

D.A.

Button

manual for developers
1.0.0

Introduction

- 1 **D.A. Button** is a button with support for **multiple target graphics** and the ability to create **animation sequences** for each separate target graphic.

With **D.A. Button**, you can:

1. Set up **color**, **scale**, and **position** animations for each individual target graphic;
2. Create **sequences** of animations that will be executed on **click**, **hover** and when **disabling interaction**;
3. Create **looped** sequences of animations.
4. Enable or disable event processing for each target graphic (for example, one of the target graphics can ignore a tap on the component, but if you tap on another target graphic, the animation will trigger simultaneously for both target graphics).

- 2 If you encounter any errors while working with the asset, please write me about it at provided contacts. I typically respond quickly to messages, offer assistance on an individual basis, and address any identified errors in the upcoming updates.

Discord Server: <https://discord.gg/rEGvwQV3hr>

Telegram Group: <https://t.me/dabutton>

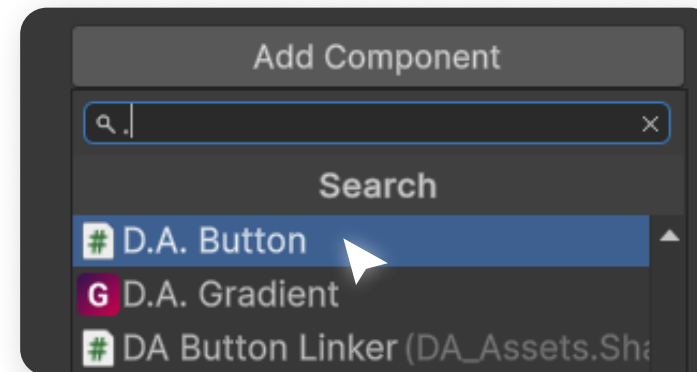
Telegram Support: https://t.me/da_assets

Email Support: da.assets.publisher@gmail.com

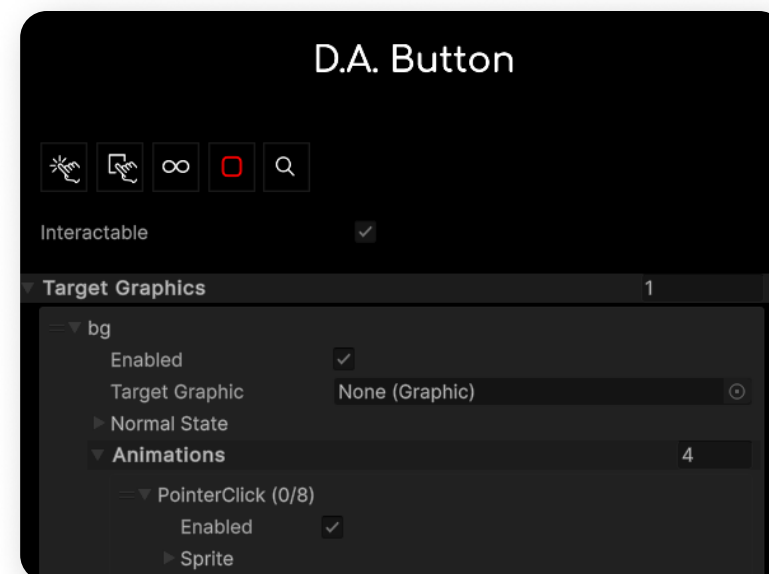
Website: <https://da-assets.github.io/site/>

Getting started

- 1 Import the asset using the Unity Package Manager.
- 2 Create an empty GameObject on the scene and add the "D.A. Button" script to it.



- 3 You will see the asset's interface with default animations.



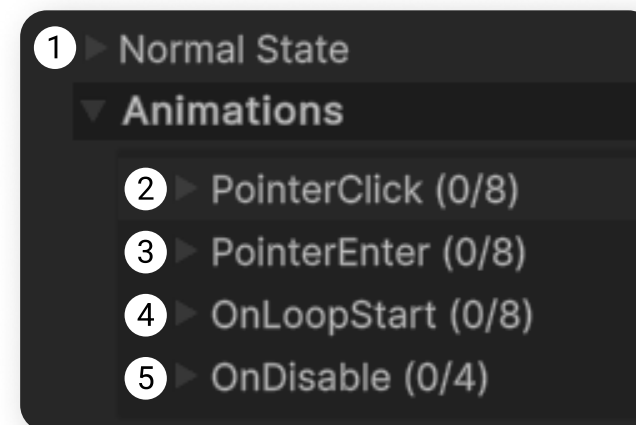
Asset UI

- 1 You can use the functions for previewing the animation that work in the Editor - you don't need to run the play mode for this.



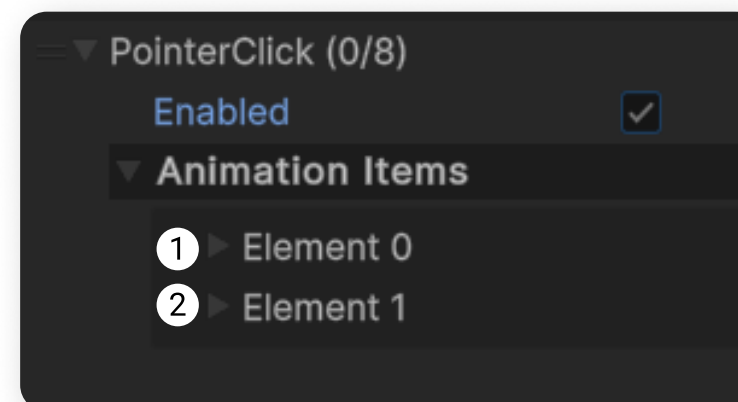
1. Simulate a click (if you have a click animation);
2. Simulate hovering (if you have a hover animation);
3. Simulate a looped animation (if you have a looped animation);
4. Stop the looped animation;
5. Search for custom animation templates;
6. Use to enable or disable the ability to select a UI element (same as in UI.Button).

- 2 Types of animations.



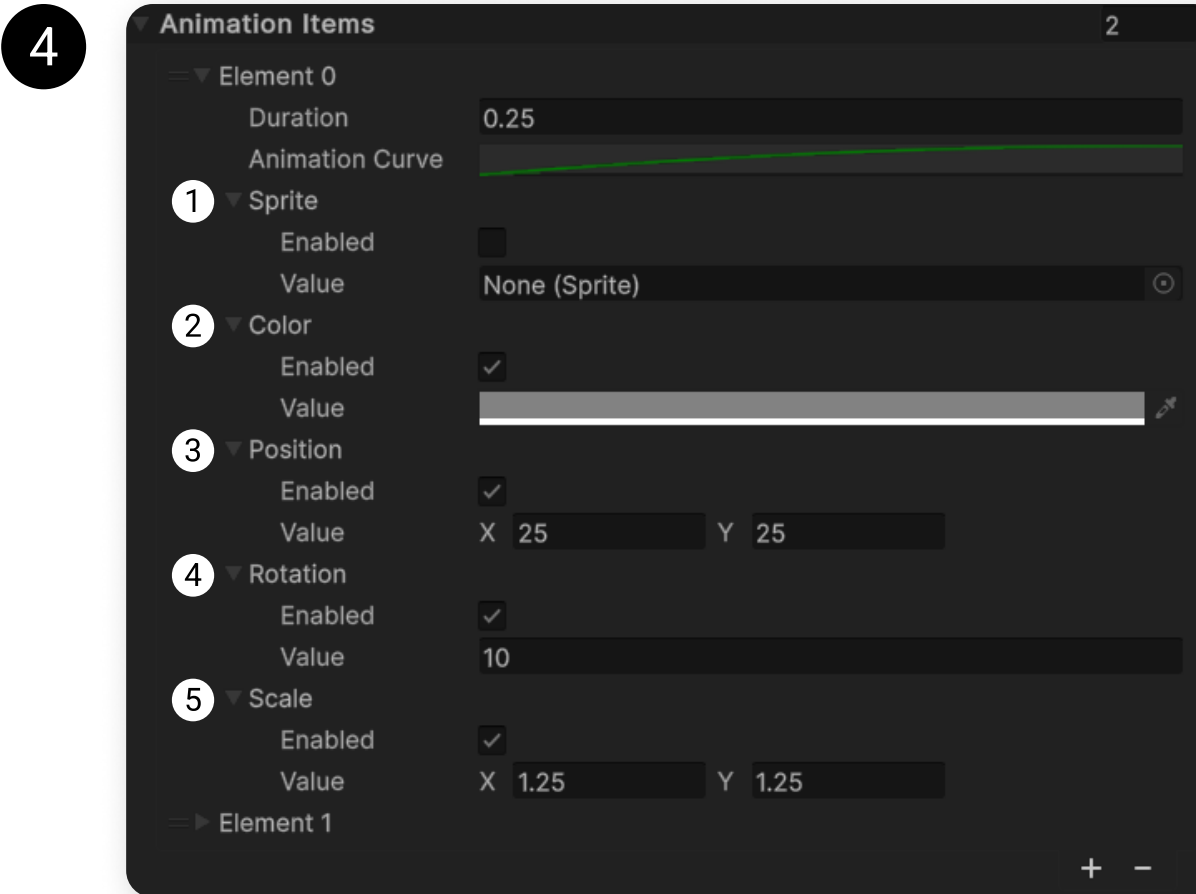
1. The normal state of TargetGraphic. **Mandatory for filling.** After the completion of any of the animations, the TargetGraphic returns to NormalState in the time specified in the parameters of NormalState.
2. Played when you click on your TargetGraphic;
3. Played when hovering over your TargetGraphic. After you move the cursor away from the TargetGraphic, it will return to the NormalState;
4. If this is enabled, when Playmode is launched, your TargetGraphic will be animated in a loop. The looped animation does not interfere with the execution of the Click and PointerEnter animations;
5. Played when you set **Interactable = false**.

- 3 Animation items.



- 1, 2 - in this array, there can be any number of animations. Animations are played sequentially upon the execution of the **event** within which the array is located (for example, upon a Click). In the case of the **OnLoopStart** event, upon completion of the last element in the array, the cycle of animations starts from the beginning.

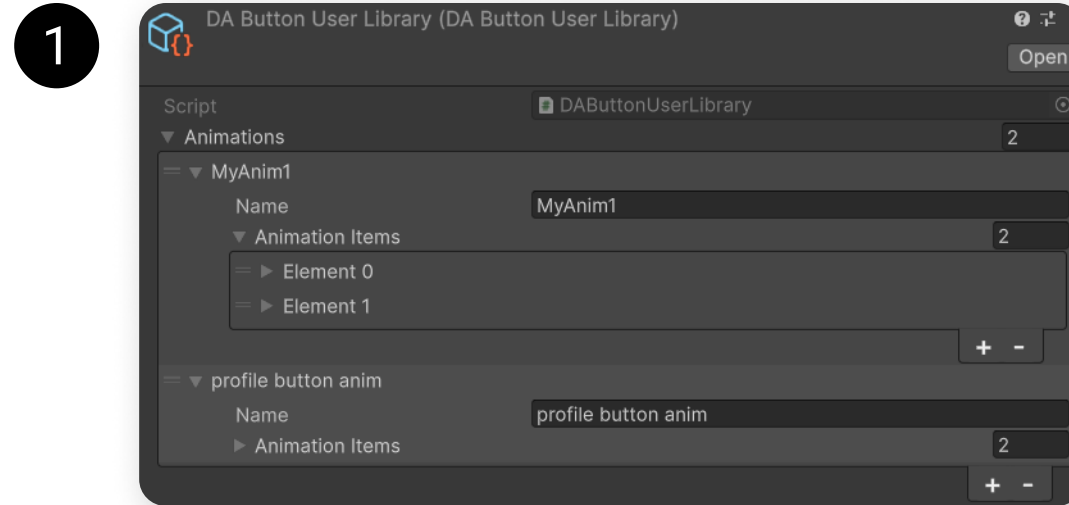
Asset UI



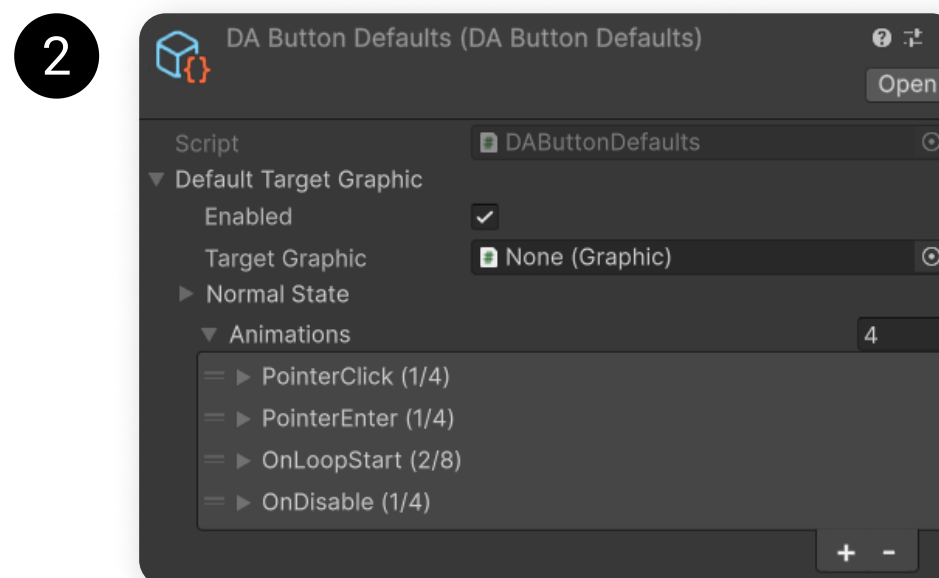
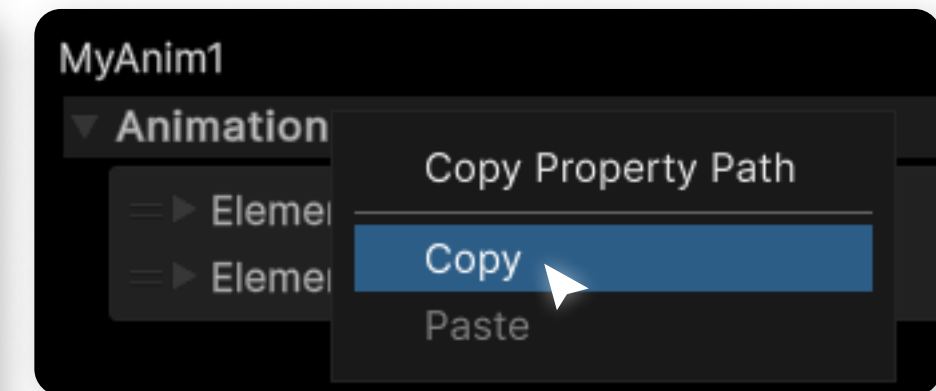
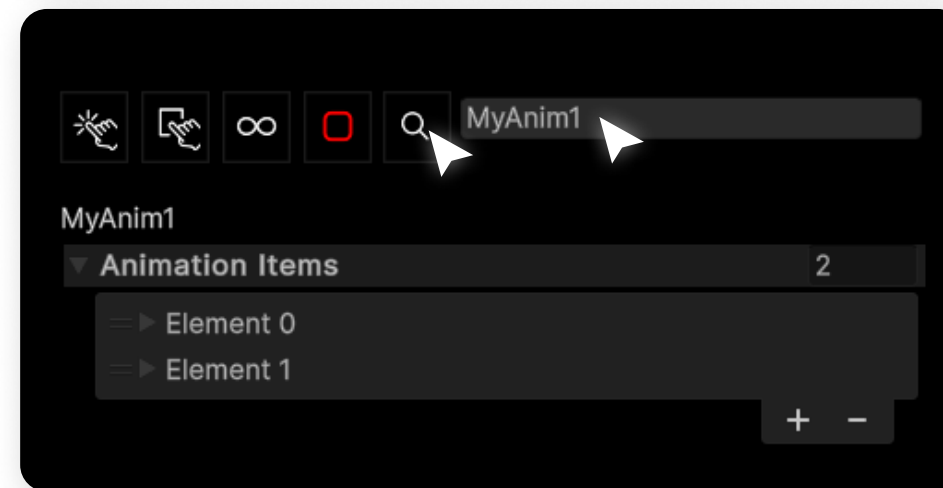
Available animation options. Executed if the Enabled flag is checked.

1. Change of sprite (instantly, without transition);
2. Animation of color through overlay;
3. Animation of local position;
4. Animation of local rotation;
5. Animation of scale.

User Library and Default values



You can add your own animation sequences to the "DAButtonUserLibrary.asset" file, so that you can later access them quickly through the asset interface by pressing the search button.



The default animations that are added to the button when the script is added in the inspector are located in the "DAButtonDefaults.asset" file. You can change them.