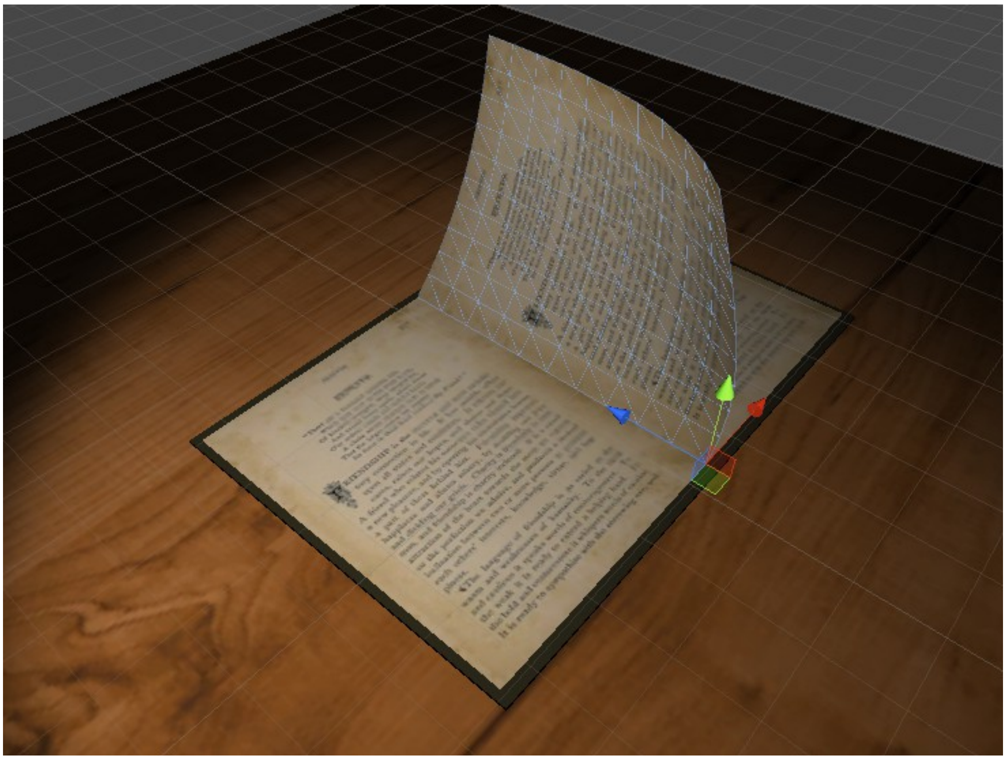
Page Curler



Getting Started

First, drag the **PageCurl** prefab from the root folder of the package into the Scene view. The prefab is already set up with everything you need to get started:

- A plane mesh (you can change its scale to better suit your needs).
- The main **PageCurl** component that controls the curling effect. You can play with its inspector fields to see the effect in real time.
- An **Animation** component with two pre-made page turning animations (one from right to left, and the other from left to right).
- Two materials using the included **PageCurl shader**, one for each side of the plane mesh. You can replace the textures in these materials with your own ones.
- A **Demo Controller** component, with some sample code showing how to control the page turning from a script.

For your own projects, you will want to remove the Demo Controller and replace it with your own control scripts that are more appropriate for your needs.

The effect itself is driven by the animations modifying the exposed inspector fields of the PageCurl component. You can create your own animations, modify the existing ones, or disable the Animation component entirely and set the values of the exposed fields directly from your own scripts.

Control from Scripts

The PageCurl component offers an easy interface to control the page turning effect from your own scripts:

- `float Flip(bool backwards = false)`

  Starts the default page turning animation. Use backwards = true if you want to turn the page in the opposite direction. Returns the length of the animation.

- `void Reset()`

  Set the page back to its original position.

- `Texture FrontTexture`

  Access the texture for the front side of the page. You can change the texture with a new one by assigning it to this variable.

- `Texture BackTexture`

  Access the texture for the back side of the page. You can change the texture with a new one by assigning it to this variable.

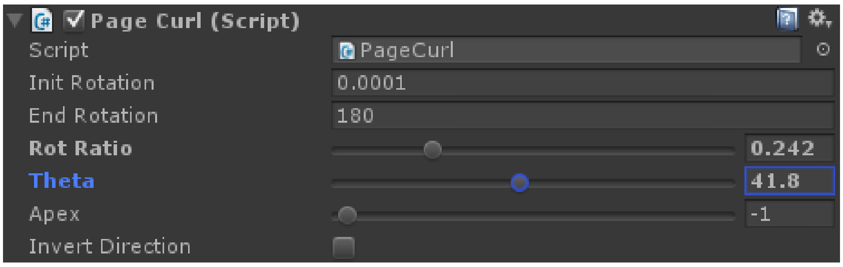
- `public void OnPageFlip()`

  You can declare this method on your own scripts and it will be called whenever a turning page animation ends. You can then start a new animation, change the textures, or run any of your custom game logic. (Your script needs to be on the same PageCurl object for it to be called).

Also, since the effect is driven by animations, you can simply access the **Animation** component and play, stop, or pause the animations directly yourself.

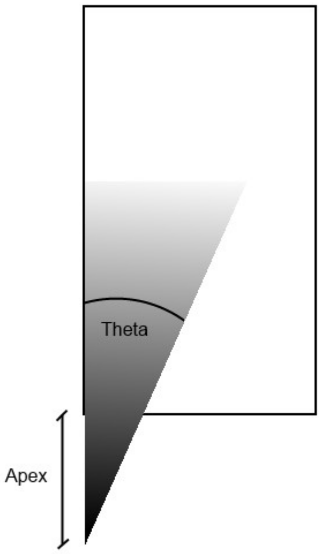
Inspector Fields

The effect works by projecting the page on to an imaginary cone and deforming it accordingly. The PageCurl exposed fields control the attributes of that cone. By animating those attributes at the same time that we rotate the plane around its edge, we can simulate the effect of turning a page.



☑ Page Curl (Script)

Script	PageCurl
Init Rotation	0.0001
End Rotation	180
Rot Ratio	0.242
Theta	41.8
Apex	-1
Invert Direction	☐



- **Init Rotation / End Rotation:** Control the starting and ending rotation of the plane in degrees. For instance, if you are using the effect on a book that is opened only by 100 degrees, you'd want to use 40 as the start rotation, and 140 as the ending one.

- **Rot Ratio:** This value controls how far along the rotation we currently are. It goes from 0 to 1. When 0, the page will be at its “Init Rotation”, and when 1, at its “End Rotation”.

- **Theta:** This is the angle of the imaginary cone. A lower value makes the curling effect more prominent.

- **Apex:** This controls the position of the imaginary cone along the page's edge. This value should always stay negative. The closer to zero, the more curved the page will be.

- **Invert Direction:** Use this to reverse the direction of the deformation (for instance, when making an animation that goes from left to right)

Creating a book

Most likely, you will want to have more than two pages of content to scroll through. When creating a book, it's recommended that you have a single PageCurl object, next to two static planes for the left and right pages (you can see this kind of setup in the included sample scene), and that you reuse the PageCurl object by swapping its front and back textures as you move through the content. You can swap the textures either before the animation starts (before the call to the **Flip** method), or right when it ends and you are notified through the **OnPageFlip** event.

You can also opt to have more than one PageCurl object, recreating the actual structure of a real book, and activate them sequentially. This is less performant, but it should work correctly as long as you don't have too many of them.