SIMPLE ZOOM

Contents Page

Getting Started

Overview:

"Simple Zoom" allows for developers to easily implement ScrollView zooming functionality to their desktop and/or mobile applications through the use of the scroll-wheel and/or pinch-to-zoom gestures.

Features:

- ✓ Easy to use and highly customizable.
- ✓ Mouse (desktop) and touch (mobile) support.
- ✓ Specify restrictions for the minimum and maximum zoom.
- ✓ Zoom on the pointer (mouse/finger) or pre-defined custom position on the screen.
- ✓ Use clamped/elastic zoom types when zooming.
- ✓ Define whether to transform the scale or size (width/height) should when zooming.
- ✓ Assign different zoom controls (buttons, sliders, zoom views etc.).
- ✓ Determine whether movement should be restricted when zooming.
- ✓ Allow users to double-tap to fully zoom in/out depending on the current zoom.
- ✓ Adjust the button/scroll-wheel increment and smoothing.

Included:

- ✓ Two example projects:
  - ➢ Photo Viewer
  - ➢ World Map
- ✓ Ready-to-use prefabs:
  - ➢ Zoom
  - ➢ Zoom View
- ✓ In-depth offline documentation

Contact:

In the event you are unable to find the information you are looking for or have found a bug, please feel free to send me an email (daniel@daniellochner.com) and I will get back to you as soon as possible.

Installation:

Once you have downloaded "Simple Zoom" from the Unity Asset Store, and the "Import Asset", window appears in Unity, verify that all items to import are selected and then click the "Import" button in the bottom right of the window.

Keep in mind that the "Simple Zoom" folder does not have to be in the root directory of your project, so feel free to move it anywhere!

Quick Start:

To add a "Simple Zoom" component to your Game Object, go to "GameObject > UI > Simple Zoom" or simply search for the "SimpleZoom.cs" script in the "Add Component" menu in the inspector.

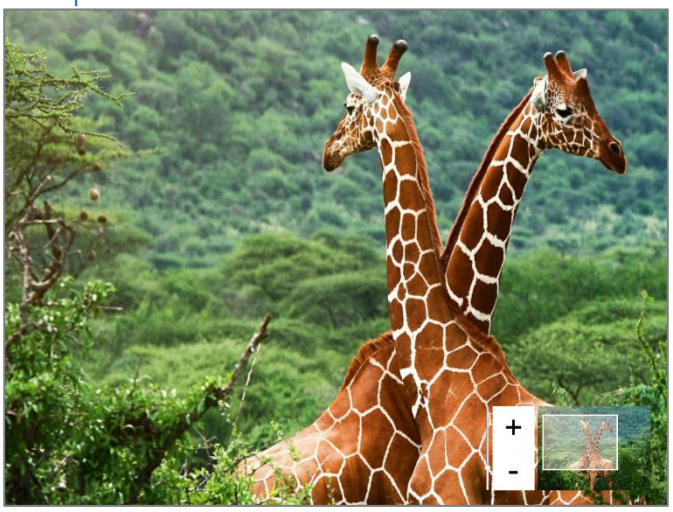
If you are unsure of how to setup your Game Object to use the "Simple Zoom" component, there are two basic prefabs that can help get you started (Zoom and Zoom View).

To add any of these prefabs to your scene, simply drag it from the "/Simple Zoom/Prefabs" folder and place it under your Canvas, or navigate to "GameObject > UI > Simple Zoom", and select it from there.

Examples:

There are two example projects included that illustrate how the "Simple Zoom" component can be used in real scenarios:

Example 1: Photo Viewer



This example shows how a basic Photo Viewer application can be created using the different zoom controls. An elastic zooming type is applicable when using a mobile device and is a common feature when viewing images.

Example 2: World Map

This example shows how a basic World Map can be created. The scroll wheel increment and smoothing values are altered due to the size of the map.

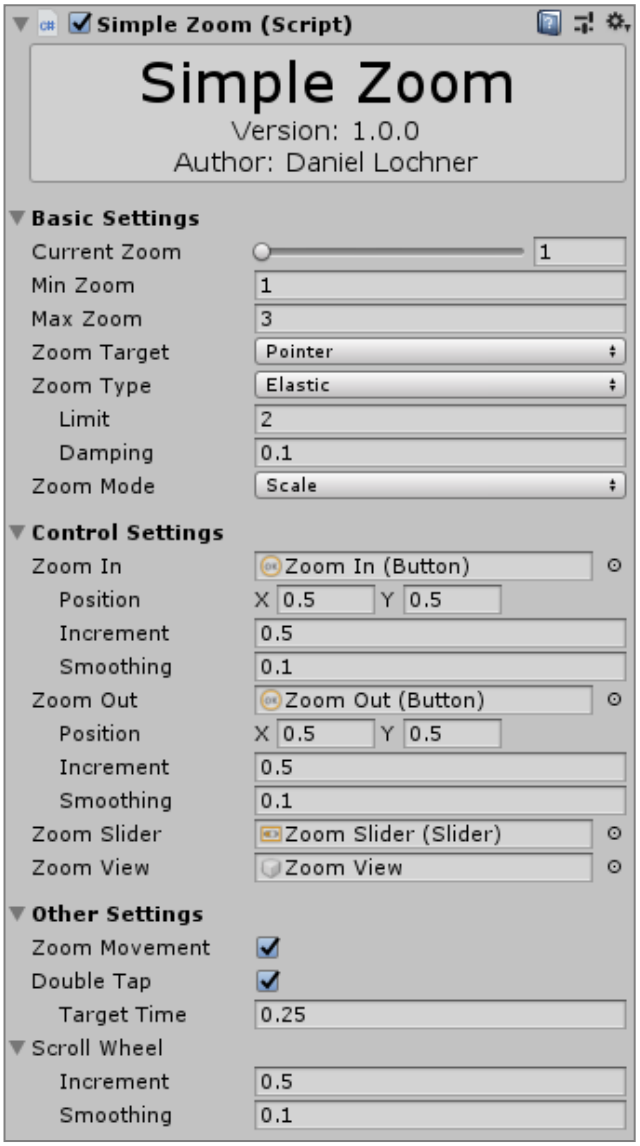
“Simple Zoom” component

Inspector Breakdown:

The “Simple Zoom” component’s inspector is broken down into three sections that can each be toggled to fold-in or out, to keep your inspector more manageable. Each section, and the configuration of each of their sub-items, are discussed in detail later.

“Simple Zoom”

1. Basic Settings:
    a. Current Zoom
    b. Min Zoom
    c. Max Zoom
    d. Zoom Target
    e. Zoom Type
        i. Damping
        ii. Limit
    f. Zoom Mode
2. Control Settings:
    a. Zoom In
        i. Position
        ii. Increment
        iii. Smoothing
    b. Zoom Out
        i. Position
        ii. Increment
        iii. Smoothing
    c. Zoom Slider
    d. Zoom View
3. Other Settings:
    a. Zoom Movement
    b. Double Tap
        i. Target Time
    c. Scroll Wheel
        i. Increment
        ii. Smoothing



Configuration:

By hovering over a property’s label in the editor, a tooltip will be displayed, providing a basic description of what that property is. For a more in depth guide as to how to configure each property, the following written guide on “Configuration” has been provided:

Basic Settings:

1. Current Zoom:
   The current value by which the scale/size is multiplied by when zooming.

2. Min Zoom:
   The minimum value by which the scale/size could be multiplied by when zooming.
3. Max Zoom:
   The maximum value by which the scale/size could be multiplied by when zooming.
4. Zoom Target:
   Determines what value the content's pivot will be set to when zooming. The two options being the pointer (which represents the mouse/finger) or custom (which is the custom value the pivot will be set to when zooming).
5. Zoom Type:
   Determines whether zooming may exceed the minimum/maximum zoom values and how it should be handled. If a mobile device is used, this may be set to “Elastic”, in which case, the minimum and maximum zoom values may be exceeded (within a specified limit) and a dampening effect may be applied to it.
6. Zoom Mode:
   Determines whether zooming transforms the scale or size (width/height) when zooming.

Control Settings:

1. Zoom In:
   The button whose On Click event listener will be assigned the "Zoom In" method invocation on setup with values specified below:
   1. Position: The custom value the pivot will be set to when zooming in.
   2. Increment: The amount by which the current zoom will be transformed by
   3. Smoothing: The smoothing by which the current zoom will be transformed into the incremented zoom when zooming in.
2. Zoom Out:
   The button whose On Click event listener will be assigned the "Zoom Out" method invocation on setup with values specified below:
   1. Position: The custom value the pivot will be set to when zooming out.
   2. Increment: The amount by which the current zoom will be transformed by.
   3. Smoothing: The smoothing by which the current zoom will be transformed into the incremented zoom when zooming out.
3. Zoom Slider:
   A slider that displays the progress of the zoom represented by a number between 0 (minimum) and 1 (maximum).
4. Zoom View:
   A small scale representation of the zoom.

Other Settings:

1. Zoom Movement:
   Should the ScrollRect's movement be restricted when zooming?
2. Double Tap:
   Should users be able to double-tap (the amount of time users have in order for a double-tap to register refers to the target time) to fully zoom in/out
3. Scroll Wheel:
   Adjust the scroll-wheel’s increment and smoothing values when zooming.

Scripting References:

Controlling a "Simple Zoom" from outside requires that you invoke public methods and/or get/set the values of properties defined. Each class that allows this level of control is discussed below, with each publically accessible property/method listed and described.

"SimpleZoom.cs"

1. Namespace: DanielLochner.Assets.SimpleZoom
2. Extends: MonoBehaviour
3. Implements: IPointerClickHandler
4. Properties:

Property	Description
CurrentZoom	The current value by which the scale/size is multiplied by when zooming.
TargetZoom	The target value that the current zoom targets based on smoothing.
ZoomProgress	The progress of the zoom represented by a number between 0 (minimum) and 1 (maximum).
ZoomMargin	The margins representing the number of pixels between the viewport and content edges, used to display the zoom view.
ScrollRect	The ScrollRect (could be the derived MultiTouchScrollRect class) associated with the SimpleZoom.
Content	The Content to be zoomed.
Viewport	The Viewport associated with the SimpleZoom.

5. Methods:

Method	Description
ZoomIn(Vector2 pivot, float increment, float smoothing)	A method that zooms in (based on the defined increment and smoothing values) when invoked.
ZoomOut(Vector2 pivot, float increment, float smoothing)	A method that zooms out (based on the defined increment and smoothing values) when invoked.

"MultiTouchScrollRect.cs"

1. Namespace: DanielLochner.Assets.SimpleZoom
2. Extends: ScrollRect
3. Implements: none
4. Properties:

Property	Description
MultiTouchPosition	The average position between all touches on the screen.

Support

Frequently Asked Questions (FAQ):

1. Where are all the Frequently Asked Questions?
   Once enough questions have been frequently asked, I will update the documentation and add them here ;)

Contact Information:

In the event you are unable to find the information you are looking for or have found a bug, please feel free to send me an email (daniel@daniellochner.com) and I will get back to you as soon as possible.

Please do not leave a review, regarding an issue with the asset, on the Unity Asset Store unless you have first consulted me asking for help.