Documentation

Animora: Powerful Timeline-Based UI Animation Tool for Unity

Animora is a robust and easy-to-use timeline animation tool, built especially for UI animations in Unity — but it's not limited to UI. With a deep focus on performance, clarity, and extensibility, Animora helps developers craft complex animations using clips and actions instead of traditional keyframing.

What sets Animora apart:

- 🔧 Code-driven, tween-based animation.
- ⚡ Super lightweight and performance-focused.
- 🧱 Fully extendable — create your own custom clips and actions.
- 🔁 Minimal runtime overhead — animations run only while active.

📁 Folder Structure

```
pgsql
KopierenBearbeiten
Assets/
├── Plugins/
│   ├── OM/                  → Root folder for all Oumari.dev assets
│   │   └── Shared/           → Common systems: Easing, Events, Ticker, T
imer, StateMachine, Search, etc.
│   ├── TimelineCreator/        → Core functionality of the timeline editor
│   └── Animora/
│       ├── Demos/            → Demo scenes: Demo 1, Demo 2, etc.
│       ├── Editor/           → Timeline UI, drawers, inspectors
│       ├── Runtime/          → Runtime components like `AnimoraPlayer`, `
AnimoraClip`
│       └── Moduls/           → All Clips and Actions grouped (UI, Camera,
Transform, etc.)
```

🚀 Getting Started

1. Import the asset into your Unity project.
2. Open one of the demo scenes and explore how Animora is wired.
3. Drag the `AnimoraPlayer` into your scene and start adding clips!

🧠 Core Concepts

`AnimoraPlayer`

Main controller for playing animations.

- Add it to your scene.
- Configure timeline settings.
- Add and arrange clips.

`AnimoraClip`

Base class for all timeline clips. Examples:

- `AnimoraClipPosition`
- `AnimoraClipRotation`

Override methods like `OnStartPlaying`, `OnEvaluate`, and `OnCompleteLoop` to implement custom behavior.

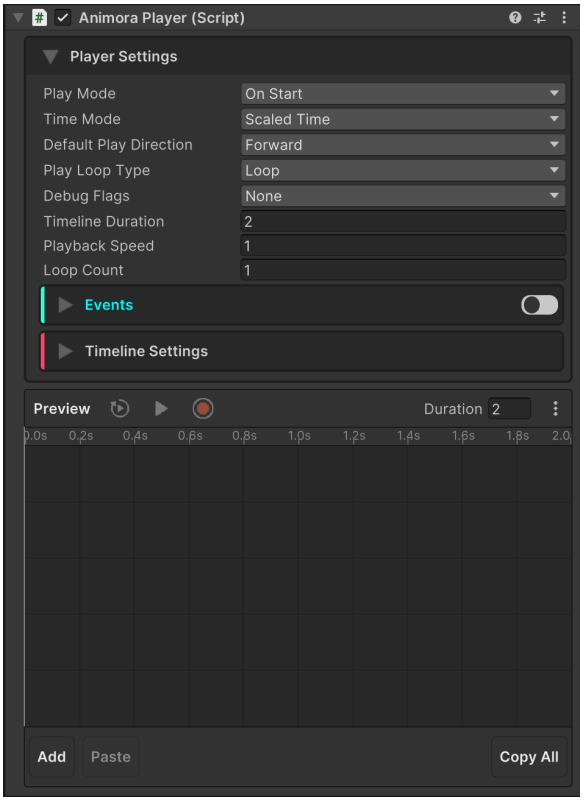
`AnimoraAction`

Attachable logic to clips. Can be triggered at key moments:

- Start Playing
- Loop Enter/Exit
- Complete
- Etc.

Built-in actions include setting active states, UI properties, transforms, etc.

🎮 `AnimoraPlayer` Settings

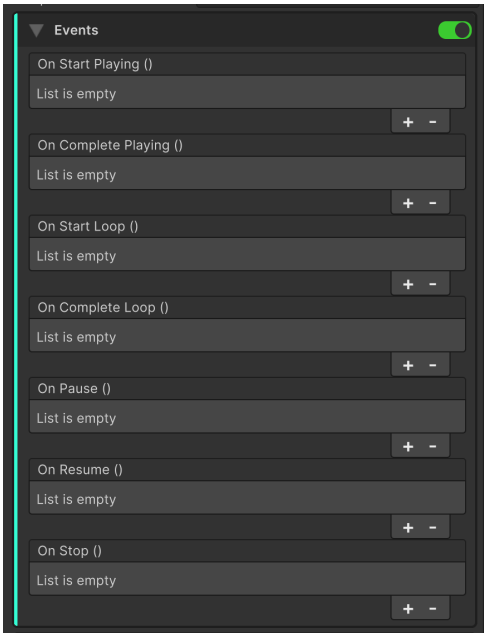


Setting	Description
Play Mode	When the animation plays: `OnStart`, `OnEnable`, or `Manual`
Time Mode	`ScaledTime` (affected by Time.timeScale) or `UnscaledTime` (useful during pause)
Default Direction	Forward or backward
Loop Type	`Loop`, `PingPong`, or `Once`
Timeline Duration	Total duration of the timeline
Playback Speed	Speed multiplier
Loop Count	-1 = infinite loop

🎛️ **Debug Mode**: Enable logs for specific events (Start, Complete, Enter, Exit, etc.)

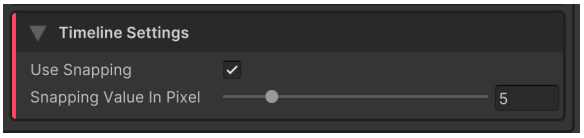
🔁 Events System

Trigger Unity Events at specific animation points:



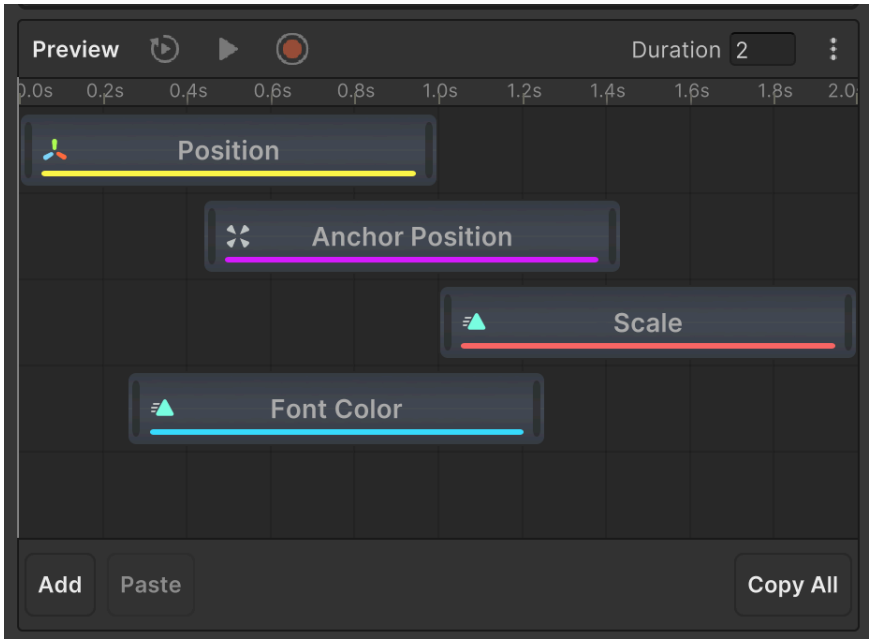
- `onStartPlaying`
- `onCompletePlaying`
- `onStartLoop`
- `onCompleteLoop`
- `onPause`, `onResume`, `onStop`

⚙️ Timeline Editor Settings



- **Use Snapping**: Toggle clip snapping.
- **Snapping Value**: Pixel distance for snapping.

🧭 Timeline UI Breakdown



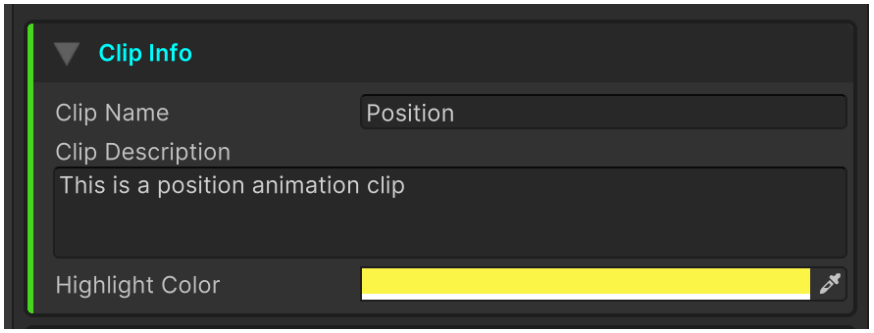
- **Preview Toggle**: Enable preview mode.
- **Replay**: Restart animation.
- **Play/Pause/Stop**: Playback controls.
- **Add Clip**: Opens search to add clips.
- **Copy/Paste/Duplicate/Remove Clip**: Standard editing operations.

✍️ Select a clip to open its **Inspector Panel** below the timeline.

🎨 Clip Inspector

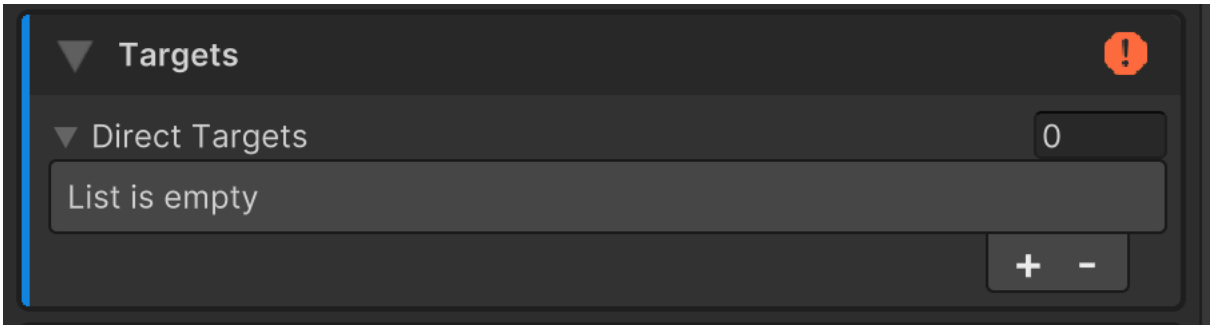
When you select a clip:

- **Clip Info**: Name, color, and description.

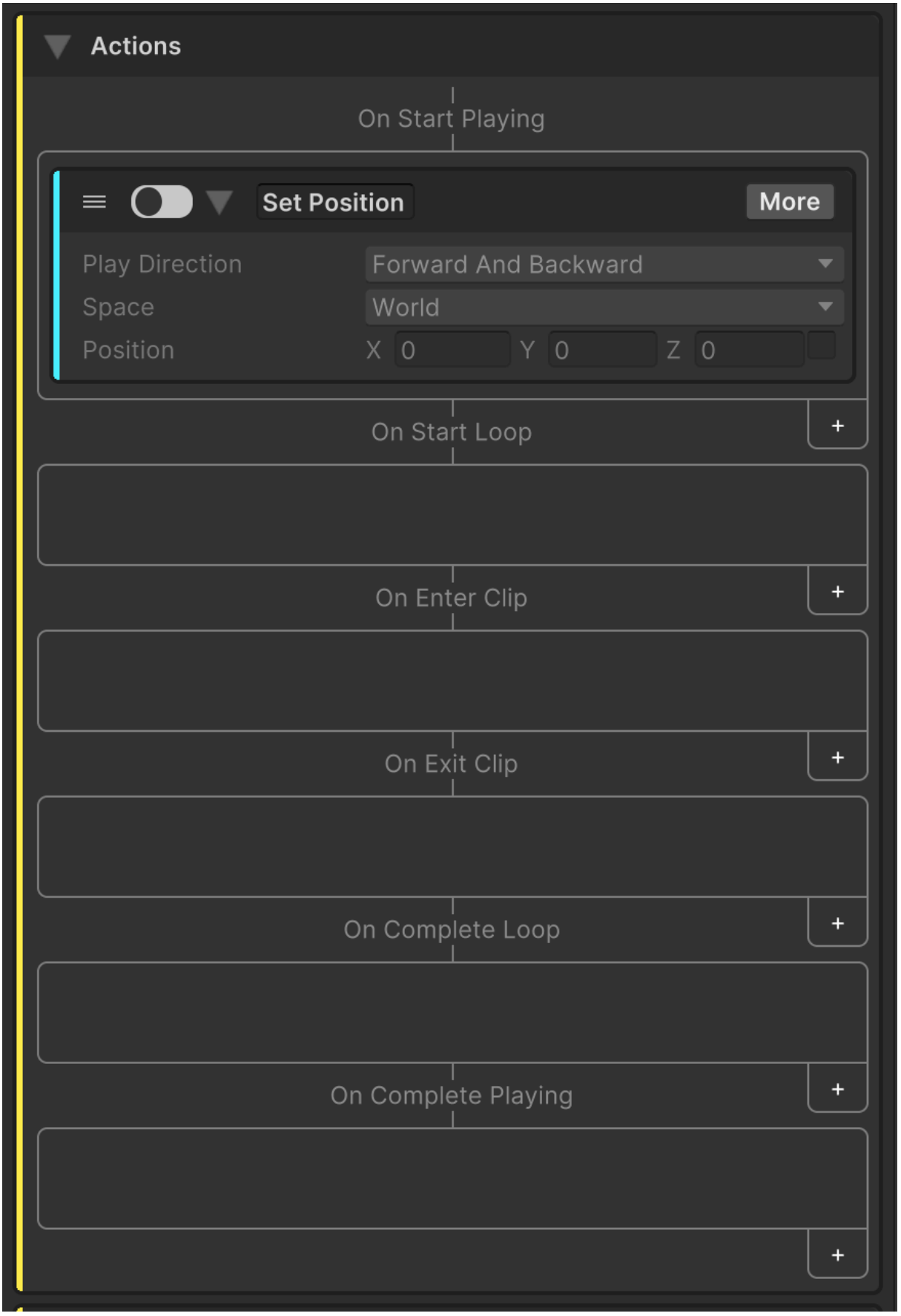


- **Time Settings**: Start time, duration, play chance, debug flags.

- **Targets**: The objects this clip will animate (e.g., `RectTransform`, `TextMeshPro`, etc.)



⚡ Actions Manager



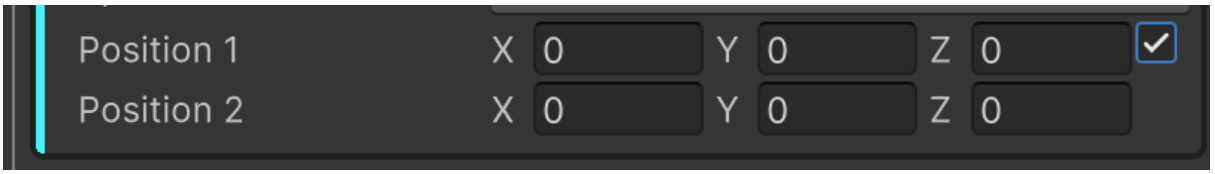
Each clip can execute **Actions** at key moments:

Trigger	When It Fires
On Start Playing	Once at animation start
On Start Loop	Every loop start
On Enter	When clip becomes active
On Exit	When clip ends
On Complete Loop	At loop completion
On Complete Playing	When entire animation ends

Each Action contains:

- **Handle**: Reorder the action
- **Switcher**: Toggle active state
- **Play Direction**: Forward/Backward/Both
- **Custom Targets**: Optional override of clip targets
- **Context Menu**: Edit or delete

AnimoraValue<T>

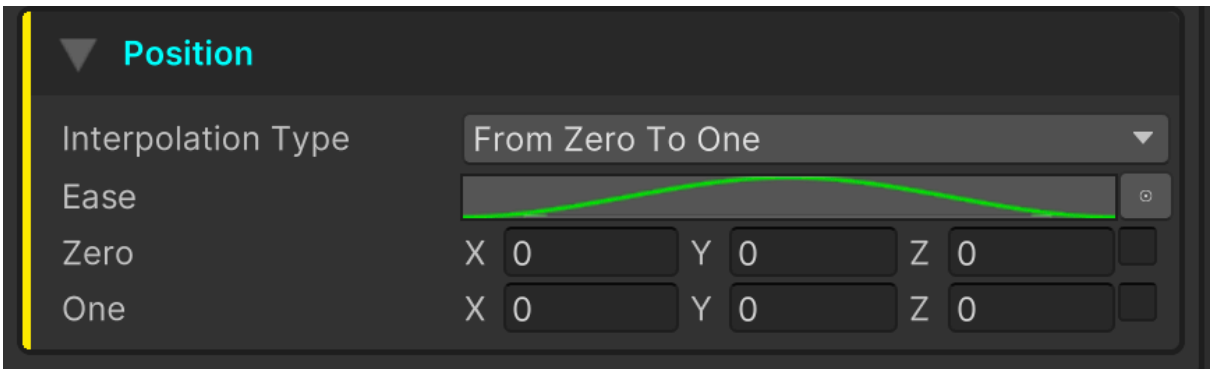


Used across Animora to define values with optional randomness.

- Can be a single fixed value.
- Or a random value between two points.

- Supported types: `float`, `Vector2`, `Vector3`, `Color`, etc.

🧮 AnimoraInterpolation<T>



A powerful system to animate values over time.

Types:

- `FromZeroToOne`
- `FromCurrentToOne`
- `FromCurrentByOne`

Ease: Built-in easing curves or custom animation curves.

Use with any value type supported by `AnimoraValue<T>`.

🎬 List of Built-in Clips & Actions

📷 Camera

- `Camera Color`, `Camera FOV`, `Orthographic Size`
- Actions: `Set Camera Color`, `Set Camera FOV`, etc.

🎨 Canvas Group

- `Canvas Alpha`
- `Set CanvasGroup Alpha`

🖼️ Graphic (Image)

- `Color Image`, `Fill Amount`

- Actions: `Set Color`, `Set Fill Amount`

🔧 Transform

- `Position`, `Rotation`, `Scale`
- Actions: `Set Position`, `Set Rotation`, `Set Scale`

🔤 TextMeshPro

- `Font Color`, `Font Size`, `Spacing`
- Actions: `Set Character/Word/Line Spacing`, `Set Font Size`

🧩 UI (RectTransform)

- `Anchor`, `Pivot`, `Offset`, `Slide To Screen`
- Actions: `Set Anchor`, `Set Offset`, `Set Size Delta`

🔁 Core

- `Nested Player` (run sub-timelines)
- Actions: `Set Active`, `Set Duration`, `Set Start Time`

📦 How to Extend

Animora is fully extensible:

- Derive from `AnimoraClip` to create new animation types.
- Implement `AnimoraAction` for new logic during playback.
- Add new types under `Moduls/` and they'll be picked up by the editor.

Thank you for purchasing **Animora**!

If you have any questions, feedback, or suggestions, feel free to reach out:

📧 **Email**: mooumari2@gmail.com

Your feedback helps make Animora better for everyone. Happy animating!