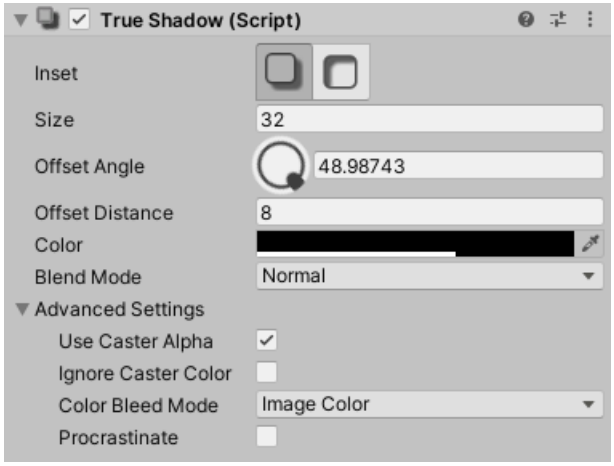
Table of Contents

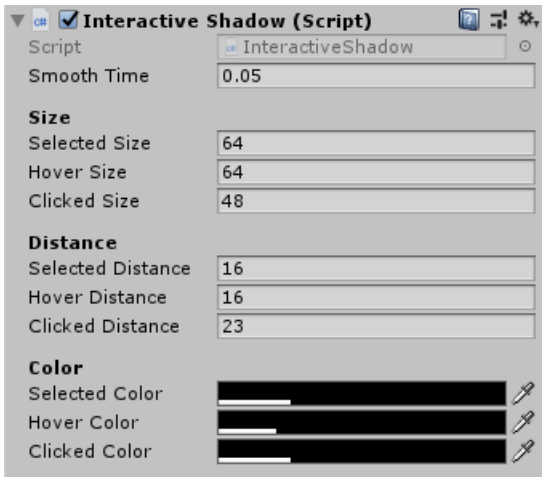
Getting Started

1. Add `True Shadow` to your UI element.



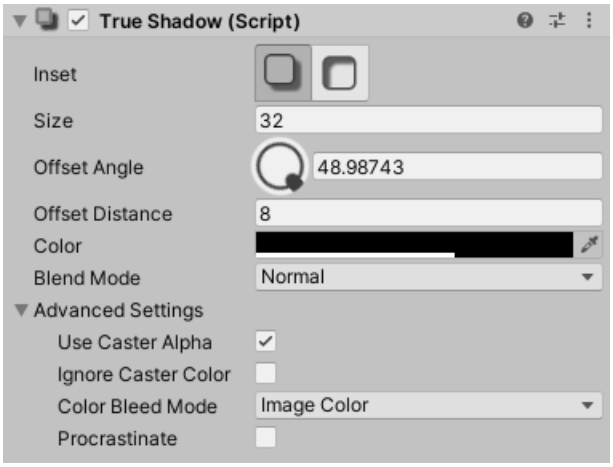
2. Tune it to your liking.

3. Optionally add `Interactive Shadow` to modify shadow properties based on user interaction.



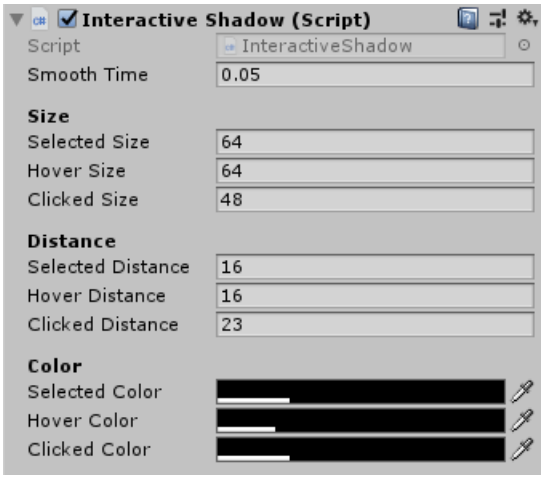
Customize

True Shadow



- **Inset**: Choose between inner and outer shadow
- **Size**: Size of the shadow
- **Offset Angle**: Direction to offset the shadow toward
- **Offset Distance**: How far to offset the shadow
- **Color**: Tint the shadow
- **BlendMode**: Blend mode of the shadow
- **Use Caster Alpha**: Whether or not the alpha channel of the Graphic affect the shadow
- **Ignore Caster Color**: When on, the color of the shadow will be what specified in the Color property. When off, the shadow color will be based on the color of the shadow caster Graphic.
- **Color Bleed Mode**: How to obtain the color of the area outside of the source image. Automatically set based on Blend Mode. You should only change this setting if you are using some very custom UI that require it

Interactive Shadow



The `Interactive Shadow` component allow you to quickly create shadows that can react to user interaction, such as by mouse or game controller.

It 3 supported states: hovered, selected and clicked work the same as the builtin `Selectable`, such as `Button`.

Scripting Considerations

True Shadow display shadows by creating hidden GameObject positioned right after the caster UI element in the hierarchy. This is so that the shadow is rendered right above the UI element.

If you are using script that interact with the hierarchy of the shadow casting UIs in some way, you should keep these hidden object in mind. For example, calling `transform.childCount` on the parent will also count the shadow object.

If you spawn shadow casting UIs at runtime, you must not assume the amount of children will be equal to the amount you instantiated. If you instantiate UIs in a loop, you must keep these hidden objects in mind when calculating the instantiated UIs `siblingIndex`.

Generally, it is not a good idea to use `siblingIndex`/`transform.GetChild()` to reference objects. Better alternative are to assign the elements in the inspector, or to use more specific API such as `FindObjectOfType`, `GetComponentInChildren`.