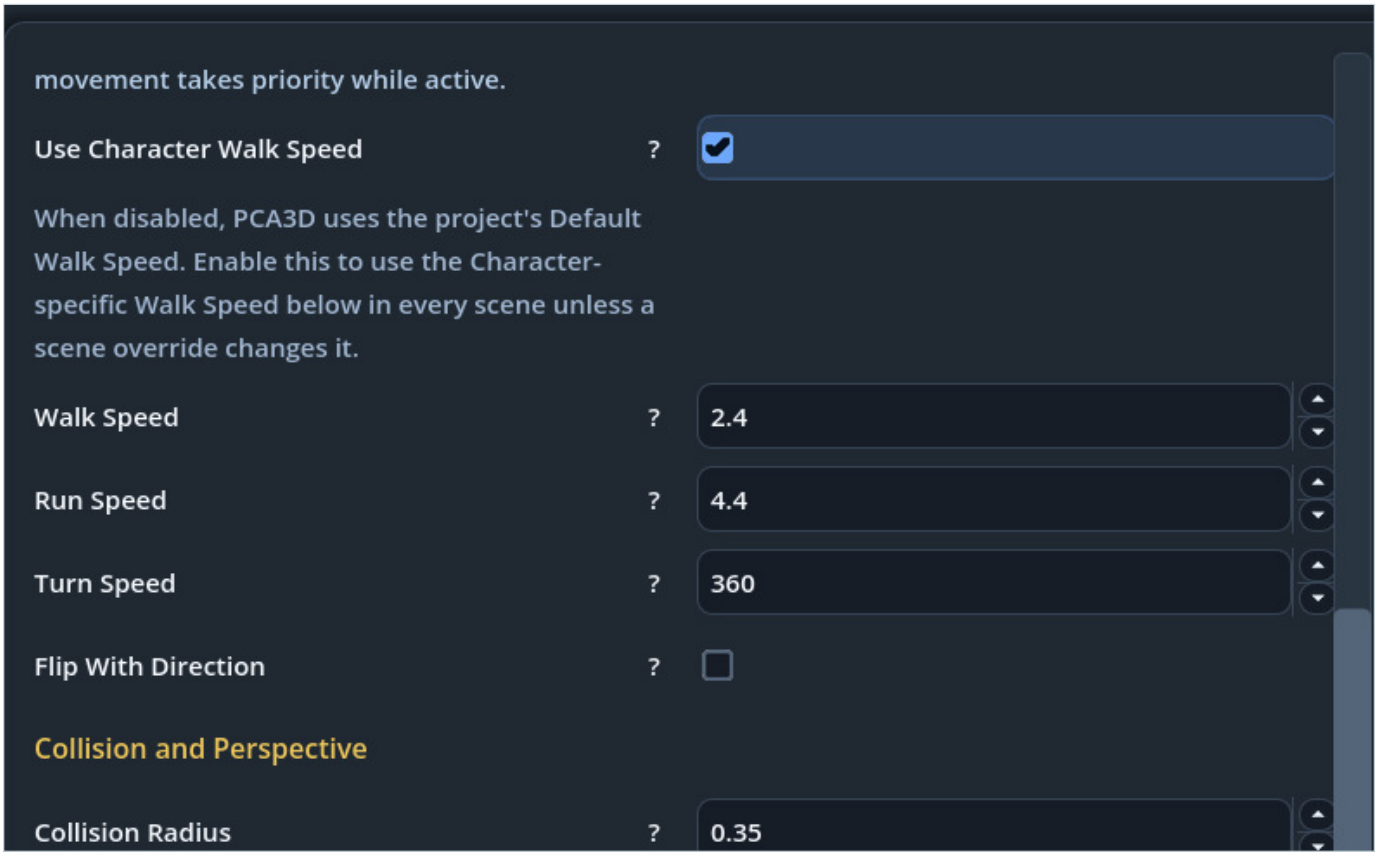


Figure 10. Walking, running, and turning are independent settings. The custom-walk-speed checkbox decides which walk value is used.

The example starts around walk speed 2.4 and run speed 4.4. These numbers make sense only relative to the room's world scale. A tiny character on an enormous floor may need a different setup from a normal-sized person in a small workshop.

Collision Radius affects clearance around the moving character. **Model Height** affects its authored size reference. **Perspective Scaling**, Near Scale, and Far Scale are for a deliberate depth-scaling treatment where the presentation uses it. Start with ordinary model scale before adding another scaling system.

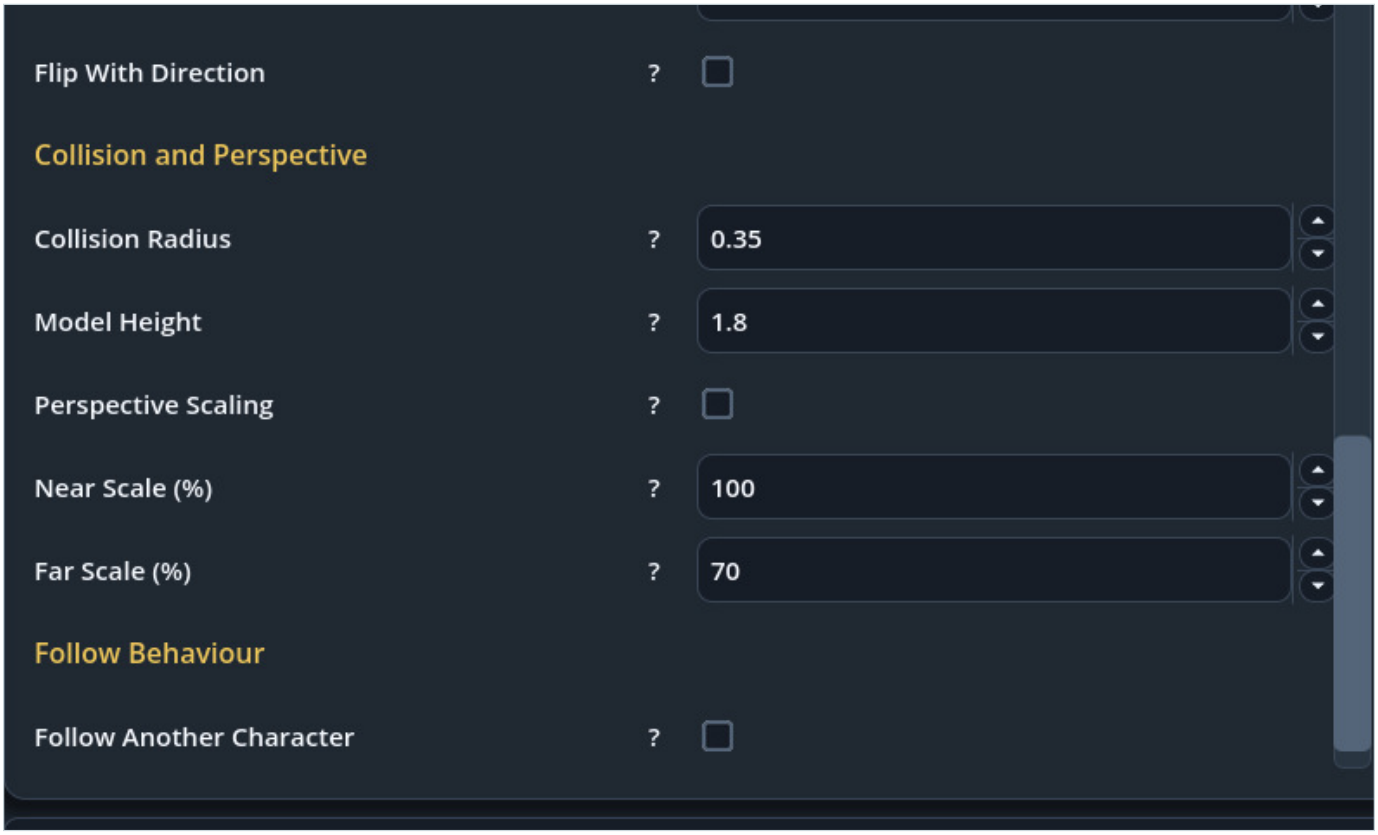


Figure 11. Collision, perspective settings, and following behavior are grouped farther down Character Settings.

Footsteps

Return to **Character** and find **Movement Audio**. **Walk Sound Override** can use the project default, an assigned audio asset, or No Sound. **Distance Between Steps** controls the stride interval.

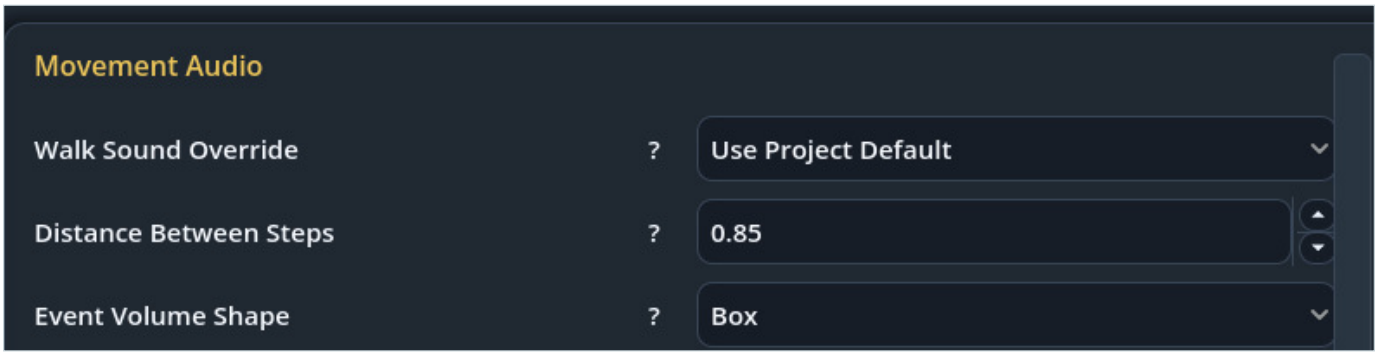


Figure 12. Walk Sound Override and Distance Between Steps control this character's footstep source and stride interval.

Choose a short footstep sample with sensible trimming. Test walking, stopping, and turning. A long recording with several footsteps already inside it may sound wrong when the engine triggers it once per stride.

Can Walk on Marked Walls / Objects belongs to the surface-walking setup. The scene needs suitable marked physical surfaces and compatible movement. Enable it for a deliberate surface-walking game, not as a general fix for an ordinary blocked floor.

8. Give an NPC an activity or a companion to follow

Start with **Stand / Scripted** when you want full control through actions. Once ordinary navigation works, choose another **NPC Behavior** in Character Settings.

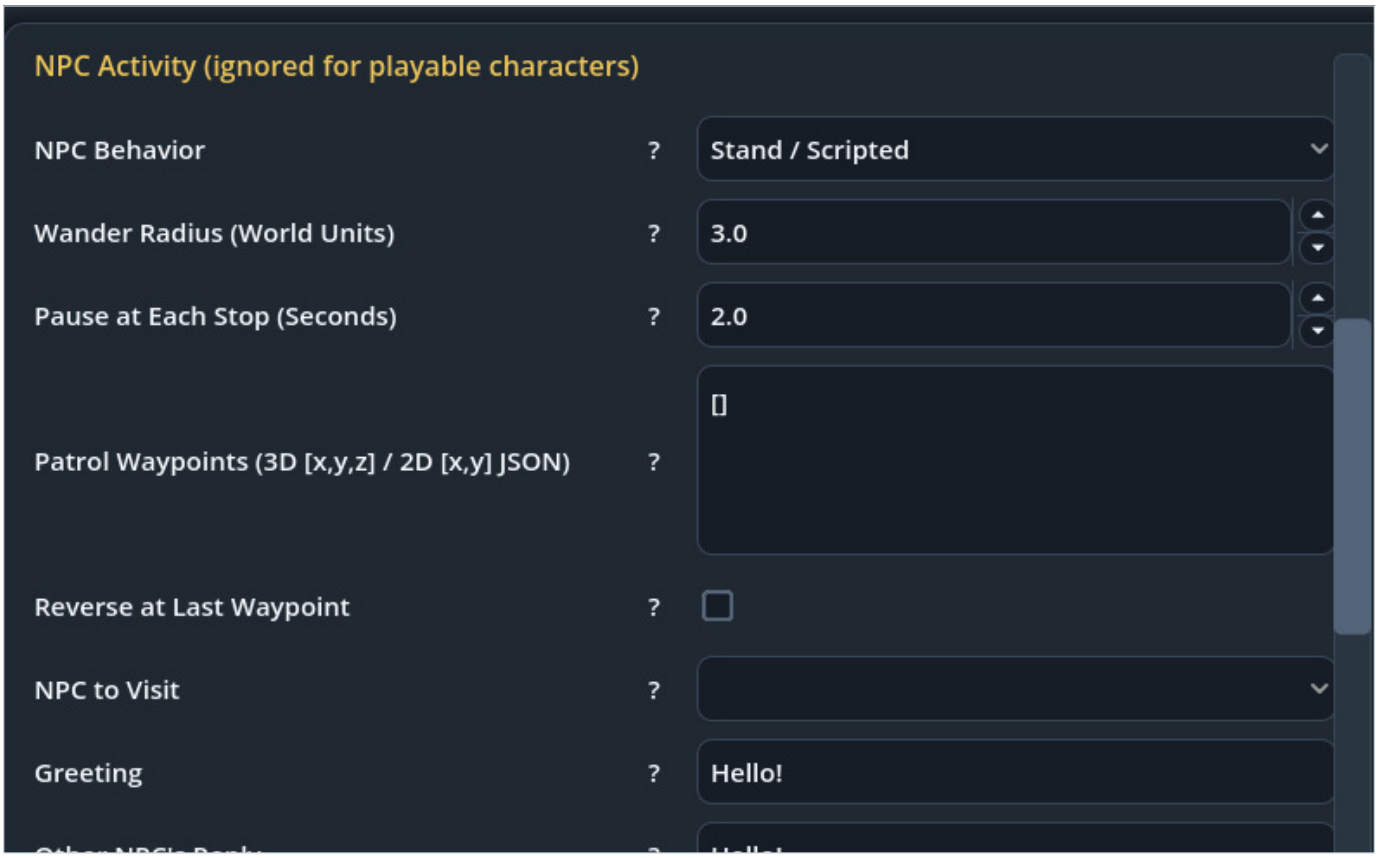


Figure 13. NPC activities are ignored for playable characters. Scripted movement takes priority while it is active.

Behavior	What to configure
Stand / Scripted	Keep it still until an action moves it.
Wander Near Start	A wander radius and pause. It stays near the scene placement.
Follow Waypoints	An authored point list, pause, and optional reversal at the end.
Visit and Talk to NPC	Another NPC to visit, greeting, reply, and pause.

For a short workshop patrol, use points such as `[[2,0,-1],[3,0,1],[1,0,1]]` in a 3D room. They are room-world positions, not offsets from the NPC. Put every point on reachable navigation and clear of furniture.

Reverse at Last Waypoint makes the return journey follow the route backward. Without it, check how the loop returns from the last point to the first. A route that crosses a table is still a bad route even when its JSON is valid.

For social activity, choose the actual other NPC and enter the greeting/reply. Do not target a nonexistent character and expect the system to invent someone to talk to.

Following another character

The separate **Follow Behaviour** section offers **Follow Another Character**, a target, following distance, and follow speed. A zero follow speed uses the character's own walk speed.

Begin with a comfortable gap and test a narrow passage, a corner, a stop, and a return trip. Two followers trying to occupy the same point can crowd each other. Use actions such as Follow Character and Stop Follow Character when following should begin or end during a puzzle.

The autonomous NPC behavior settings are ignored for playable characters. If an NPC seems unexpectedly eligible for player control, check Role, Playable Character, and Default Starting Character before troubleshooting its patrol.

9. Choose the player and set personal interfaces

Select Ivo and open **Player Settings**. A playable character is one the player can control. The default starting character is the one chosen to begin the game. A game can have several playable characters but only one intended default.

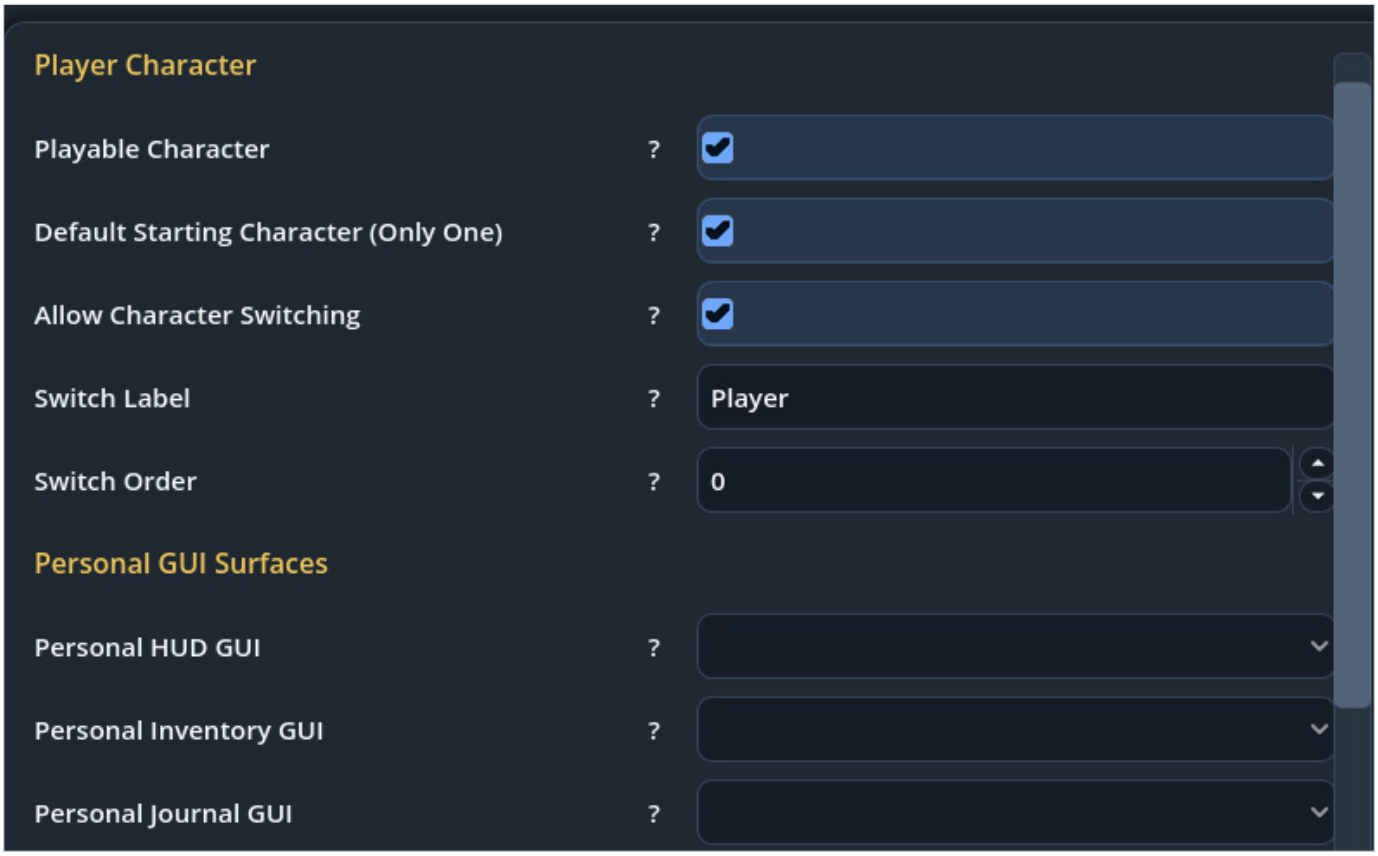


Figure 14. Playable status, the starting-player flag, and character switching are related but separate choices.

1. Enable **Playable Character** for a character the player may control.
2. Set its role consistently in Character Settings.
3. Mark only the intended **Default Starting Character**.
4. In Project Settings, confirm the selected starting player.
5. Enable **Allow Character Switching** where appropriate, and set a readable **Switch Label** and order.
6. Save, start a fresh test, and confirm which character actually receives input.

If another character does not appear in the starting-player selector, check its playable eligibility, apply the definition, then revisit the selector. For an ordinary NPC, use the NPC role and clear both playable and default-starting flags.

Personal GUI surfaces and inventory

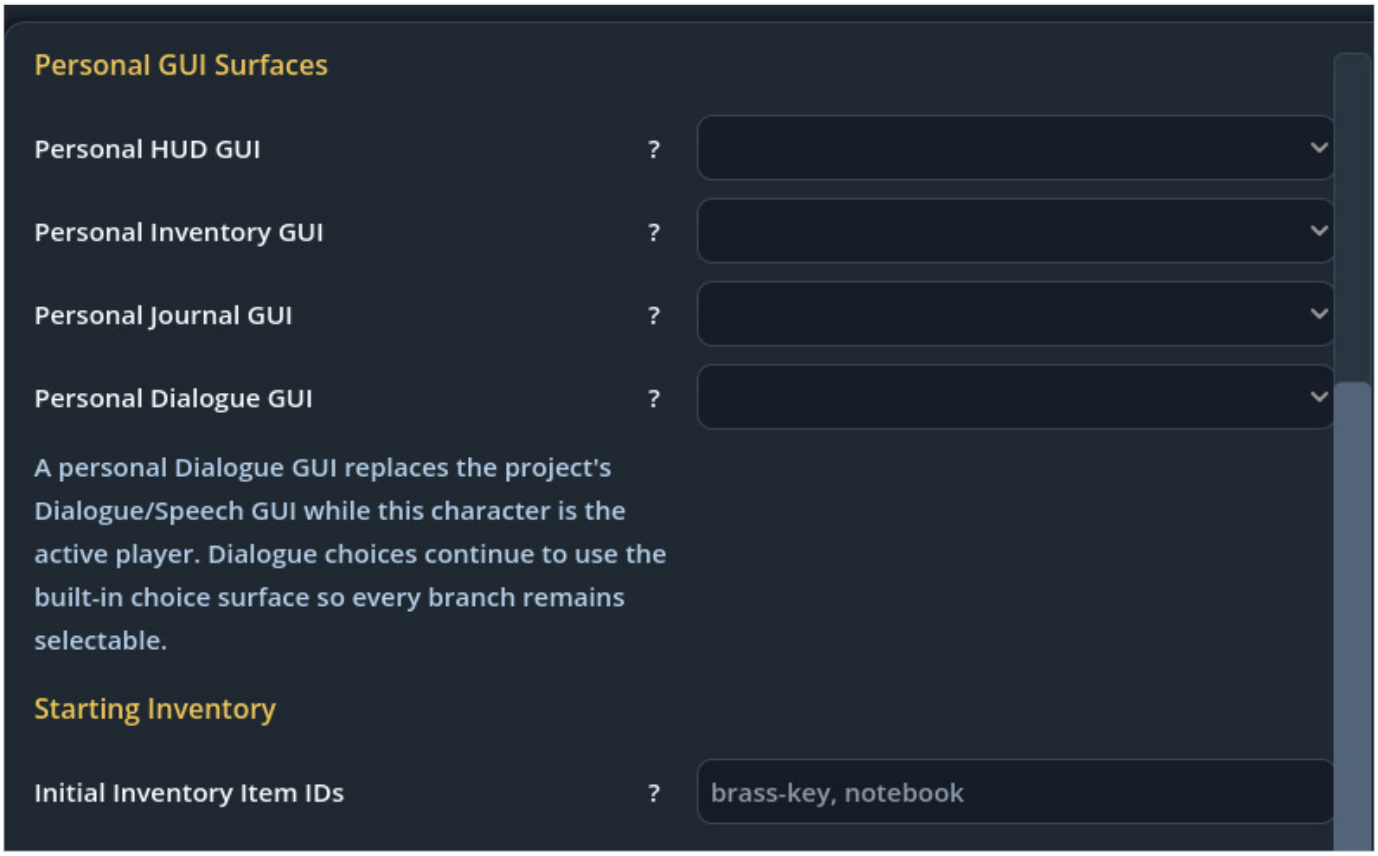


Figure 15. A character can select personal HUD, inventory, journal, and dialogue GUIs.

Choose personal GUIs when different playable characters need different interfaces. Leave the selection empty when the game's shared/default interface is appropriate. Create the GUI records before selecting them here.

A personal dialogue GUI decides which authored interface is used; it does not automatically put a portrait image into an unrelated GUI element. Chapter 16 explains the portrait bindings.

Initial Inventory Item IDs supplies that character's starting items. Create the inventory definitions first, use their actual IDs, and test the game's inventory-sharing mode. An item ID that has no definition cannot display a useful icon or response.

Script When Player Takes Control on the Character page is an optional hook for a control change. Keep it short and avoid immediately switching control back to the previous character. Test starting the game and switching later; those are distinct moments.

10. Give the character something to say and do

Characters can respond to verbs, inventory items, and events. Put the response in the page that matches its cause. A Talk response belongs in **Verb Actions**; a response to using a particular inventory item belongs in **Inventory Actions**.

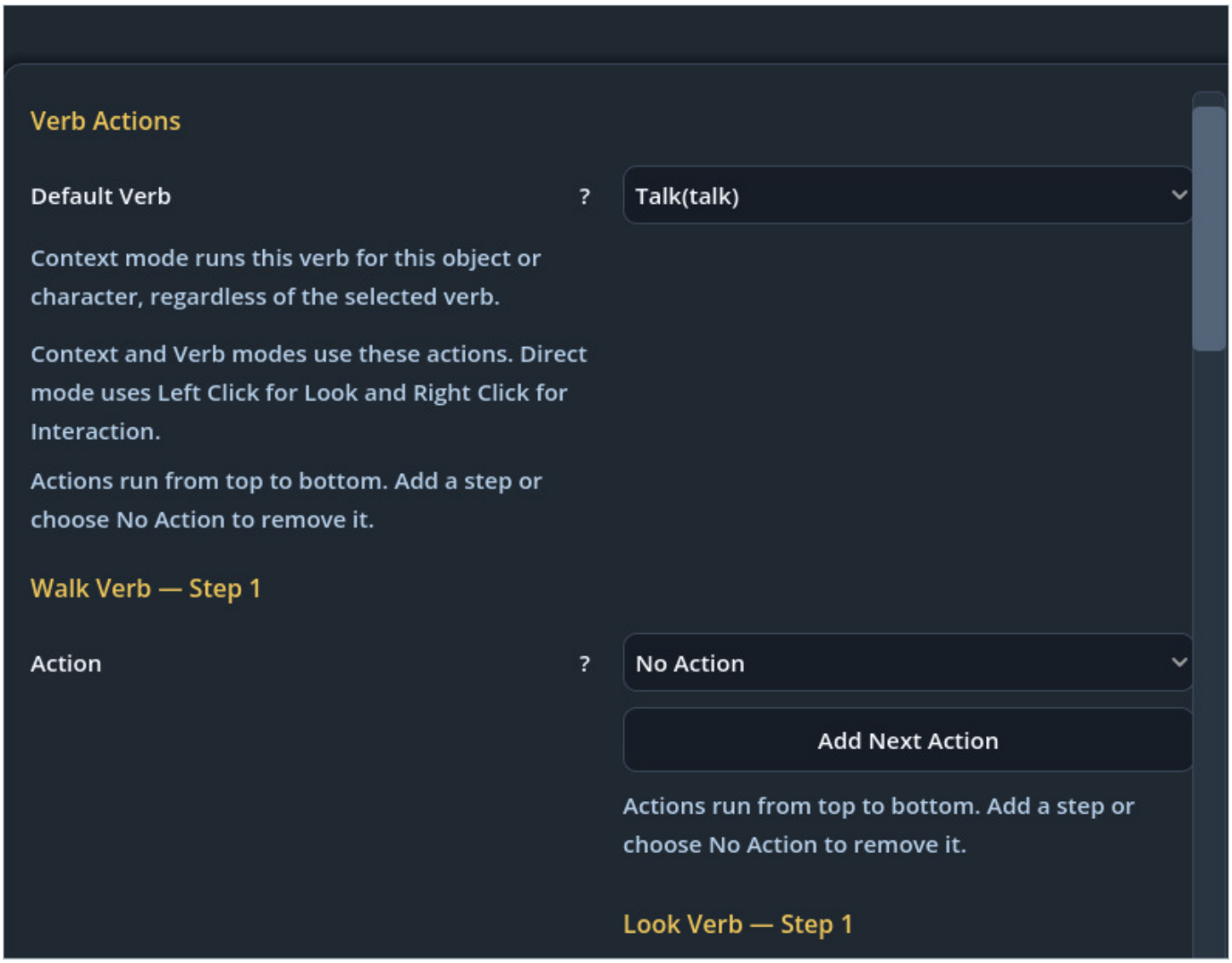


Figure 16. Verb Actions stores the responses to your game's verbs. A character's display name and its response text are separate fields.

For a first conversation, give Elias a simple Talk response before adding a full dialogue tree:

1. Select Elias in Character Designer and open **Verb Actions**.
2. Find the Talk verb used by your game.
3. Choose a speech action such as **Character Say**, or open the action-list editor for a sequence.
4. Set the speaker to Elias and enter a short line.

5. Apply the page, save, and use Talk on Elias in Test Scene.

Use an action list when one interaction needs several steps: face the player, speak, give an item, set a variable, and return control. You can use **Start Dialogue** to begin an authored conversation. Choose the actual dialogue ID after creating that dialogue.

Character Events and event volumes

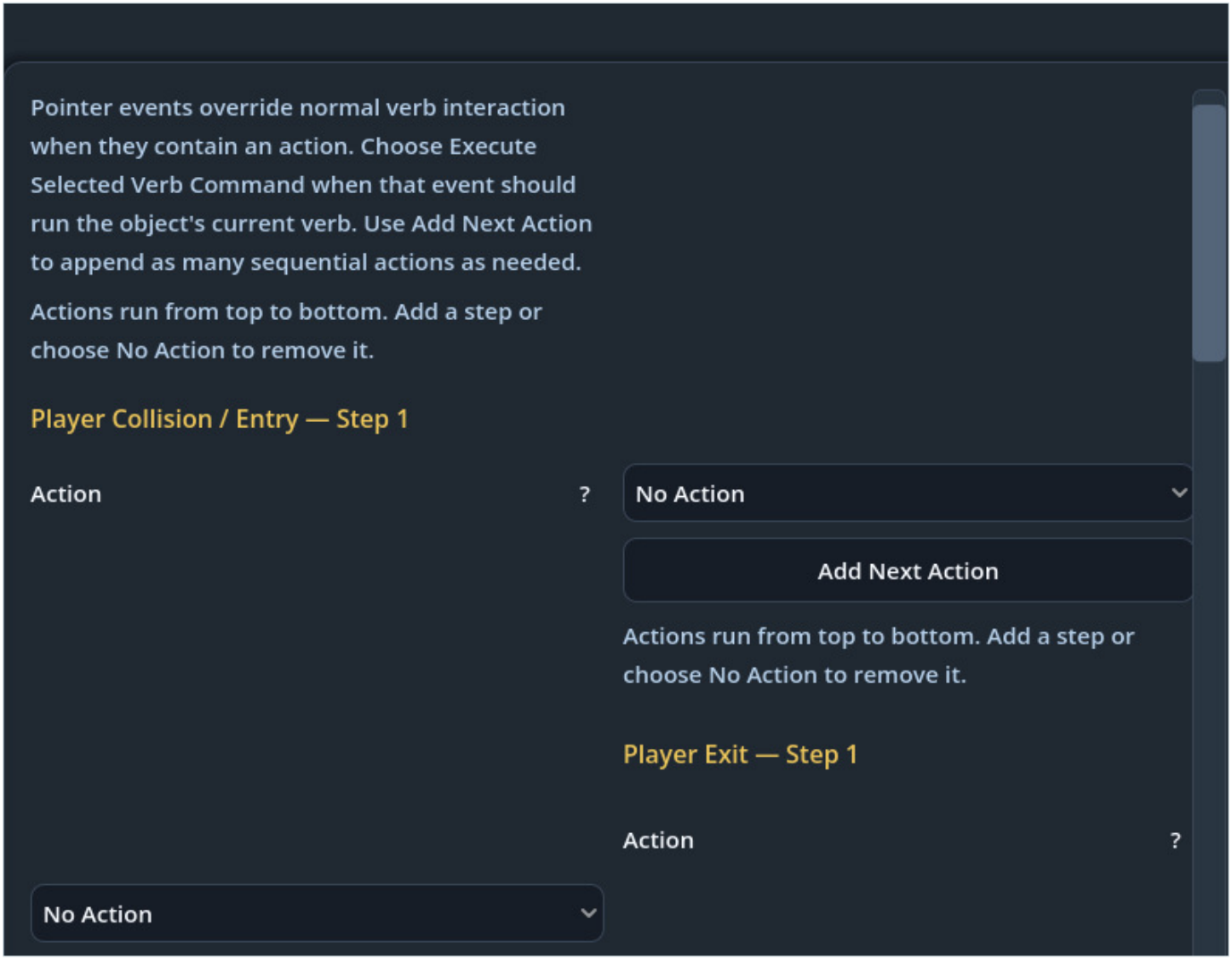


Figure 17. Character Events provides responses to contact and pointer activity. The event volume determines where the relevant detection occurs.

Open **Character Events** for the available entry/contact, exit, and pointer hooks. Configure an event volume on the Character page, then test the specific event. Pointer entry, a verb interaction, and physical contact are different triggers; do not put the same speech in all three unless you want it to fire through all three routes.

For a collision greeting, keep the response short and consider a variable so it happens once. A greeting that repeats whenever the player edges across the boundary will get old rather quickly.

Inventory Actions

Create the inventory item first. On the character's **Inventory Actions** page, choose that item and define what happens when it is used on the character. For example, showing a key to Elias could start a special conversation or set a puzzle variable.

Decide whether the item should remain in the inventory. Merely using it on someone does not mean it must be removed. Make a removal or transfer an explicit step in the action list.

> **Before you continue**
> Test the same response through the actual interaction style your game uses. A working action in isolation does not prove the selected verb, hotspot, and approach settings reach it.

11. Finish the character's extra settings

The 2D / 2.5D page

This page supplies image-based presentation settings such as sprite source, size/fitting, stage or billboard behavior, and attachment data where applicable. It does not convert a plain 3D model into hand-drawn animation.

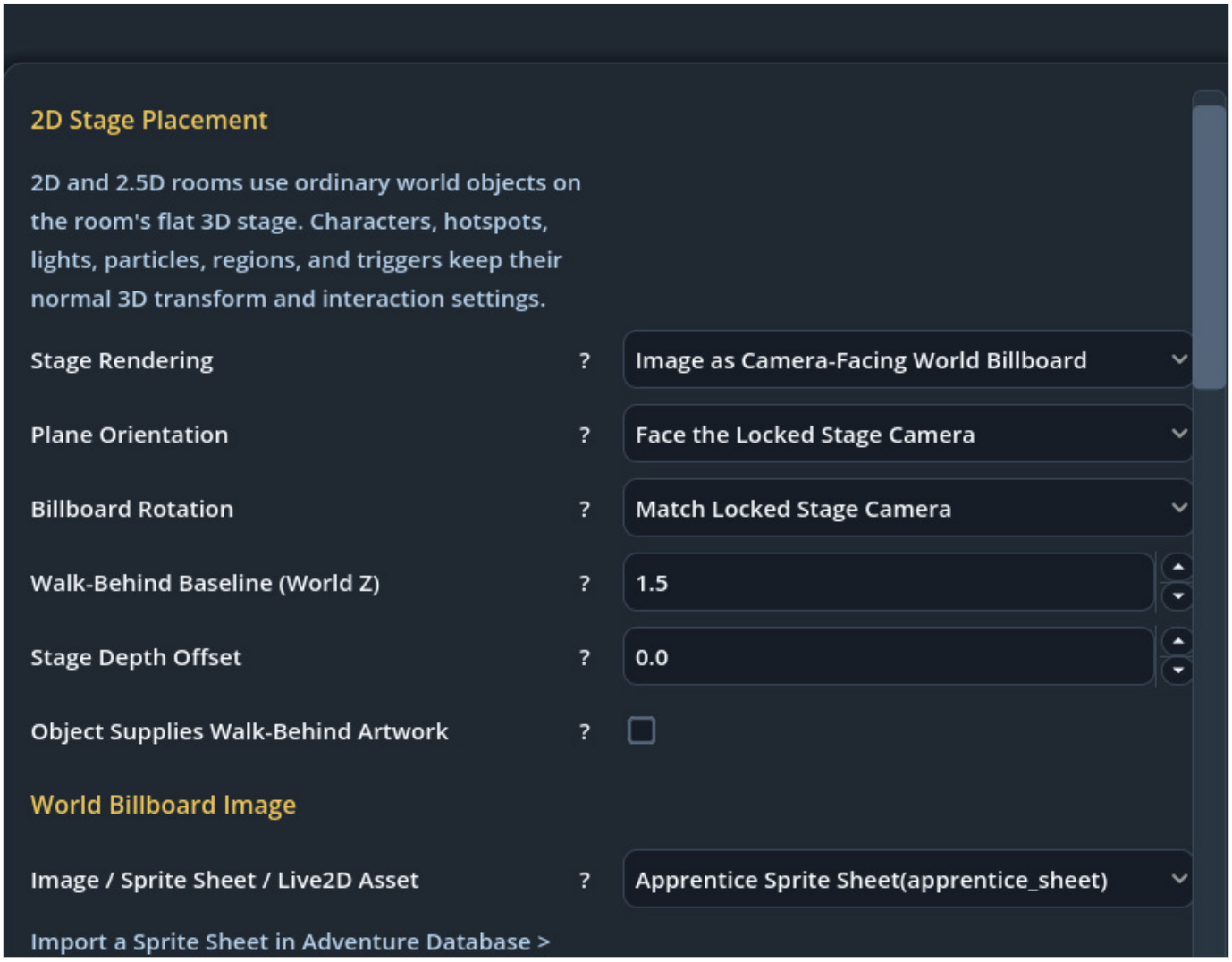


Figure 18. The 2D / 2.5D section belongs to image-based character presentation. Configure it alongside the character's animation states.

For a sprite character, check the visible image bounds, height, and foot alignment in the scene. Transparent space around the artwork can make a correctly sized rectangle look like a very small person. Use the visible-bounds fitting settings when appropriate, then inspect the feet at several animation frames.

A camera-facing sprite in a 3D room still needs usable ground navigation. A flat background scene needs its own stage and navigation setup. The scene's picture dimensions and the walkable area are separate choices, as explained in the Scene Designer guides.

Custom Properties

Use this page for character-specific game data, such as a trust score or a quest flag. Give properties clear names and choose suitable values. A property does not create gameplay by itself: an action or script needs to read or change it.

For example, a `has_seen_key` flag can decide which Talk response to use. Keep the reusable definition's starting value separate from the value that changes during a running game.

Raw JSON (Advanced)

This page exposes the selected record's stored data. Use the ordinary controls whenever they cover the job. Before changing advanced data, save a working copy and note the exact field being changed.

Malformed JSON, a renamed reference, or a frame list pointing to missing assets can break a setup that was otherwise sound. After an advanced edit, apply it, revisit the ordinary page, run Validate, and test the result. Do not use Raw JSON as a substitute for first checking which character is selected.

Scene-only adjustments and attachments

Select the placed character in Scene Designer for position, visual offsets, model fitting, facing, root-motion handling, effects, and pointer feedback. Attachment points connect carried objects or effects to a model, bone, or sprite position. Their placement must suit the actual artwork and animation.

Use the attachment workflow in the Scene Designer guide for equipment, props, and contact points. Lip sync's Face Node and mouth anchors solve a different problem: they locate the face that follows speech.

12. Add a still portrait

A portrait is the close-up used while the character speaks. It can share the scene model or use entirely different artwork. You can give one character several portraits for different moods or situations.

Open **Portraits** in Character Designer. The list at the top holds that character's portrait records. Selecting a different character changes the list.

Start with one image

1. Import the portrait as an Image asset.
2. Select the character and open **Portraits**.
3. Add a portrait and give it an ID such as `calm` and a display name such as **Elias - Calm**.
4. Set **Source** to **Still Image**.
5. Choose the picture in **Image / Live2D Asset**.
6. Mark **Default Portrait** if this should be the normal fallback.
7. Click **Validate and Apply Portrait**, then save the project.
8. Make the character speak in Test Scene and check the result.

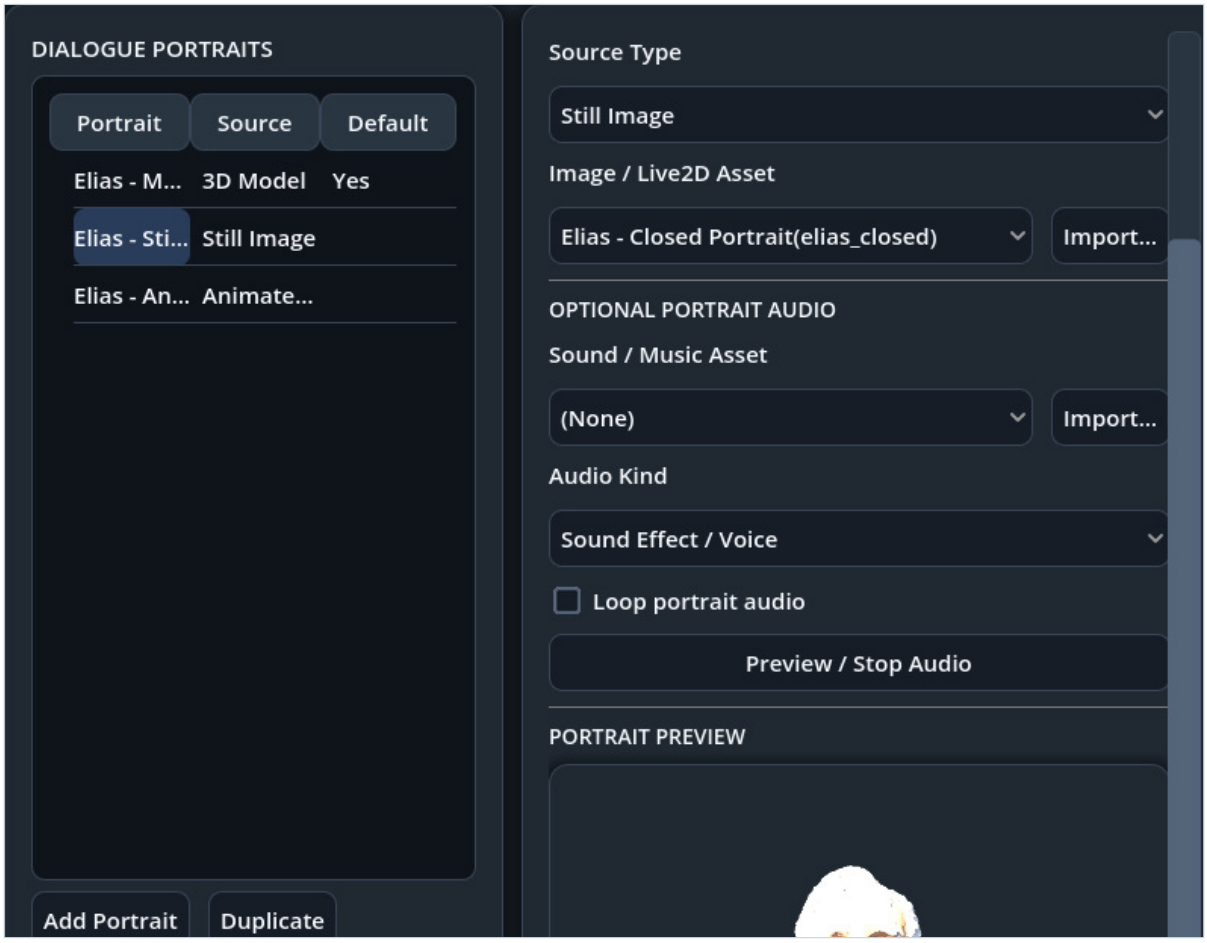


Figure 19. A still portrait references an imported Image asset. This example uses a picture captured from the engine's bundled character model.

Use artwork with enough resolution for the dialogue panel. Leave sensible space around the face, especially when the GUI may crop or fit it differently. Transparent PNG artwork is useful when you want the panel's background visible around the character.

Default, named, and duplicated portraits

The portrait's **ID** is its reference; its display name is for readable editing. A dialogue line can request a particular portrait. When no particular portrait is requested, the engine uses the character's default, with the first available portrait as a fallback when necessary.

Keep one clear intended default. Duplicate a portrait when you want the same framing and settings with a different expression. Change the duplicate's ID and artwork before using it in a line. Reordering the list is not a good replacement for assigning a deliberate default.

A portrait record lives inside its character. A portrait called `angry` on Elias is not automatically the `angry` portrait on Ivo. Choose the speaker first when assigning a line's portrait.

Before you continue
A still portrait should appear correctly before adding lip sync. Check the speaker, default portrait, source image, and dialogue GUI if it does not appear.

13. Animated, video, and Live2D portraits

Animated Image Frames

Choose **Animated Image Frames** when you have a sequence of complete portrait pictures.

1. Import the individual frames as Image assets.
2. Add or select the portrait and set its source to Animated Image Frames.
3. Enter one image **asset ID per line** in the frame list, in playback order.
4. Set the initial/fallback image where appropriate.
5. Choose **Frames per Second**, **Playback Speed**, and **Loop**.
6. Validate and apply, then check the preview.

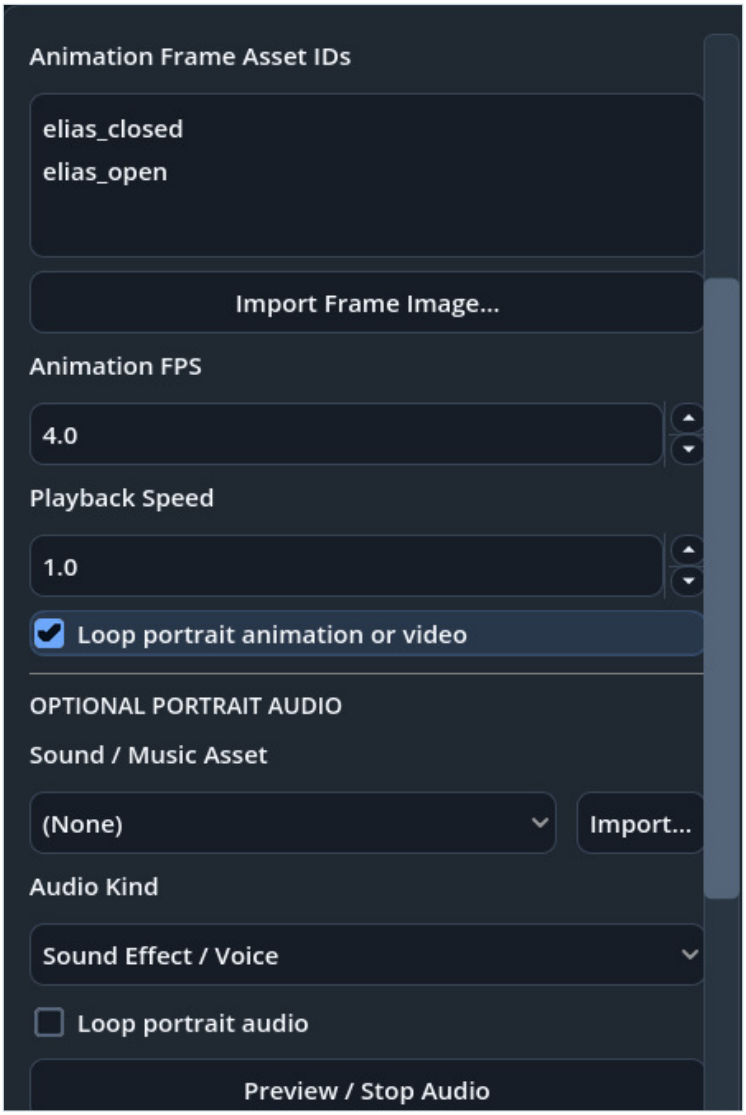


Figure 20. The frame list contains imported asset IDs, one per line. FPS and playback speed affect the portrait's ordinary animation.

All frames should use the same canvas size and face position. If the head jumps between frames, fixing playback speed will not line up the artwork. Two pictures at 4 FPS are useful for a simple demonstration, but a polished idle portrait normally needs purpose-made frames.

This ordinary portrait animation is separate from mouth timing. You can animate blinking or a subtle idle pose, then let lip sync supply a separate mouth layer. If you choose whole-face lip-sync frames, those speech images temporarily replace the portrait's visible frame.

Video

Choose **Video** for an imported video portrait. Select the video asset and check its playback behavior in the intended game build. Use a format supported by your project's video playback setup.

A video of a speaking face already contains its mouth movement. The image-based mouth mappings do not rewrite the face inside that movie. Match the video and voice deliberately instead of expecting automatic facial editing.

Live2D Cubism Model

Choose **Live2D Cubism Model** only for a compatible imported Cubism asset and a working Cubism runtime setup. The engine needs the compatible GDCubism extension and Cubism Core; selecting this source does not install them or build a rig.

First make the model display and animate as a portrait. Then use **Live2D Parameters** in the character's lip-sync mapping. Use the parameter IDs authored into that model. Chapter 20 explains the mouth controls.

Optional portrait audio

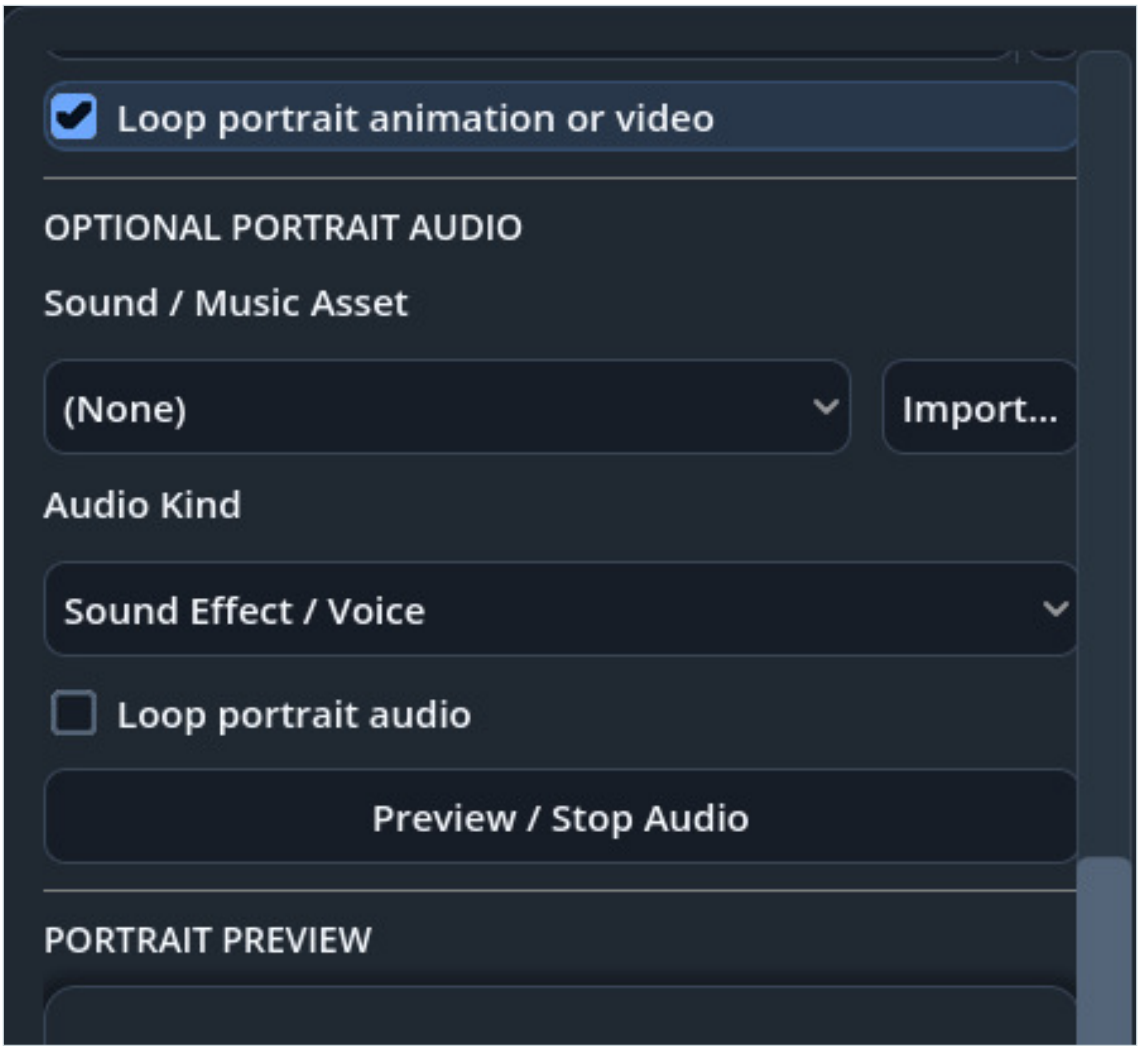


Figure 21. Portrait audio is a separate optional accompaniment. The spoken dialogue line has its own Voice Audio assignment.

A portrait can have an **Audio Asset**, audio kind, and looping choice. Use this for an intentional portrait accompaniment. Leave it empty for a normal conversation unless you have a specific reason to use it.

Assign the spoken recording to the dialogue line or speech action. Putting that same line in a looping portrait-audio slot can produce repeated or competing playback. It also loses the clear connection between the line, its speaker, and its lip-sync track.

14. Make a 3D model portrait

A 3D portrait is a separate view of a model. It has its own framing and optional animation. It does not move the scene character or change the scene's camera.

1. Add a portrait and choose **3D Model**.
2. Select the imported model in the portrait's model field.
3. Decide whether the portrait should use a model animation.
4. If enabled, choose an actual clip from that model.
5. Adjust the portrait framing while watching the preview.
6. Make it the default if appropriate, then validate, apply, and save.

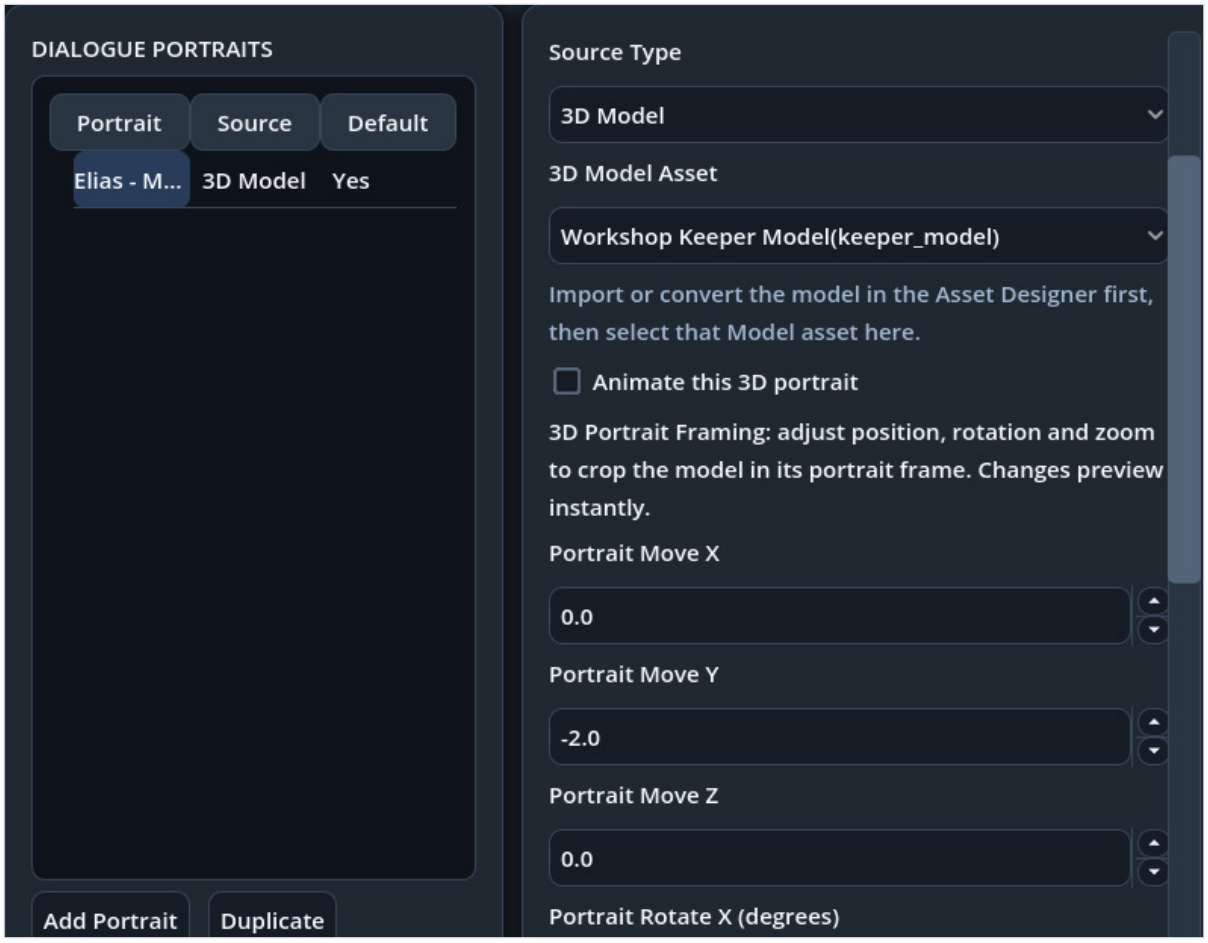


Figure 22. The portrait selects its own source model. It can use the same imported asset as the scene character.

Frame the face, then fine-tune

The preview initially frames the available model. A full-body model often needs a tighter view for dialogue.

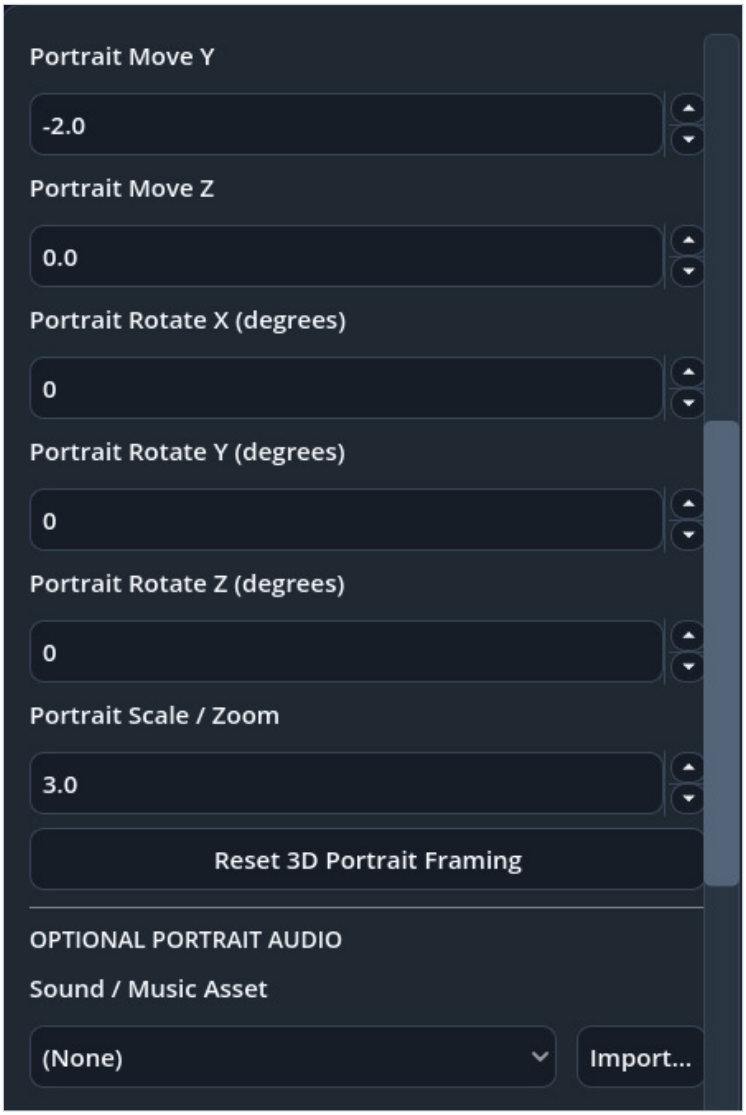


Figure 23. Portrait Position X/Y/Z, rotation, and scale adjust this model view. They do not change its room placement.

Use the framing controls in small steps. Start near the reset values, increase the zoom/scale, then adjust the position until the head and shoulders sit comfortably in view. The offsets are relative to the model's fitted size, so the same number does not mean the same number of scene-world metres on every model.

Portrait rotations use degrees. If the face points away, correct the portrait rotation here. Do not rotate the entire scene character merely to fix its dialogue close-up. Use **Reset Portrait Framing** when an experiment leaves the face lost outside the view.

The pictured Elias model uses Scale 3 and a Y offset of -2 for a close-up. Treat these as values for this asset, not a universal portrait preset. Different models have different origins and proportions.

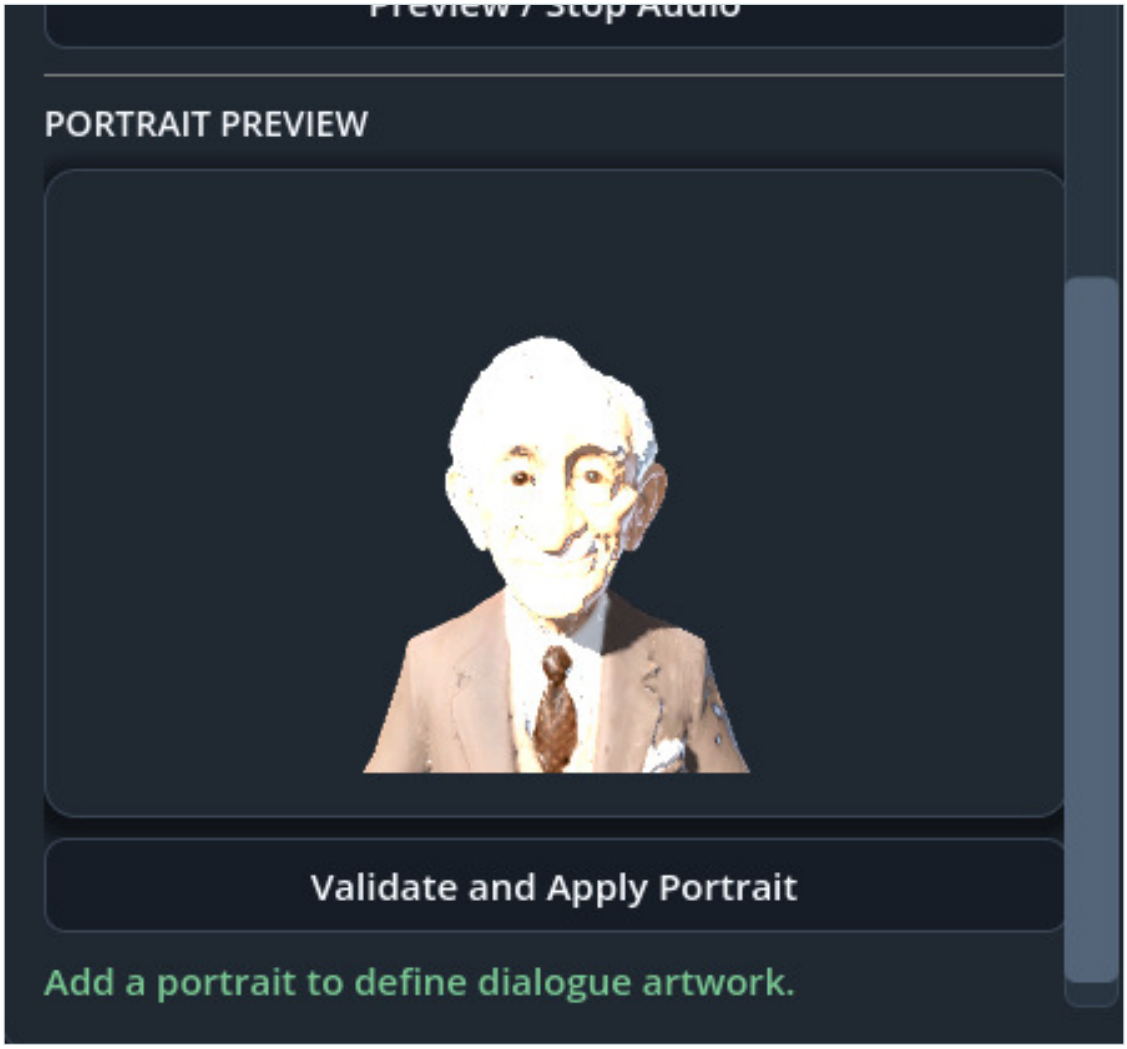


Figure 24. The native portrait preview shows the model with its own lighting and framing. Check the chin and forehead before accepting a close-up.

Portrait animation and facial movement

Model Animated and the chosen model clip control the portrait's ordinary motion. The scene's state ID `talk` is not necessarily a clip called `talk`; select an actual imported clip for the portrait.

Start with a calm clip or an unanimated portrait while checking lip sync. A large nod or a hand moving across the face can make mouth timing difficult to judge. After speech works, add the intended idle or expressive motion and test again.

For a model portrait with lip sync, the **Portrait** mouth mapping must reach that portrait model's facial controls. A working Scene Character mapping alone does not guarantee a separate model portrait will move its mouth.

15. Connect a line, its speaker, and its voice

Begin with one short recording. The example line is Elias saying, **“Every afternoon, that bell calls the town home.”** It is one of the bundled Clockmaker recordings.

Import your own recording as an Audio asset and preview it in Assets. Trim unwanted silence and leave the file long enough to contain the complete line. Give it a clear name and a stable ID, such as `voice_elias_01`.

Set up one line in Dialogue Designer

1. Create or select the dialogue.
2. Add a **Line** node.
3. Choose the **Speaker**.
4. Enter the line's **Text**.
5. Choose a **Portrait**, or leave the default selection when the character's default is appropriate.
6. Set **Voice Audio** to the recording for this exact line.
7. Connect the line to the next node or an End node and set the dialogue's start.
8. Apply/save the dialogue and start it from a Talk response or action list.

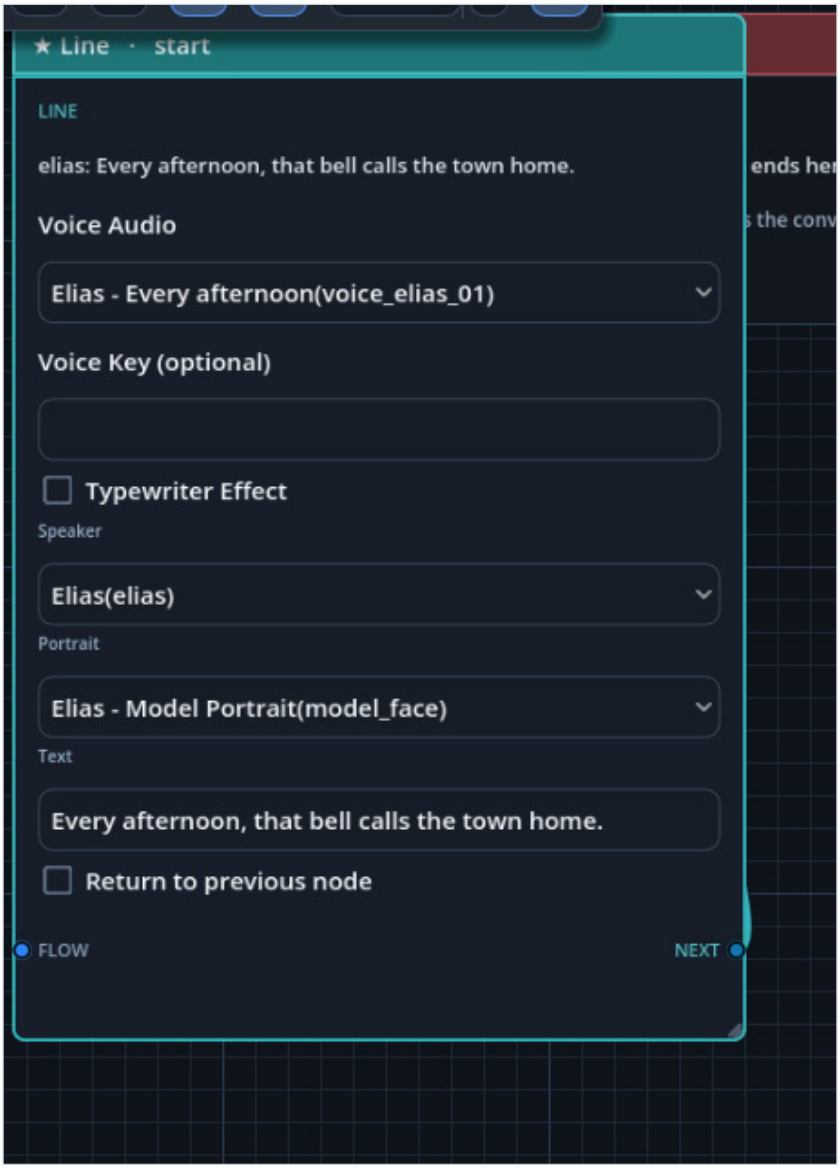


Figure 25. A line ties together Speaker, Portrait, text, and Voice Audio. Choose the speaker before selecting one of that character's portraits.

Use **Voice Key** only when your project's voice/translation setup uses a key to resolve recordings. A direct Voice Audio assignment is easier for the first test. A voice key is not the same thing as the character ID or the portrait ID.

The action-list route

For a single spoken response, add **Character Say** or **Say**, choose the speaker, enter the text, and assign Voice Audio. Use **Player Say** when the current playable character should deliver it. These actions can use the character's default portrait.

A general **Play Sound** action does not identify a speaking character. For controlled voice playback outside a normal speech line, use **Set Voice Speaker** followed by **Play Voice**. Chapter 23 covers that route and its pause/seek controls.

Typewriter text is a different choice

The typewriter checkbox reveals written text a little at a time. Its typing sound can use the default, silence, or a selected sample. This is separate from lip sync.

A series of typing blips is not a phonetic recording of the sentence. If you want text-only talking, choose the character's **Simulated Talking** lip-sync mode and suitable mouth artwork. If you have a spoken recording, let that recording and its track drive the mouth.

16. Put portraits in a dialogue GUI

The built-in dialogue presentation can show the selected character portrait. Use that first while checking the character and recording. It removes a custom GUI from the list of possible causes when something is missing.

If the game uses an authored dialogue GUI, create the image/model portrait area through the GUI Designer's dialogue/system binding workflow. The portrait system uses the system groups `@playerpic` and `@NPCpic` for the player and NPC picture slots. Preserve those exact tags, including their case.

Choose the intended personal dialogue GUI in **Player Settings** when different playable characters need different layouts. Leaving that choice empty allows the game's shared/default presentation.

Test an NPC line, a player line, and a speaker change. Check that the old portrait clears or updates correctly and that its image is not hidden behind another GUI element. A regular decorative Image element has no reason to follow the speaker unless it is bound to the portrait system.

Keep the portrait's rectangle large enough for the face, and check it at your intended game resolution. A carefully framed 3D portrait can still look cropped if the authored GUI gives it a very narrow display area.

> **Before you continue**
> Before working on mouth timing, the line should show the intended text, select the correct speaker and portrait, and play the correct recording in your own test.

17. Choose how speech will drive the mouth

Lip sync has two jobs. A **timing track** says which mouth cue belongs at each moment in the recording. A **mapping** says which picture, morph, or other control should represent that cue on this character.

The track belongs to the Audio asset. The mapping belongs to the character. You can reuse a good character mapping across many recordings, but each recording needs its own timing.

Turn it on deliberately

1. Select the character and open **Lip Sync**.
2. Check **Enable Lip Sync**.
3. Choose **Speech Timing**.
4. Choose **Lip-sync Targets**: Scene Character, Portrait, or Both.
5. Choose a **Missing Track Fallback**.
6. Start with Rest Mouth Shape **X**, Mouth Strength **1**, and a small transition value.
7. Configure the Scene Character and/or Portrait mapping below.
8. Click **Apply Changes** and save the project.

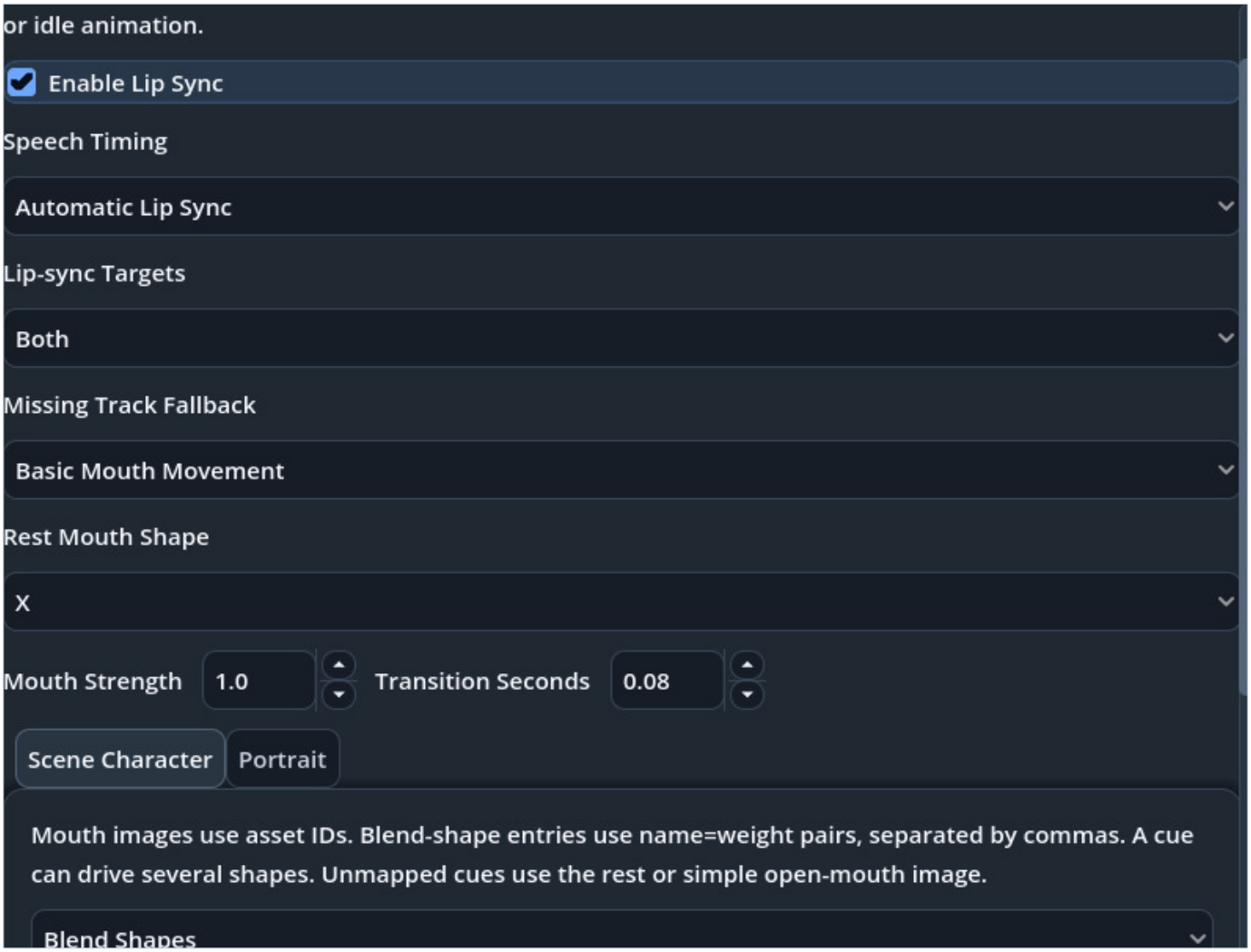


Figure 26. Lip sync is opt-in. This example uses Automatic Lip Sync, Both targets, and Basic Mouth Movement as the missing-track fallback.

Older characters can have lip sync disabled. Do not assume that updating the engine or importing a voiced line enables it for every existing character.

Speech Timing	Use it for
Automatic Lip Sync	A recorded line with prepared mouth cues. Generate and review the track before play.
Basic Mouth Movement	Simple opening and closing based on the recording's loudness. It does not recognize phonemes.
Manual / Imported	A recorded line with cues you edited or imported.
Simulated Talking	Text without a spoken recording. It creates talking-like motion, not audio-synchronized phonemes.

The automatic setting does not run speech recognition during every conversation. It follows the timing prepared during authoring. Games use the saved track.

Targets, fallback, strength, and transition

Choose **Both** when the face in the room and its dialogue portrait should move together. Their mappings can differ: a scene model might use Blend Shapes while a painted portrait uses Whole Frames.

Missing Track Fallback decides what happens when prepared timing is unavailable. Off gives you no fallback speech motion. Basic follows loudness when voice audio is available. Simulated talking is useful for text-only lines; when a recording is present, that fallback uses basic audio-driven movement.

Mouth Strength scales the configured facial motion. Start at 1, then reduce an exaggerated result. **Transition Seconds** smooths changes between cues. Too much smoothing can make a quick line feel late or muddy. Check fast consonants as well as long vowels.

A mouth resting open usually calls for checking the rest cue and its mapping, not simply reducing every animation speed.

18. Map a 3D face with blend shapes

A blend shape is an authored deformation of a mesh: an open mouth, rounded lips, a smile, and so on. The model must already contain these shapes. Lip sync cannot create a facial rig from a plain mesh.

The bundled models used here contain `Mouth_Open`, `Mouth_Wide`, and `Mouth_Round` on `Mesh1_0`. Those are names from these models. Your own asset can use entirely different names.

Find the controls your model actually has

Open **Preview Lip Sync**, select the character, and click **Inspect Mouth Targets**. The result lists available mesh blend shapes and skeleton bones. Scroll the result when the skeleton list is long.



Figure 27. Inspect Mouth Targets reports the imported model's actual mesh and bone names. Use those names when filling in the mapping.

If there is no suitable facial mesh or jaw, choose an image-based method or revise the model in your art program. A Head bone is not automatically a hinged jaw.

Fill in the Scene Character mapping

1. Return to Character Designer's Lip Sync page.
2. Select the **Scene Character** mapping tab.
3. Choose **Blend Shapes**.
4. Put the relevant mesh node's unique name or relative path in **Face Node**. Here it is `Mesh1_0`.

5. Enter each cue's shape weights as `name=weight` pairs, separated by commas.
6. Leave the rest/closed cues empty when the mesh's zero-weight pose is the intended closed mouth.
7. Apply, open the preview again, and seek through the line.

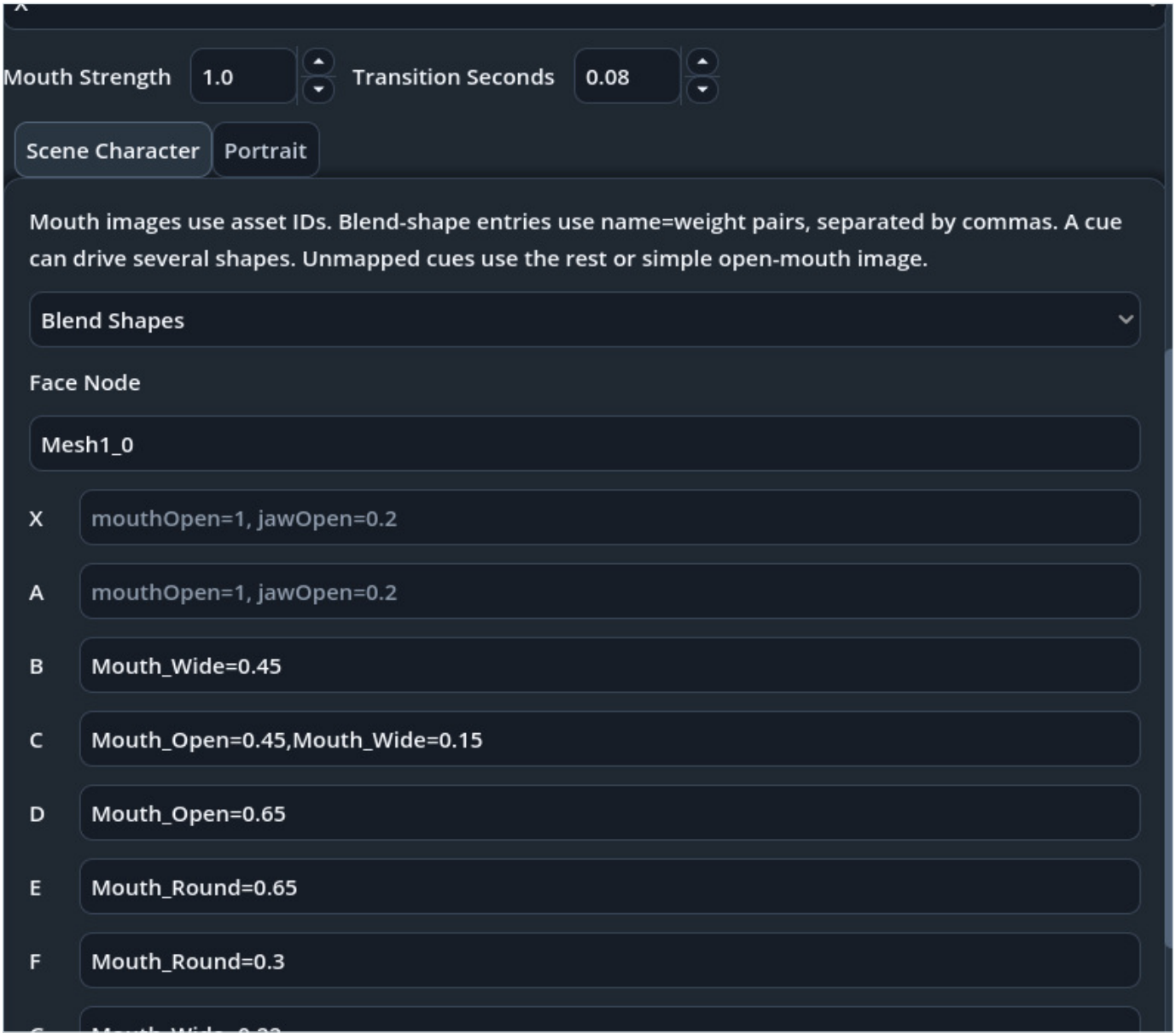


Figure 28. Blend Shapes maps the cue letters to named mesh controls. Gray placeholder examples in empty fields are not saved mappings.

Start with a restrained mapping. The example uses:

Cue	Entry for this bundled model
X	Leave empty; use its neutral face.
A	Leave empty; use closed lips.

Cue	Entry for this bundled model
B	`Mouth_Wide=0.45`
C	`Mouth_Open=0.45, Mouth_Wide=0.15`
D	`Mouth_Open=0.65`
E	`Mouth_Round=0.65`
F	`Mouth_Round=0.3`
G	`Mouth_Wide=0.22`
H	`Mouth_Open=0.3`

This is a practical approximation for three available controls, not a claim that three shapes reproduce every phoneme perfectly. A more detailed facial rig can use more carefully authored combinations. Use Inspect Mouth Targets and preview your actual mesh.

Map the portrait separately

Choose the **Portrait** mapping tab. If its 3D portrait uses the same model, begin with the same Face Node and cue entries. If it uses another model, inspect that model's controls and map its names instead.

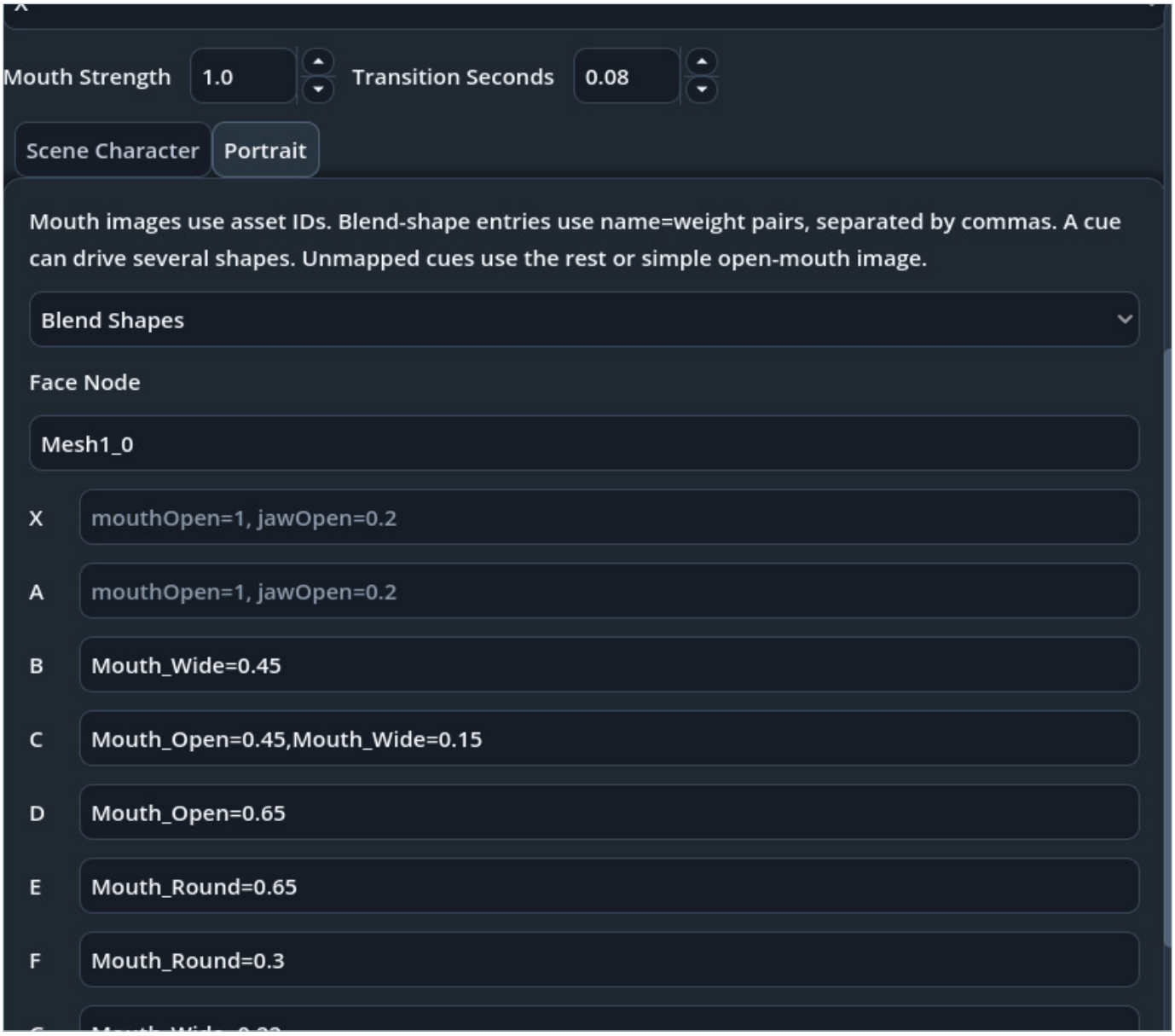


Figure 29. The Portrait mapping is separate from Scene Character. Both targets can share timing while using different facial controls.

Set the overall targets to **Both**, apply, and preview both faces. Lip sync works on the facial controls without replacing the character's walking or idle state. Check a speaking character while standing and while moving before trusting the final result.

19. Map a drawn portrait or sprite face

You do not need a 3D facial rig to make a portrait speak. You do need suitable artwork. Choose between complete face images and a separate transparent mouth layer.

Whole Frames: the straightforward starting point

Prepare at least two aligned pictures: the normal closed-mouth face and an open-mouth version. Import both as Image assets. Keep their dimensions, head position, and framing identical.

1. Make a working Still Image portrait using the closed-mouth picture.
2. Open **Lip Sync**, choose the **Portrait** mapping tab, and select **Whole Frames**.
3. Assign the closed picture to **X**.
4. Assign the open picture to **D**.
5. Add images for the other cues when you have suitable artwork.
6. Apply the settings, open Preview Lip Sync, enable **Portrait Preview**, and choose the portrait.
7. Seek and play the line, then check the result in the game.

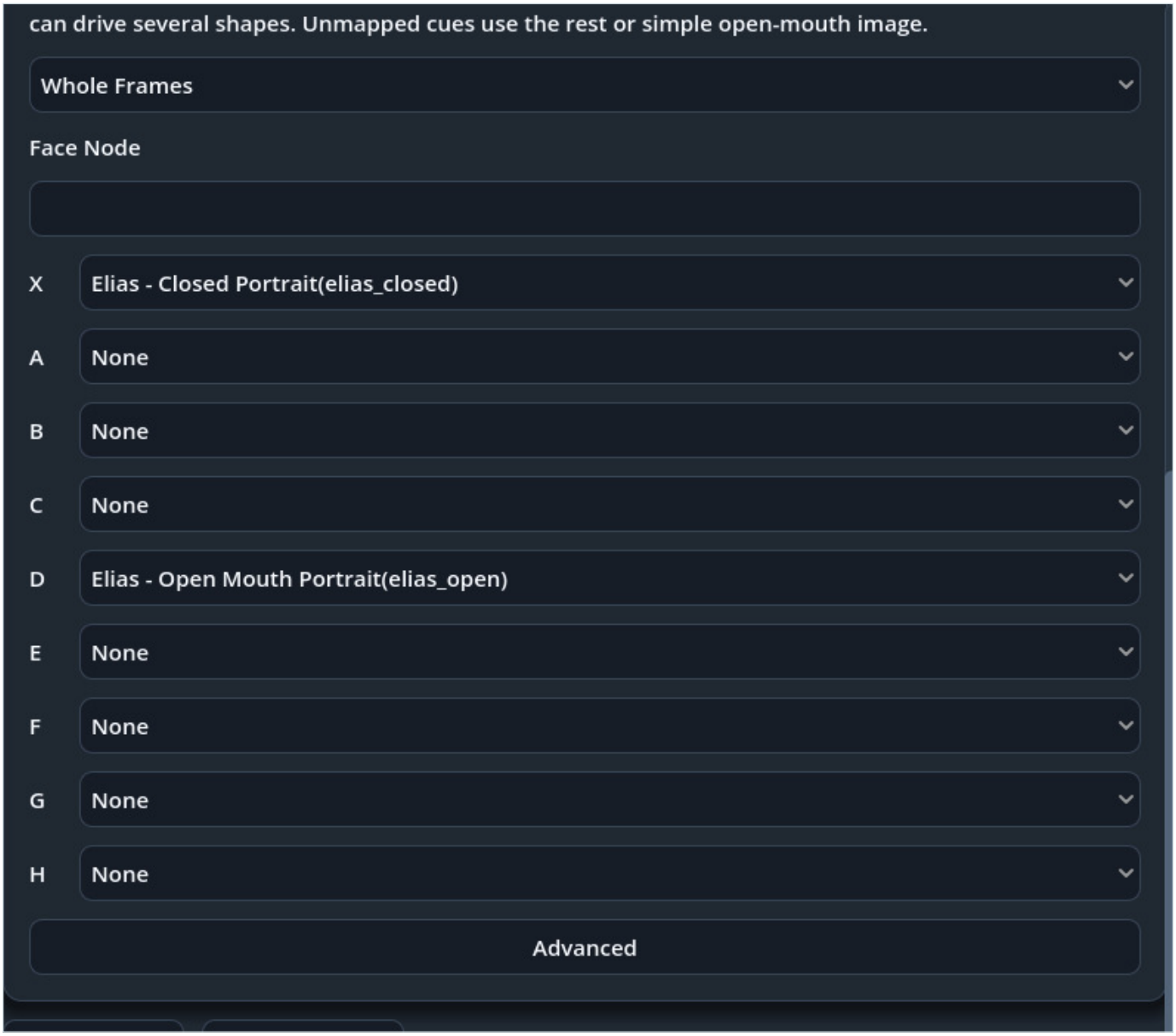


Figure 30. Whole Frames selects complete imported pictures for the cue letters. X and D provide a simple closed/open starting point.

Unmapped cues can use the rest or simple open-mouth fallback. A two-frame mouth will look less precise than a complete set, but it is a useful first test of the connection.

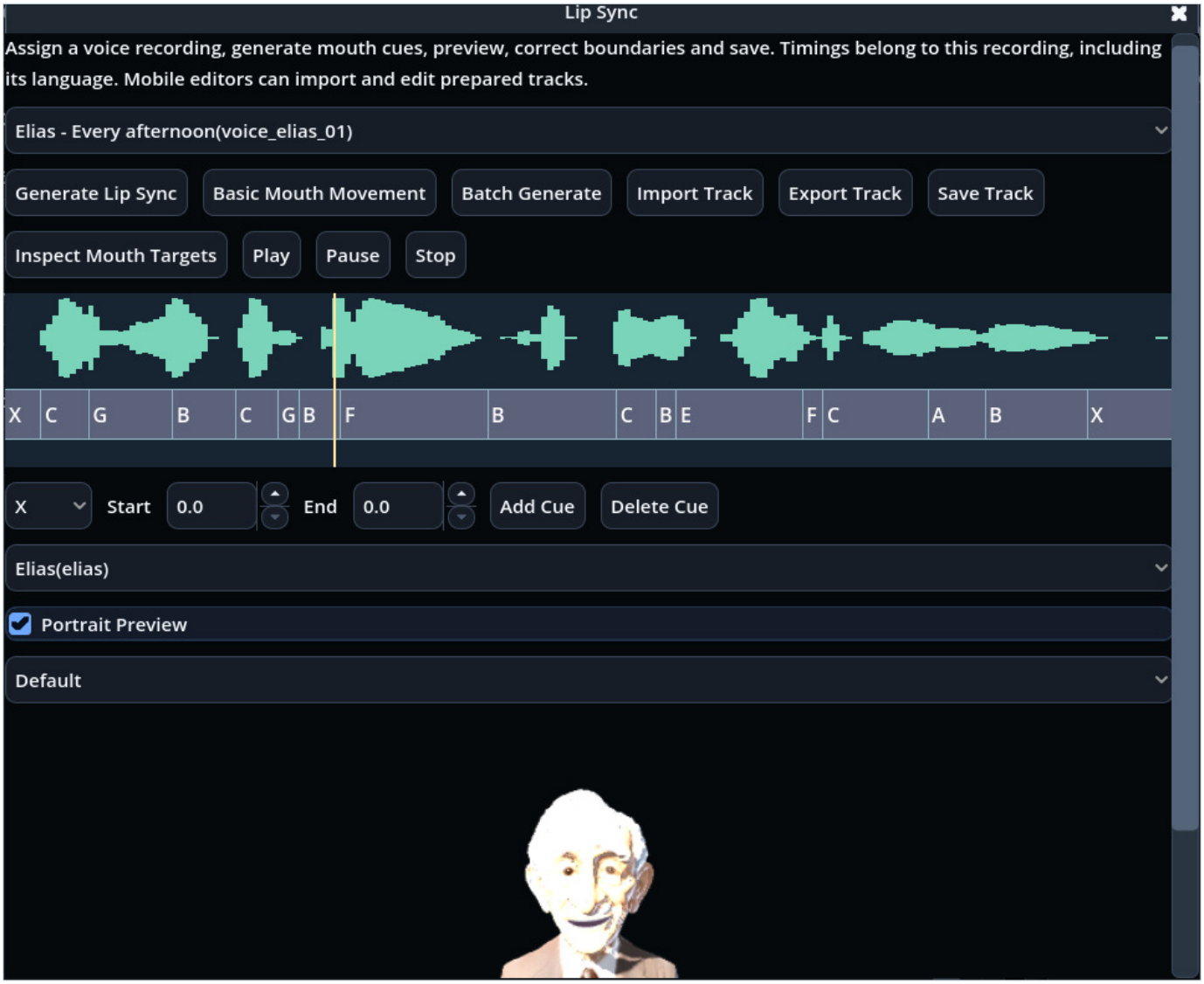


Figure 31. The same saved recording can drive a still portrait through whole-face images. This is separate from the model's scene mapping.

A Whole Frames mapping replaces the visible frame during speech. If your portrait normally blinks or changes pose, make sure the speech images match that pose. Otherwise the entire face will appear to jump when talking starts.

For an image-based scene character, use the **Scene Character** mapping tab. Check all the directions or animation poses used in the game. One front-facing head picture will not suit every frame of a turning, full-body sprite.

Separate Mouth Layer

Use **Separate Mouth Layer** when the face/body should keep its normal animation and only the mouth should change. The cue pictures should be small transparent mouth images, not complete face portraits.

1. Import the transparent mouth images.

2. Choose Separate Mouth Layer in the relevant mapping tab.
3. Assign a closed/rest image to X and an open image to D, followed by other cue artwork.
4. Open **Advanced** and adjust Mouth Anchor X/Y/Z and Mouth Scale.
5. Preview at rest and in motion; correct the alignment before refining the timing.

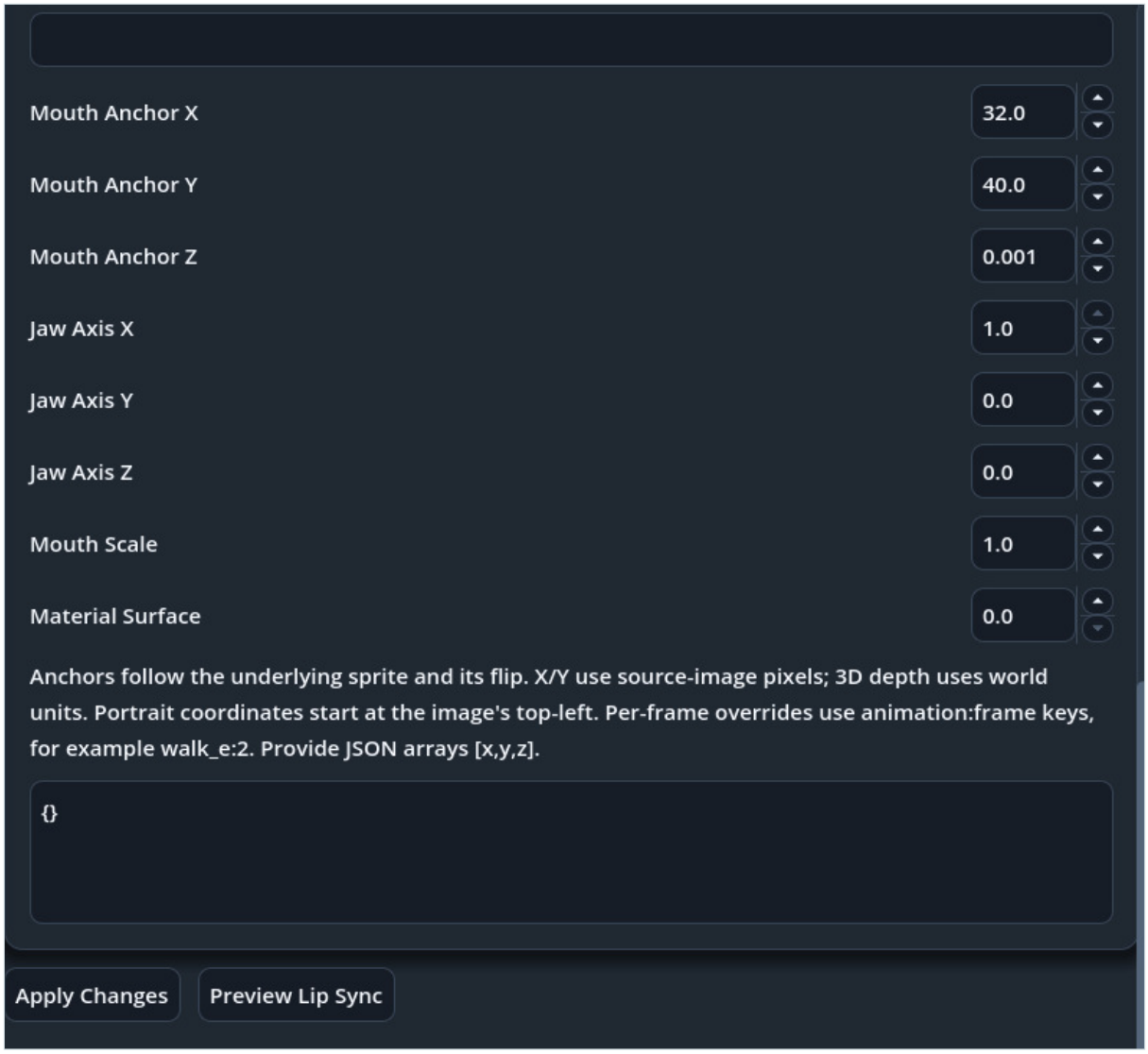


Figure 32. Advanced exposes the mouth anchor, scale, and per-frame overrides. These are configuration controls; this reference view has no mouth artwork assigned.

X and Y anchors use source-image pixels. Portrait coordinates begin at the picture's top-left. For a mouth layer in 3D, depth uses world units. Make small changes and inspect the actual result.

The mouth follows the underlying sprite and its flip. If the head changes position between frames, use the per-frame anchor overrides. They are a JSON object with `animation:frame` keys and three-number arrays, for example `{"walk_e:2":[32,40,0.001]}`. Use the actual animation name and frame number in your character; the example key is a format illustration. For animated portrait frames, the keys use `normal:0`, `normal:1`, and so on.

Keep {} when no overrides are needed. Invalid JSON or arrays with the wrong number of values will prevent the mapping from applying. A consistent sprite sheet is much easier to maintain than dozens of corrective offsets.

Know what the letters mean

Cue	Approximate mouth shape
X	Rest or silence.
A	Closed lips, such as M, B, or P.
B	Clenched teeth.
C	Open mouth.
D	Wide open mouth.
E	Rounded lips.
F	Puckered lips, such as W or OO.
G	Lower lip against upper teeth, such as F or V.
H	Raised tongue, such as L.

These letters are visual cue labels. They are not the literal written letters from the sentence. Do not map cue B only to words containing the letter “B.”

20. Jaw bones, mouth textures, and Live2D

These methods suit assets prepared for them. They are alternatives to the blend-shape and picture workflows, not extra boxes every character needs filled in.

Jaw Bone

Choose **Jaw Bone** when the model contains an actual jaw bone whose rotation opens the mouth.

1. Inspect the imported skeleton and identify the correct jaw bone.
2. Choose Jaw Bone in the mapping method.
3. Enter the target node/path if needed and the exact Jaw Bone name under Advanced.
4. Choose the **Jaw Axis** and a modest **Jaw Angle** appropriate to that rig.
5. Preview a sustained open cue and verify that the jaw, rather than the whole head, moves.

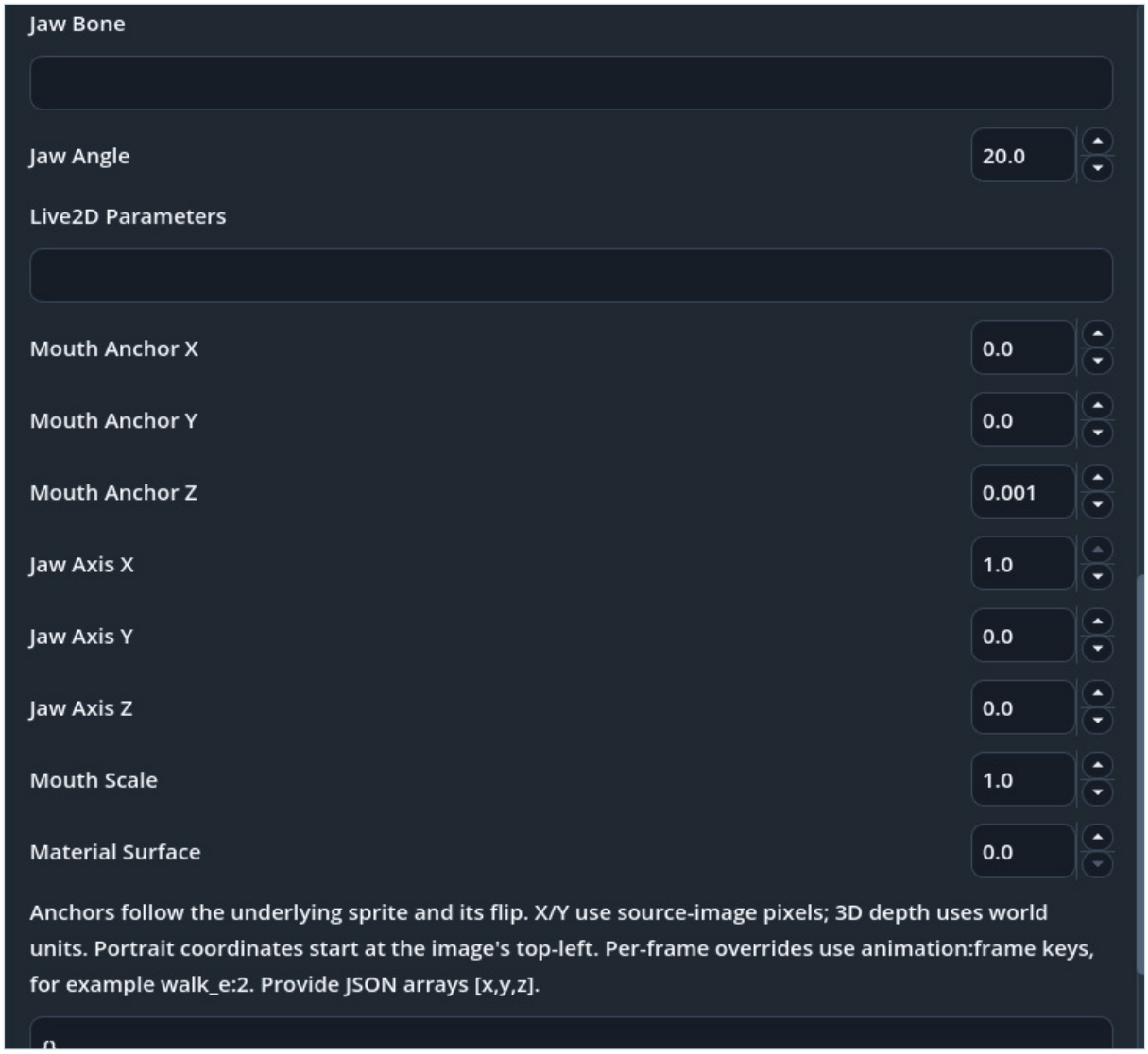


Figure 33. Jaw settings need a real jaw in the imported rig. The empty bone field here is deliberate; the pictured demo uses blend shapes instead.

A wrong axis can twist the jaw sideways. A Head bone can tilt the entire head. Neither becomes a jaw just because its name was entered into the field. The bundled facial example in this guide is tested with blend shapes, not a jaw setup.

Mouth Texture

Choose **Mouth Texture** when the mouth is represented by textures on a prepared mesh/material surface. Assign the cue image assets and point Face Node to the correct mesh. In Advanced, **Material Surface** is the surface index, beginning at zero.

The supported texture replacement expects an appropriate StandardMaterial3D surface. A custom shader may need its own integration. Replacing a whole face or body material unintentionally is a sign that the wrong surface or mesh was selected.

Live2D Parameters

Use **Live2D Parameters** with a compatible Cubism model and runtime extension. Enter the authored mouth parameter IDs, separated by commas. Leaving the field empty lets the system look for the model's LipSync parameter group, with `ParamMouthOpenY` as a fallback.

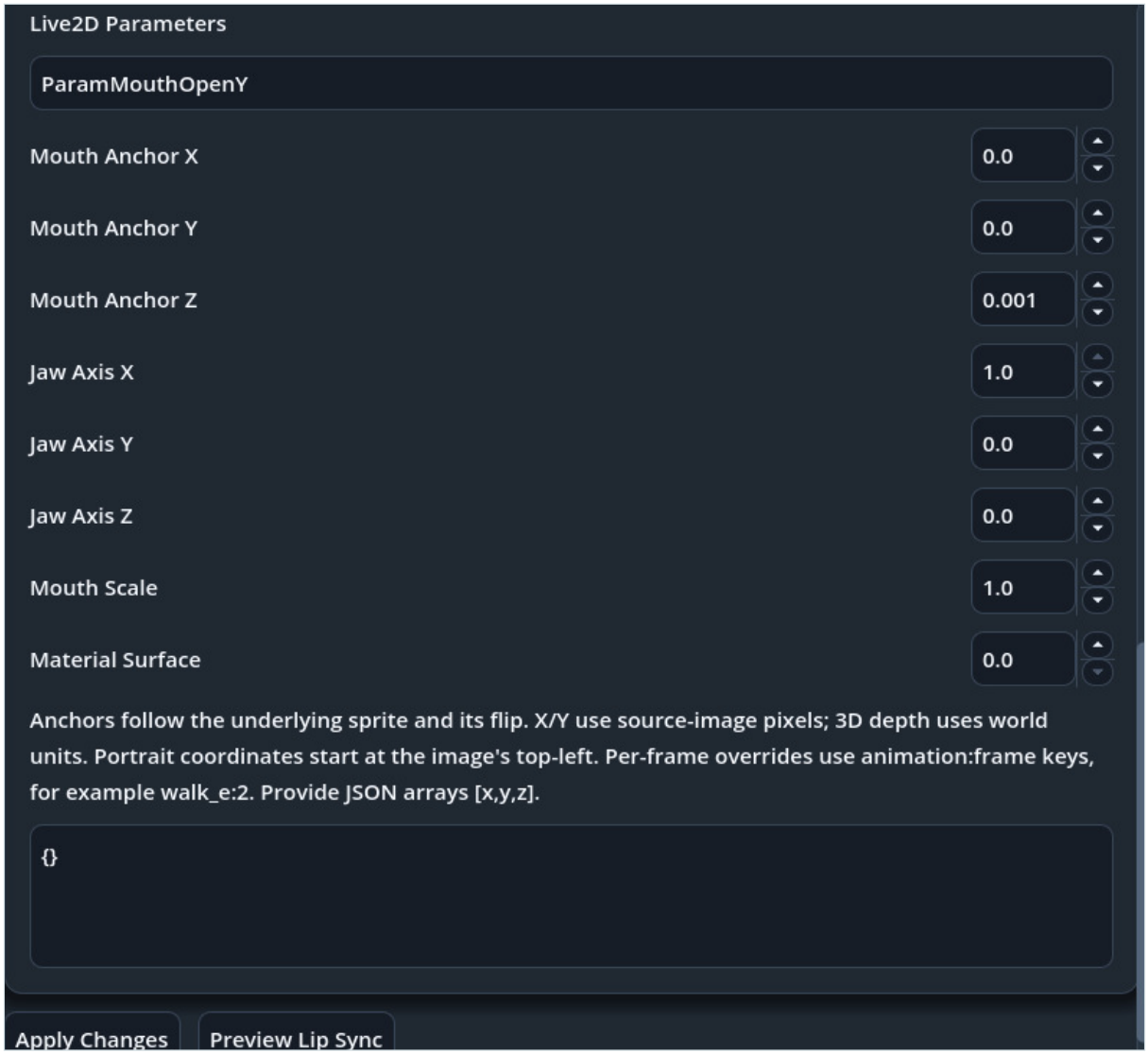


Figure 34. The parameter field takes model parameter IDs. A typed name alone cannot supply a missing Cubism rig or extension.

Only the relevant mouth parameters are held during speech; other authored motion can continue. Check the model outside lip sync first, then test mouth motion with a single recording. This guide does not claim that the optional Cubism extension or a separate Live2D model was installed for the screenshots.

21. Generate, preview, and correct a timing track

Now connect the recording to the face. Keep the character's mapping applied before opening the track preview, so the preview reads the saved configuration.

Open the recording

From the character's Lip Sync page, choose **Preview Lip Sync** and select the required voice Audio asset. Select the character below the timeline. If the dialog opens on a different recording, change the top selector before generating anything.

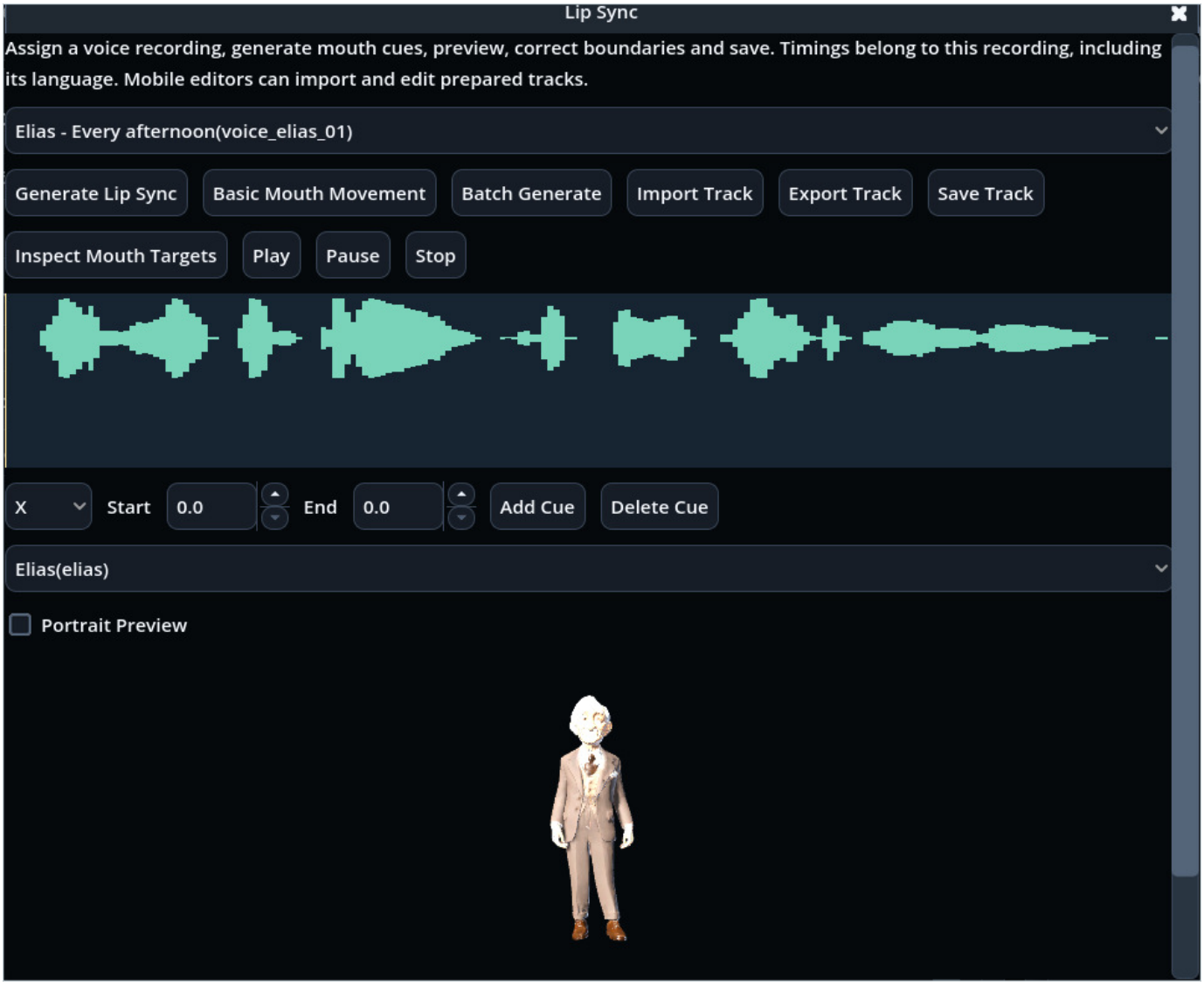


Figure 35. The waveform shows that audio was decoded. The empty cue lane shows that this recording does not yet have prepared mouth cues.

The waveform alone is not a lip-sync track. It shows loudness over time. The cue lane beneath it is where the X/A–H sequence appears.

Automatic generation

1. Select the correct recording and character.
2. Open **Advanced** and check **Recording Language**.
3. Leave **Rhubarb Executable** empty to use the bundled desktop recognizer, unless you deliberately configured another compatible executable.
4. Optionally enter the exact spoken sentence in **Dialogue text**.
5. Click **Generate Lip Sync** and wait for the result.
6. Play it, inspect the face, and correct cues that need attention.
7. Click **Save Track**, then save the project.

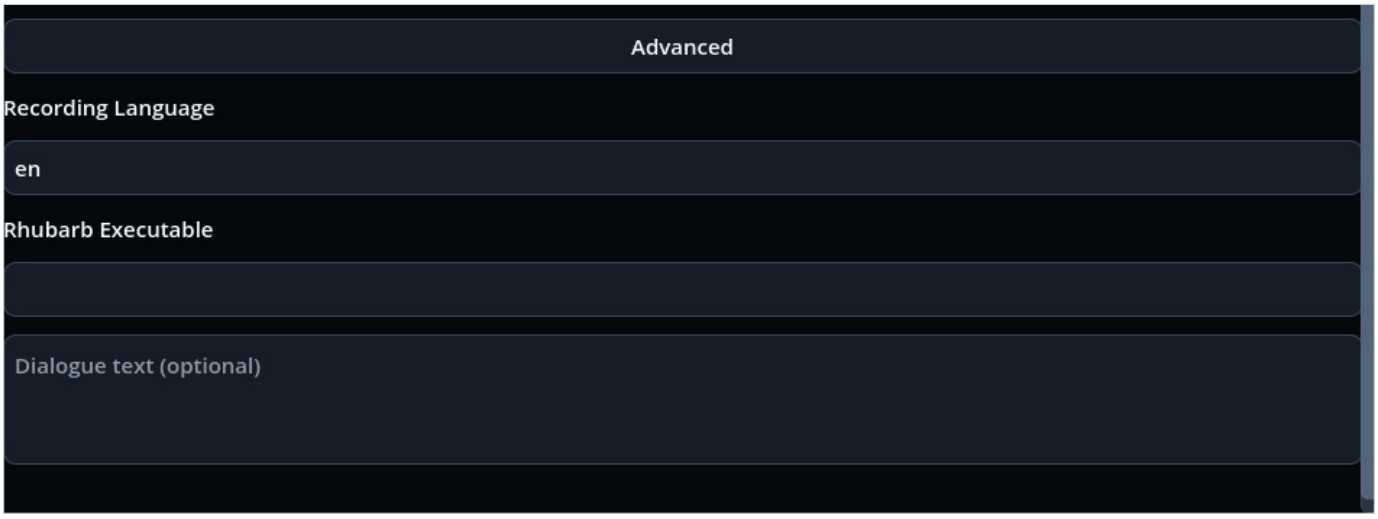


Figure 36. Recording Language refers to the voice, not the subtitle translation. Advanced also holds the optional recognizer path and transcript.

The desktop generator works with decodable recordings up to ten minutes long. Use individual speech lines rather than a whole scene recording. English uses word recognition; other recording languages use less precise phonetic recognition and deserve careful review.

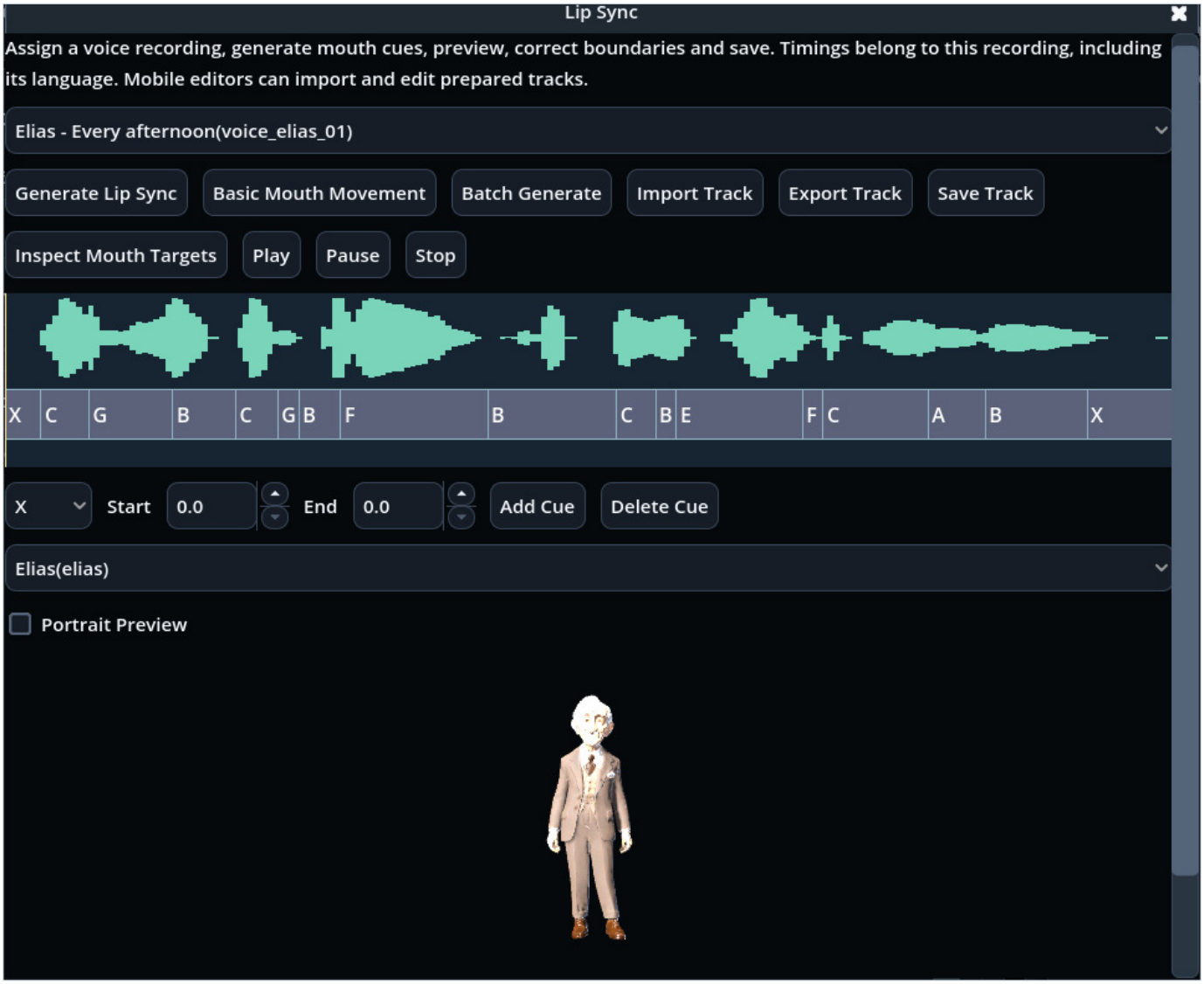


Figure 37. The native generator produced 17 cues for this short demo line. The letters now occupy timed blocks beneath the waveform.

Basic Mouth Movement is an alternative generation route based on loudness. Use it for a simple open/closed result or when phonetic recognition is unsuitable. It does not distinguish a rounded vowel from an F/V shape.

Read and edit the timeline

Click the waveform to seek. Click a cue block to select it. Its shape, **Start**, and **End** appear below the timeline.

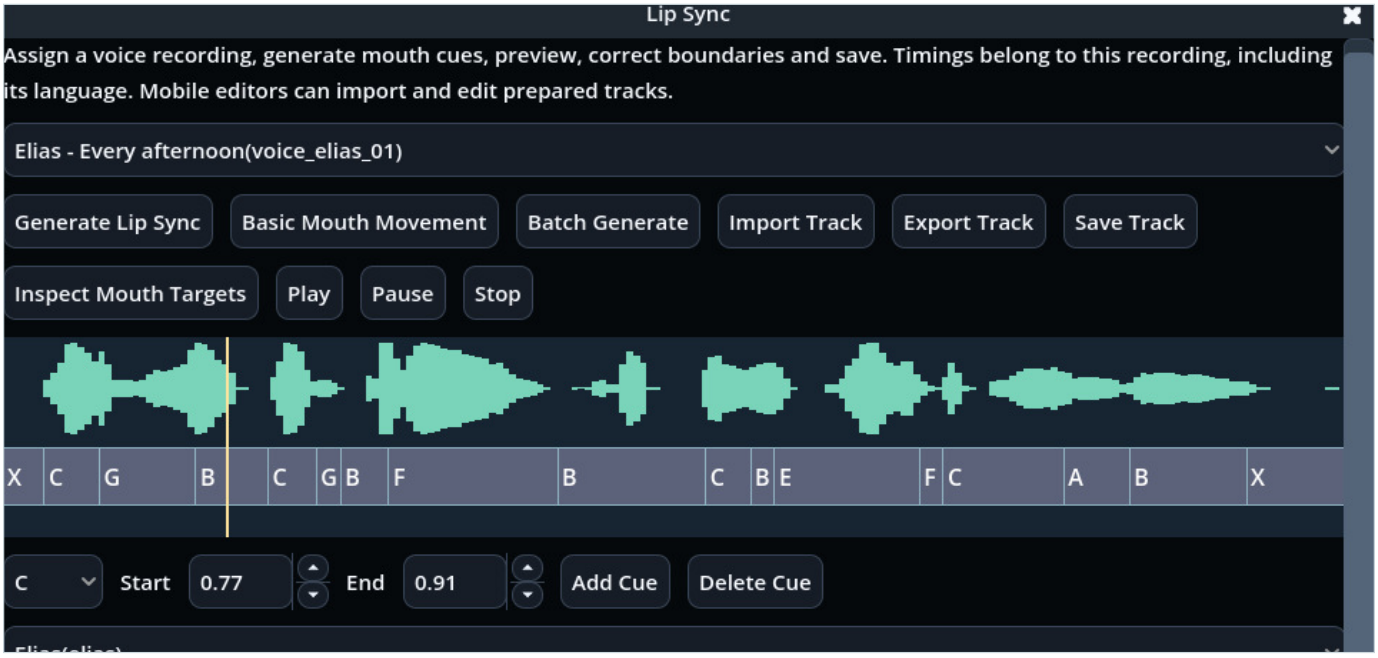


Figure 38. A selected cue exposes its shape and start/end seconds. Use the waveform and the spoken sound to judge the boundary.

For a correction:

1. Replay a short section and locate the wrong movement.
2. Select its cue block.
3. Choose the correct X/A–H shape.
4. Adjust Start and End in seconds, or drag a cue boundary on the timeline.
5. Replay the transition into and out of the edited cue.
6. Save Track once the correction is right.

Keep cues in chronological order and avoid overlapping intervals. The editor checks the track when you change it. Use **Add Cue** to fill a suitable gap and **Delete Cue** to remove a wrong block. If there is no space for a new cue, shorten or remove an existing one first.

A gap returns to the rest cue. Use that deliberately for silence; do not stretch an open-mouth cue through a breath simply to fill the lane.

Preview the portrait as well

Enable **Portrait Preview** and choose Default or a particular portrait. Seek to an open cue, then to a rest cue. Turn Portrait Preview off to inspect the scene face again.

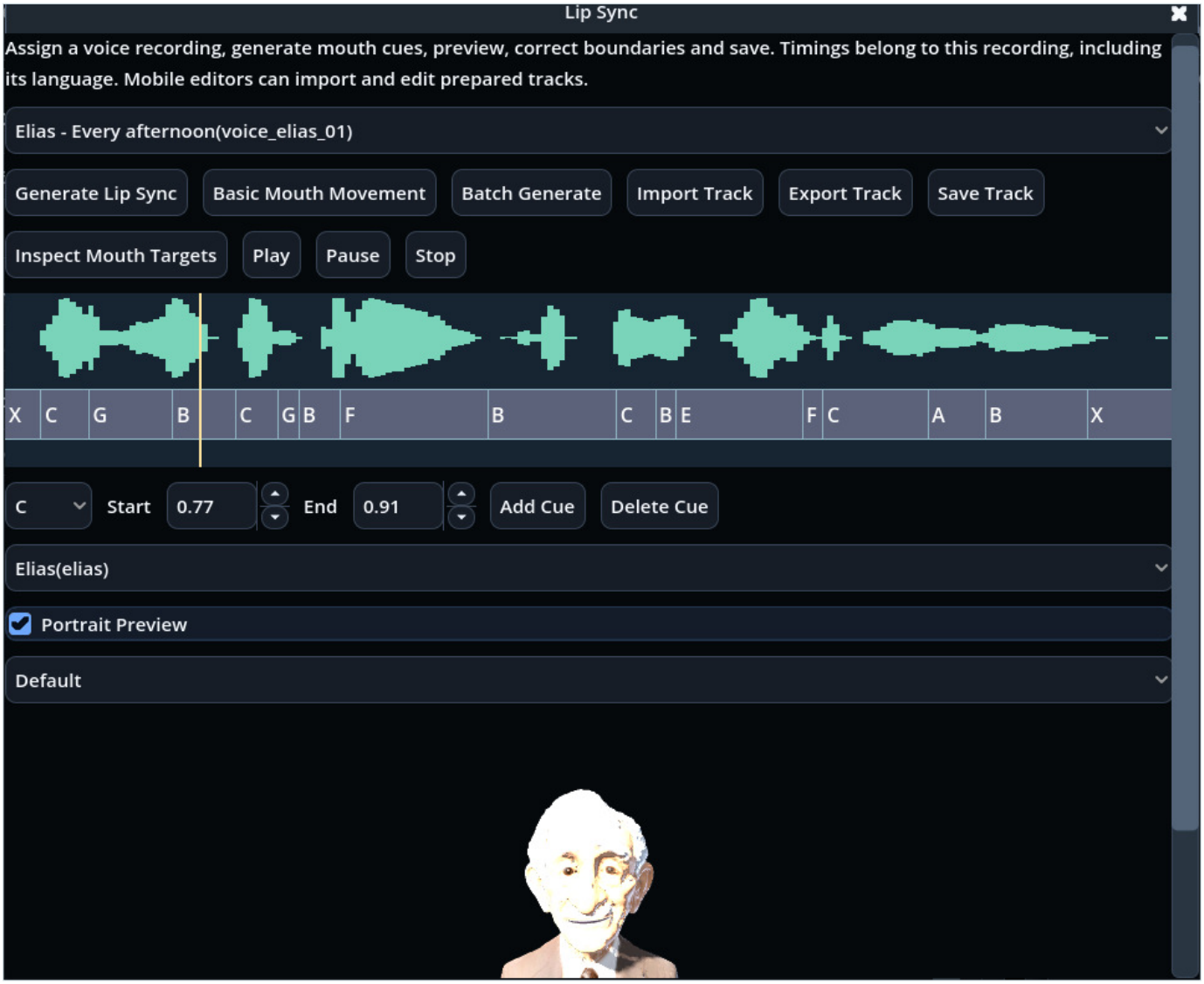


Figure 39. Portrait Preview uses the portrait's own source and framing. Check it separately from the full scene model.

Play, **Pause**, and **Stop** are for reviewing the recording. Seeking is particularly useful when you want to inspect a single mouth shape. Remember to listen on your own machine as well as watching: a technically valid cue sequence can still feel mistimed.

Save Track is not Export Track

Save Track attaches the timing to the Audio asset in the project. **Export Track** writes an external JSON copy. **Import Track** brings compatible native or Rhubarb mouth-cue JSON into the editor, where you should preview it and then save it to the asset.

Save the project after saving the track. An exported JSON file sitting elsewhere on the computer is not a replacement for the asset's saved timing.

22. Manage many recordings and translated speech

Batch Generate

Use **Batch Generate** when the character mappings are ready and you have several voice recordings to process. Select only the intended speech assets.

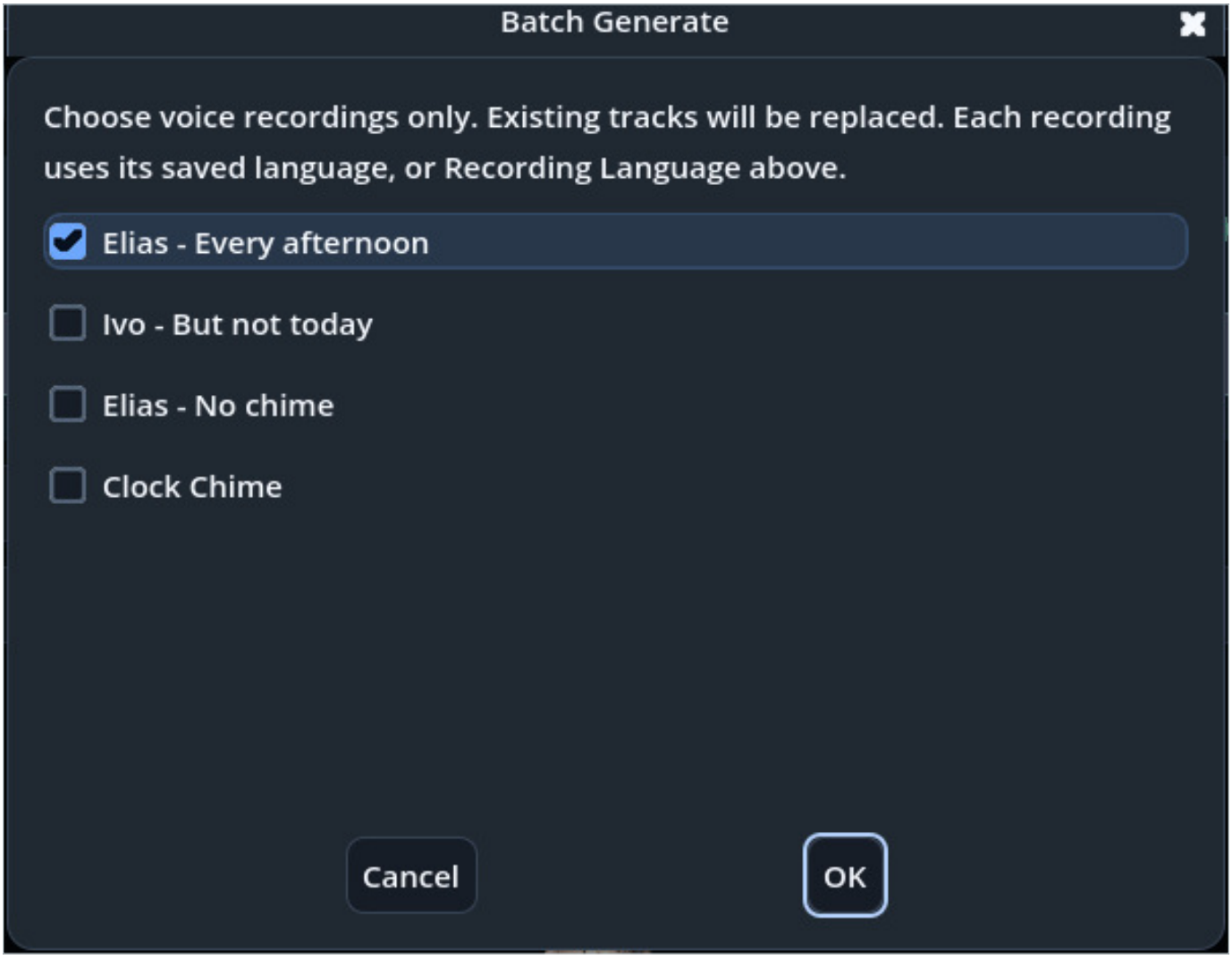


Figure 40. Batch Generate lists Audio assets. Choose voice recordings carefully: existing tracks on the selected recordings will be replaced.

1. Check the recording language before starting.
2. Open Batch Generate and select the desired voice files.
3. Confirm the batch and wait for it to finish.
4. Review any reported per-recording errors.

5. Open the results one by one, preview them, and correct weak sections.
6. Save the project.

Batch generation saves tracks for the processed assets. It still needs a review pass. Do not select footsteps, music, and ambient loops simply because they appear in the same list.

Replacing a voice file

Timing belongs to the exact recording, including its pauses and pace. If you replace or re-record the voice, regenerate or revise its track. The saved audio fingerprint helps detect a mismatch.

Changing the subtitle's punctuation does not necessarily change the timing, but re-recording the sentence does. A more polished voice take can be just a fraction longer and still make the old mouth motion look wrong.

Languages and voice keys

Treat each language's voice recording as a separate audio source with its own track. Do not copy English cue times to a translation spoken at a different speed.

Use the project's translation/voice-key workflow to resolve the appropriate recording. Set Recording Language to the language being spoken while generating that recording's track. The character's art mapping can usually be reused: the images or facial shapes still represent the same visual mouth poses.

After changing the game's language, test the actual spoken line, portrait, and track together. A successful subtitle import alone does not prove that the translated voice asset is assigned.

Preparing elsewhere and importing

Prepared tracks can be imported and edited in the editor, including on platforms where the desktop recognizer is unavailable. The released game follows saved cues; it does not need to run Rhubarb during play.

Keep the external JSON next to the matching recording in your production archive. Use clear filenames and language labels so the next person does not import a valid track onto the wrong take.

23. Use actions and scripts during play

Character Designer stores the starting setup. Actions and scripts decide when a character speaks, moves, changes state, or has lip sync enabled during the game.

A useful first action list

On Elias's Talk response, make a short list:

1. Add **Enable Lip Sync**, choose Elias, and set Enabled on.
2. Add **Character Say**. Choose Elias, enter the sentence, and select its Voice Audio.
3. Choose whether the line should wait for the player to continue.
4. Apply the list, save, and test it through Talk.

The first step is optional when the character already starts enabled. It is useful when a previous puzzle or cutscene may have disabled speech motion. It does not generate timing or build a mouth mapping.

For an authored conversation, use **Start Dialogue** and select the dialogue. The line nodes then choose their speakers, portraits, text, and recordings.

Voice and lip-sync actions

Action	What to set or expect
Enable Lip Sync	Character and Enabled. Runtime on/off override for an already configured character.
Set Voice Speaker	The character whose mouth should follow direct voice/message playback. An empty selection clears that default.
Play Voice	Voice Audio or Voice Key, optional text, and Wait for Voice.
Pause Voice	Pause the active voice; its mouth timing pauses with it.
Resume Voice	Continue the active voice and timing.
Seek Voice	Move to the chosen position in seconds.
Stop Voice	Stop the active voice session.
Say / Character Say	An explicit speaker, text, and optional voice.
Player Say	Speech from the current playable character.

Normal Say actions identify their own speaker. You do not need a separate Set Voice Speaker before every normal line. Use the default-speaker action for direct Play Voice or a voiced general message.

If **Wait for Voice** is on, the list waits for playback to finish before executing the following step. Put it off when later steps must pause, seek, or overlap the current recording. An immediate Pause Voice after a waiting Play Voice has nothing left to pause.

Disabling lip sync restores the resting appearance. It does not mute or delete the voice recording. Keep audio muting, text display, and mouth motion as deliberate separate choices.

Visual scripts

Use the same shared actions as visual nodes. Connect them in execution order and fill in the same character/audio fields. A mouth-related node does not need to replace the rest of your interaction graph.

For example: interaction start, Enable Lip Sync, Character Say, then the puzzle's variable or inventory actions. Keep the voice and its line together so a future edit does not accidentally play one person's audio with another person's portrait.

GDScript example

Put this inside the entry function supplied by the script editor. Keep that script template's wrapper and parameters. Replace the sample IDs with records in your game.

```
await SetLipSyncEnabled({
    "characterId": "elias", "enabled": true
})
await Say({
    "speakerId": "elias",
    "portraitId": "model_face",
    "text": "Every afternoon, that bell calls the town home.",
    "voiceAssetId": "voice_elias_01",
    "waitForContinue": true
})
```

The explicit `portraitId` requests that portrait for this line. For ordinary action-list editing, choose a particular portrait on a Dialogue Designer line when you need that selection exposed in the editor.

PCA Script, BASIC, and Lua

These examples show direct voice playback. They assume the character mapping and audio track are already prepared. They use the same action/property names; the dictionary syntax and waiting notation differ.

PCA Script

```
function run()
    SetVoiceSpeaker({"characterId": "elias"})
    PlayVoice({"voiceAssetId": "voice_elias_01", "wait": true})
    SetVoiceSpeaker({"characterId": ""})
end
```

BASIC

```
FUNCTION Run()
    AWAIT SetVoiceSpeaker({"characterId": "elias"})
    AWAIT PlayVoice({"voiceAssetId": "voice_elias_01", "wait": true})
    AWAIT SetVoiceSpeaker({"characterId": ""})
END FUNCTION
```

Lua

```
local Script = {}
function Script.run()
    await(SetVoiceSpeaker({characterId = "elias"}))
    await(PlayVoice({voiceAssetId = "voice_elias_01", wait = true}))
    await(SetVoiceSpeaker({characterId = ""}))
end
return Script
```

For pause/seek controls in an existing GDScript sequence, the calls are `await PauseVoice({})`, `await SeekVoice({"seconds": 0.5})`, and `await ResumeVoice({})`. Use them while a non-waiting voice is still active, not after the complete line has already finished.

Movement, states, and control

The shared character actions also cover walking, following, facing, room changes, and player control. Configure states in Character Designer first, then use the relevant actions from the list/node picker.

Choose a real character and a reachable destination for **Character Walk To**. Use **Follow Character** and **Stop Follow Character** for a companion. Keep room transfers consistent with the single reusable-character placement rule. For a gesture, choose an animation/state that actually exists on the selected character.

Do not run a second lip-sync clock in a custom loop merely because a body animation is also playing. The built-in speech session already connects the recording, scene face, and portrait.

24. Test the complete character

Test in small passes. It is much easier to identify a missing portrait before you add a conversation with twenty choices and three camera changes.

A sensible test order

1. Start a fresh Test Scene and confirm the intended player receives input.
2. Walk, stop, turn, and run if running is part of the game.
3. Check the NPC's position, collision clearance, and autonomous activity.
4. Use Talk and confirm the correct character answers.
5. Check the name, text, and chosen portrait.
6. Listen to the recording and check that it is the exact take assigned to the line.
7. Watch the mouth at the beginning, a few consonants/vowels, a pause, and the end.
8. Repeat with Portrait Preview and the scene face if both are targets.
9. Test the next line with a different speaker and a different portrait.
10. Test a saved/exported game as well as the editor preview.

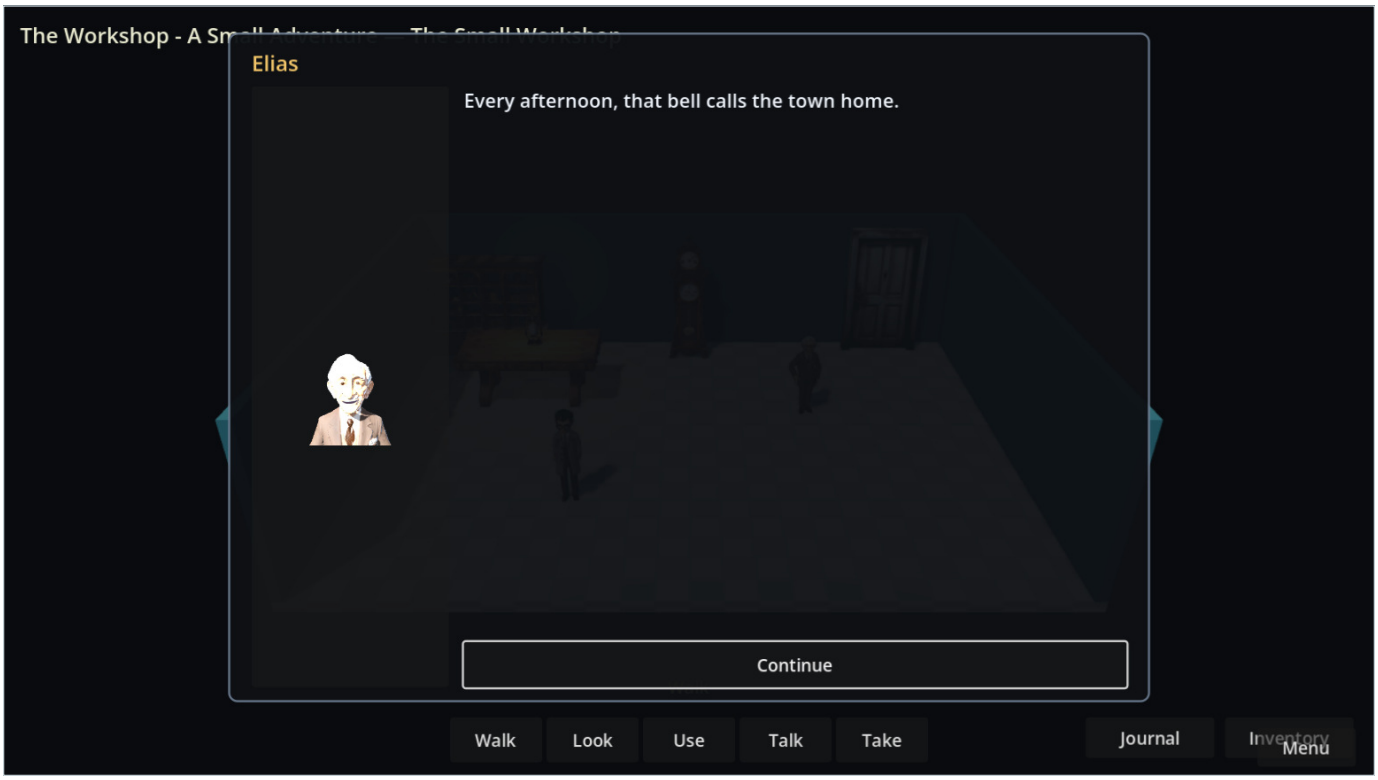


Figure 41. In the running test, a saved track drives the scene model and a still-image portrait. The voice clock is paused at 1.3 seconds to inspect the shared cue.

The pictured test uses two different mappings at once: the scene mesh's named blend shapes and the portrait's closed/open pictures. That is the purpose of the separate target tabs. Both follow the same recording rather than running independent speaking animations.

Check the ordinary rest appearance after the line. The portrait's ongoing animation should resume where applicable, and a mouth layer should return to its mapped rest. Test disabling and re-enabling lip sync if your game uses that feature during speech.

Quick diagnosis

What you see	Check first
A polygon or placeholder instead of the model	Asset preview, Model / Asset assignment, character link, visible scale, and applied changes.
Model appears but does not walk	Default/state IDs, source model, exact clip names, loop settings, and movement/navigation.
Correct clip but the character slides away	Root motion, actor movement speed, model fitting, and facing alignment.
Player stops too far from someone	Physical collision, blocking, and approach distance. Event volume is a separate control.
A portrait never appears	Speaker, default/requested portrait, source asset, dialogue style, and custom GUI picture binding.
Wrong portrait appears	Speaker's portrait list, requested ID, default selection, and any personal dialogue GUI.
Recording plays but no mouth moves	Enable Lip Sync, target selection, applied mapping, timing mode, and the asset's saved track.
Only the scene face moves	Targets must include Portrait; check its separate mapping and GUI/source type.
Only the portrait moves	Targets must include Scene Character; check the actual scene mesh or sprite mapping.
Basic opening works but phonemes do not	Basic mode is loudness-only; use a prepared automatic/manual track for cue distinctions.
Mouth changes but looks late or wrong	Correct recording/take, cue boundaries, language, transition time, and mapping weights.
Mouth twists or the whole head moves	Wrong jaw bone, axis, or mesh node. Inspect the imported rig.
Entire portrait jumps while talking	Whole-frame images differ in size/pose/alignment. Use aligned art or a mouth layer.

What you see	Check first
Mouth overlay drifts during animation	Source-image anchors, flips, per-frame positions, and fitting.
Mouth stays open in silence	Rest cue/mapping, cue gaps, track timing, and the source model's neutral pose.
Imported timing looks valid but is wrong	It belongs to another recording, language, or take. Regenerate or retime it.
Generate cannot run	Decodable audio, length, recording language, desktop recognizer availability/path, and its reported error.
Settings seem to revert	Apply the correct section, save the project, and check whether a placed instance has an override.

Separate a data problem from a presentation problem

Seek the recording in Preview Lip Sync. If the expected cue is selected but the face stays still, inspect the mapping and asset. If the face responds correctly there but fails in the game, check the selected speaker, portrait, GUI binding, and runtime enable/disable actions.

If timing is good on one face and poor on another while both share the voice clock, compare their mappings and smoothing/artwork. Do not shift the entire recording's track to compensate for one badly aligned mouth layer.

Save, validate, and export

Apply character, state, portrait, and lip-sync changes in their own editors. Save each reviewed audio track, then save the project. Run **Validate** and resolve missing references before making a release.

Test the exported game with the model/image assets and the correct recordings included. The saved lip-sync timing travels with the Audio asset. The runtime does not need to recognize speech again.

25. Character Designer reference

Use this index when you know what you want to change but cannot remember where it lives.

Job	Where to look in this guide
Understand assets, definitions, and placements	Chapters 1–4.
Add, clone, delete, or rename a character	Chapter 2.
Assign a model and solve placeholder geometry	Chapter 3.
Fit, face, place, and move between rooms	Chapter 4; Scene Designer for the placed instance.
Idle, walk, run, talk, and gestures	Chapter 5, Animation States.
Sprite sheets and image animation	Chapter 6; 2D / 2.5D in Chapter 11.
Display name, speech color, role, enable	Chapter 7, Character Settings.
Walk/run/turn speed and footsteps	Chapter 7; Character > Movement Audio.
Collision size, perspective scaling, following	Chapters 7–8.
Wander, patrol, and social activity	Chapter 8, NPC Behavior.
Starting player and character switching	Chapter 9, Player Settings and Project Settings.
Personal GUIs and initial inventory	Chapter 9.
Events, verbs, and inventory-item responses	Chapter 10.
Custom Properties and advanced record data	Chapter 11.
Still portrait, default, and expressions	Chapter 12.
Animated/video/Live2D portraits and portrait audio	Chapter 13.
3D portrait source, clip, and framing	Chapter 14.
Speaker, line, voice, and typewriter text	Chapter 15.

Job	Where to look in this guide
Portraits in authored dialogue GUIs	Chapter 16.
Enable, timing, targets, and fallback	Chapter 17.
Scene/portrait blend-shape mapping	Chapter 18.
Whole frames, mouth layers, and cue letters	Chapter 19.
Jaw, material textures, and Live2D parameters	Chapter 20.
Generate, inspect, edit, import/export, save	Chapter 21.
Batch recordings, replacement takes, translations	Chapter 22.
Action lists, visual scripts, and script examples	Chapter 23.
Final tests and fault-finding	Chapter 24.

Keep this short checklist beside a new character

1. Artwork imported and visible in Assets.
2. Reusable character created with the right ID and model/image source.
3. Idle and walk states mapped and tested.
4. Scene placement, feet, facing, collision, and approach checked.
5. Player/NPC role and behavior set deliberately.
6. Default portrait applied and visible during speech.
7. Spoken text, speaker, and exact voice recording assigned together.
8. Lip sync enabled with the intended targets.
9. Scene and portrait mouth mappings applied.
10. That recording's timing generated/imported, reviewed, and saved.
11. Rest, speaker changes, and exported playback checked.
12. Project saved and validation reviewed.

Once these pieces work for one line, add the rest of the conversation. Reuse the character's mapping, prepare timing for each new recording, and give every important change a quick test before building more on top of it.