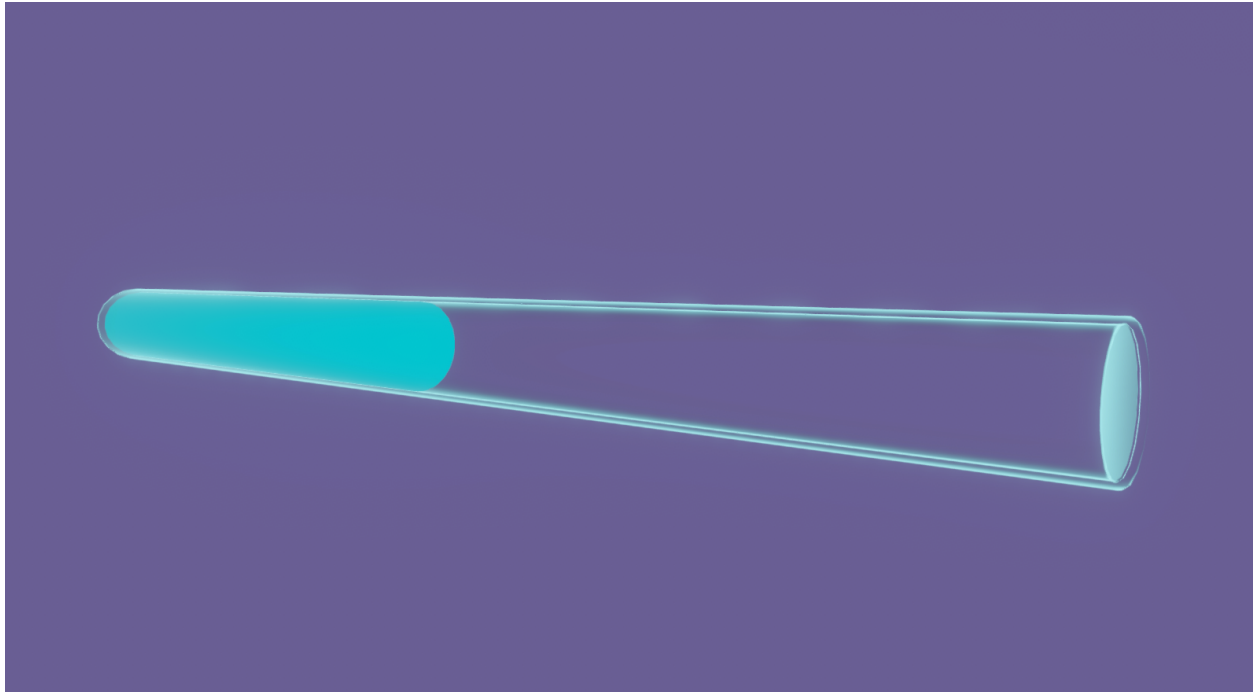# Modular 3d Progress Bars

Modularify

Straight Progress Bar

The straight loading bar aims to replicate a glass container which is filled based on a percentage. The color of the inner part interpolates linearly between two given values set in the inspector.
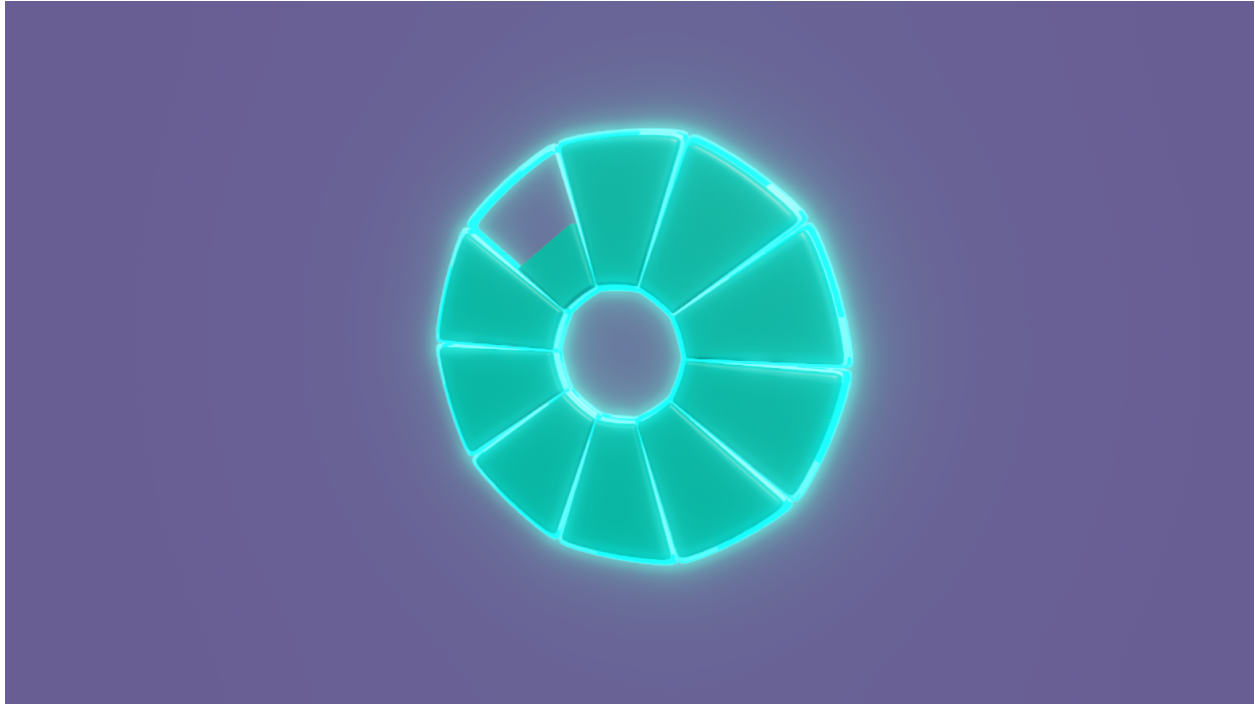
The parameters of the script under the *Loading Bar Settings and Colors* category are:

- **Percentage:** A float value that ranges from 0 to 1. Indicates how much of the progress bar has been filled.
- **Empty color:** HDR which represents the color of the inner part when the percentage is 0
- **Full color:** HDR which represents the color of the inner part when the percentage is 1
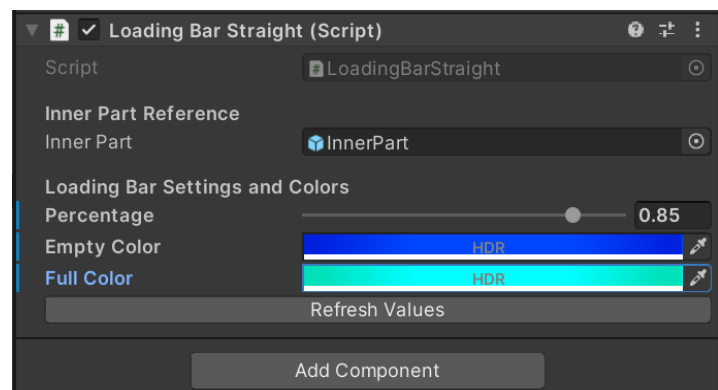
The **Refresh Values** button updates the colors and percentage when clicked.

Segments Progress Bar

The segments loading bar is divided into different parts which fill according to the percentage value passed as parameter or modified in the inspector.

The parameters of the script that can modified in the inspector under the Loading Bar Settings and Colors categories are:
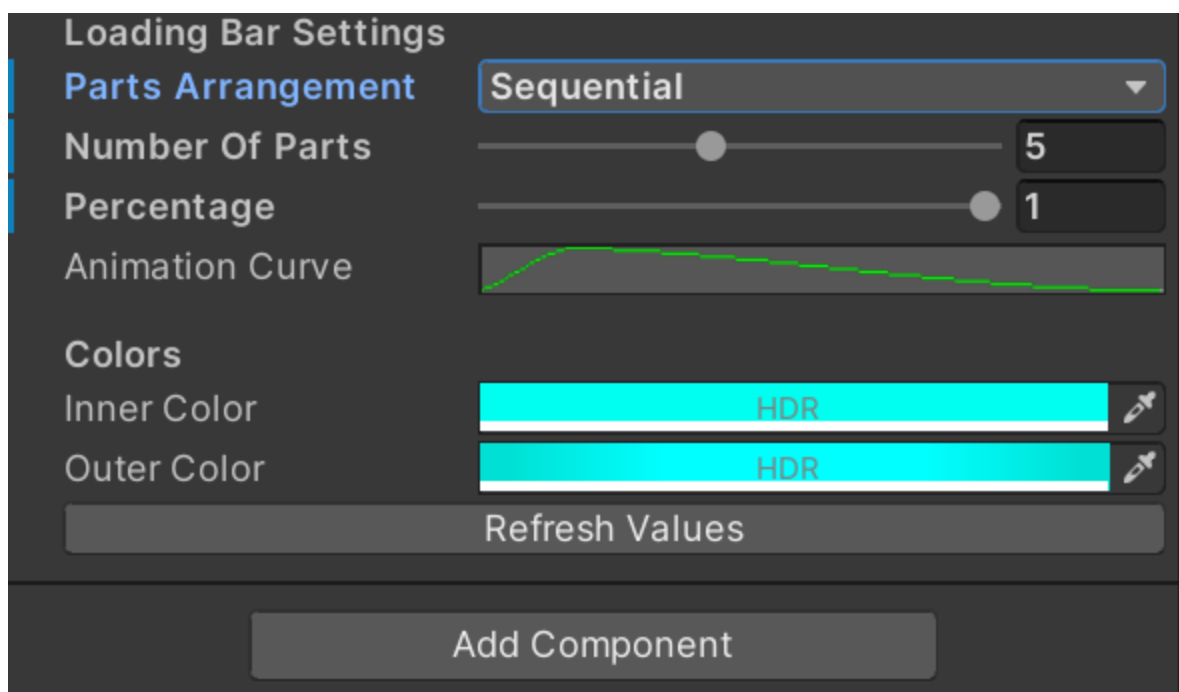
- **Parts arrangement:** Enum that can be set to be *Circle division* and *Sequential*
  - *Circle division*: when changing the number of parts they will always be arranged to complete a full circle
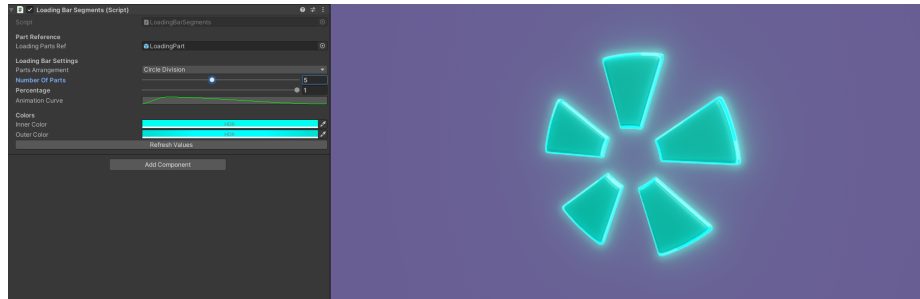
  - *Sequential*: when changing the number of parts they will always be one after another without leaving gaps; this however will not complete a full circle when the number of parts is less than 10.
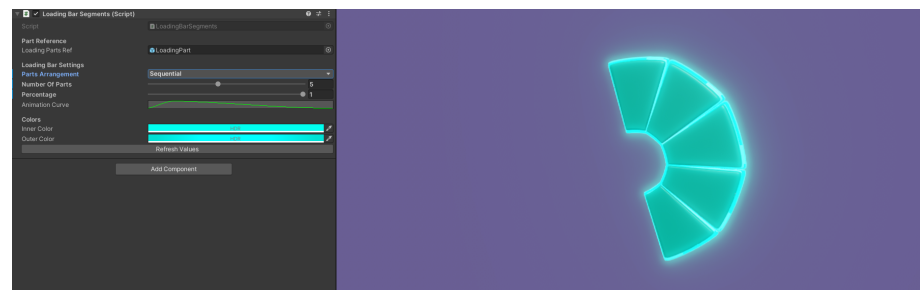

- **Number of parts:** Changes the number of segments of the progress bar. Its value ranges from 1 to 10 (included).
- **Percentage:** float value that ranges from 0 to 1 which indicates how much the progress bar is filled.
- **Animation curve:** Curve used to perform a small animation when each segment is filled.
- **Inner color:** HDR color used to fill the flat part of each segment
- **Outer color**: HDR color used to fill the edges of each segment
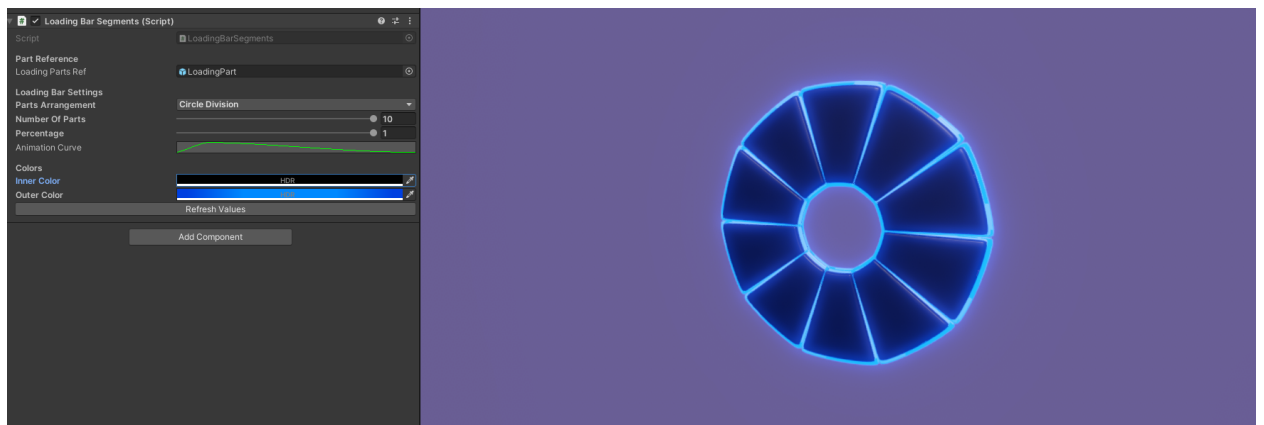
The **Refresh Values** button updates the previous values on the progress bar.

Straight Parts Progress Bar
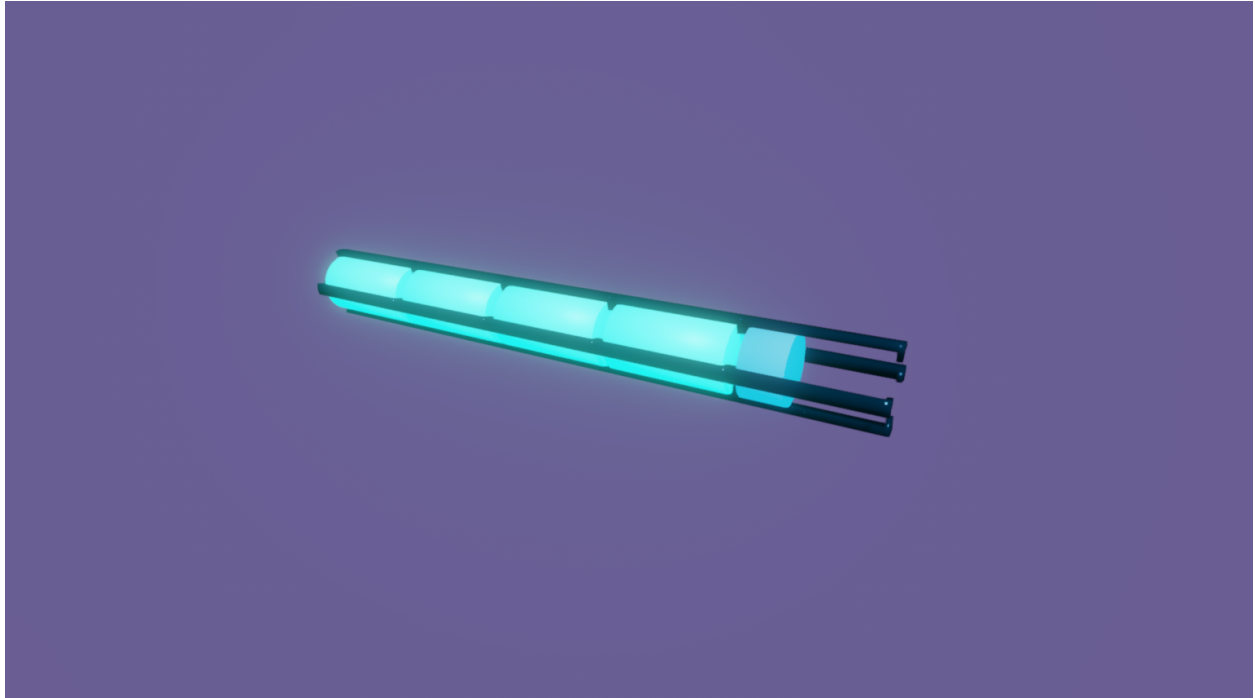
The straight parts progress bar aims to simulate many containers which can be filled based on a specific amount. The color of the inner parts interpolate linearly between two values. They can change color individually or all at the same time. Moreover this progress bar has an outer part which has a color of its own and can be highlighted based on the percentage of the progress bar.

The parameters of the script under the *Loading Bar Settings, Inner Colors* and *Outer Color* category are:
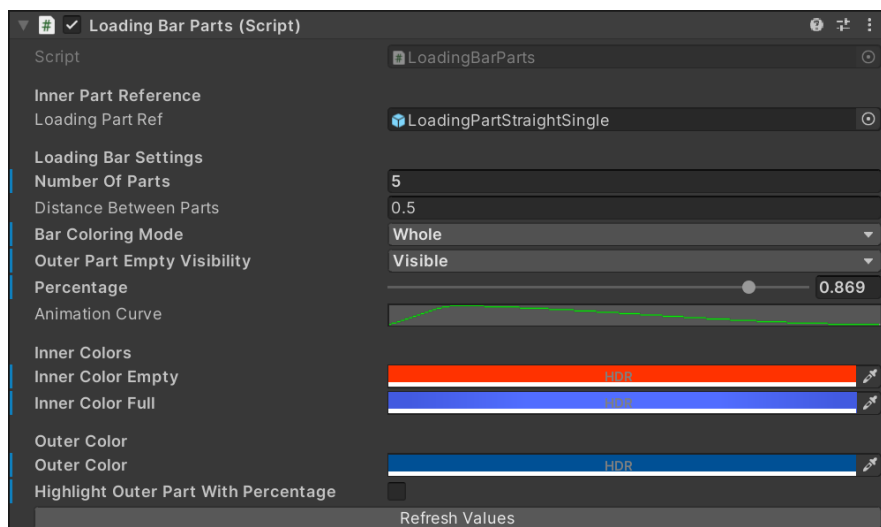
- **Number of parts:** The number of different parts the bar has. The minimum is 1.
- **Distance between parts:** float value which represents the distance of each individual part from the previous. The default value is 0.5 units.
- **Bar coloring mode:** Enum with values *Single* and *Whole*.
  - *Single:* Each part will linearly interpolate between the empty and full color
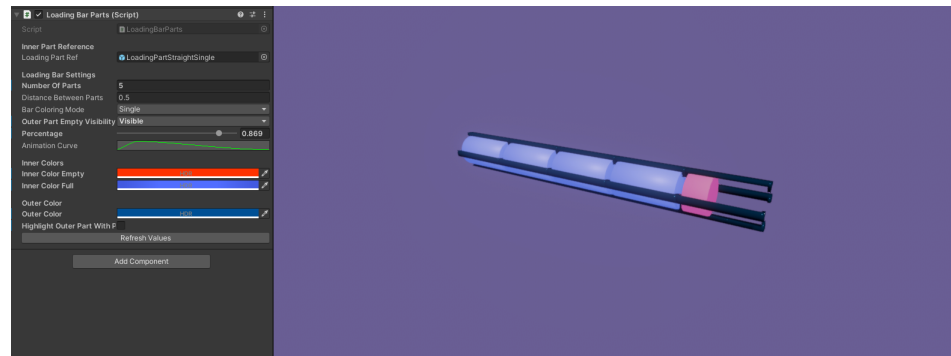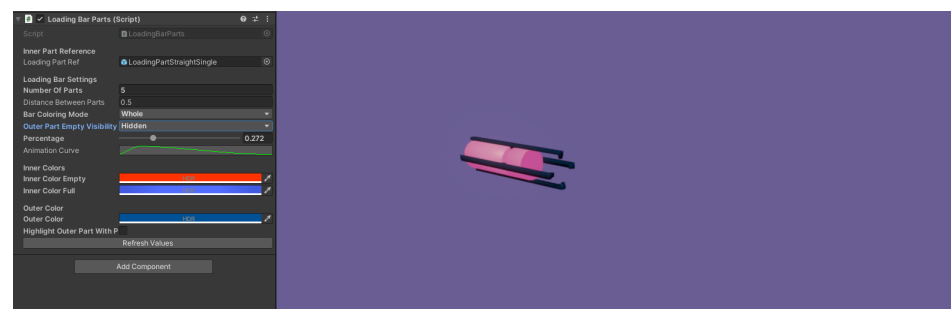

 - *Whole:* The color of each part will be the same for every segment.
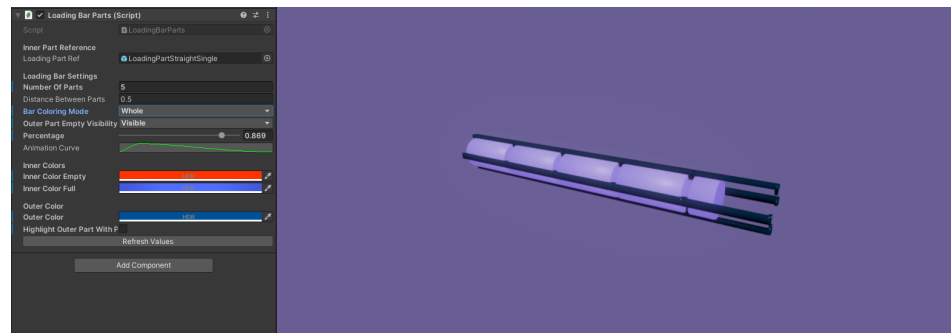
- **Outer part empty visibility:** Enum with values *Visible* and *Hidden*
 - *Visible:* The outer part is always visible when the percentage changes
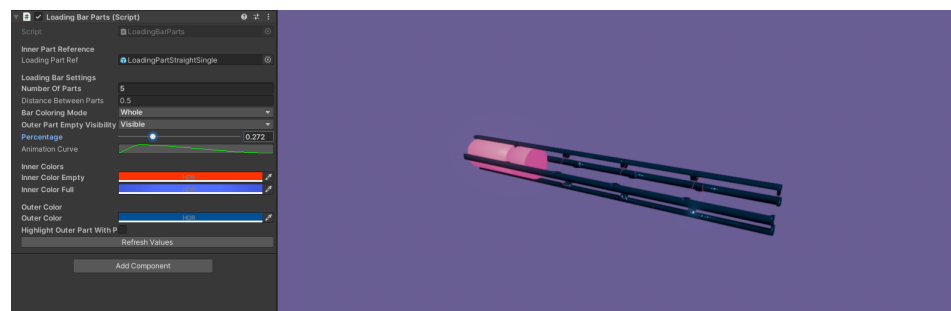
  - *Hidden:* The outer part will disappear if the current segment is not part of the total percentage

- **Percentage:** Float value that ranges from 0 to 1 which indicate the amount the progress bar is filled
- **Animation curve:** Curve used to perform a small animation when each segment is filled
- **Inner color empty:** HDR color of the inner part when the bar is empty
- **Inner color full:** HDR color of the inner part when the bar is full
- **Outer color:** HDR color of the outer part
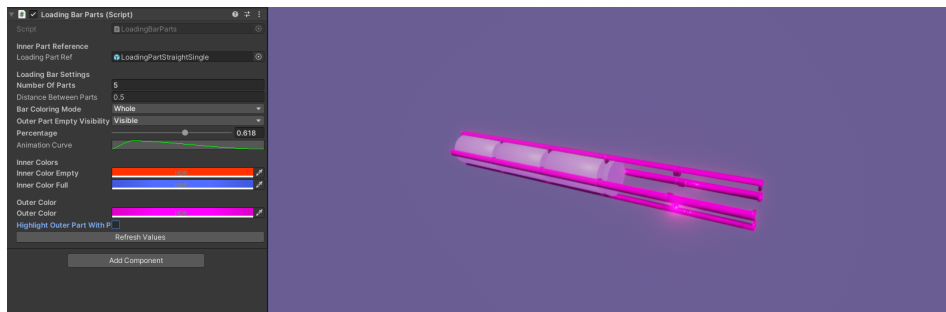
- **Highlight outer part with percentage:** Bool value that enables highlighting of the outer part, making it brighter, based on the current percentage
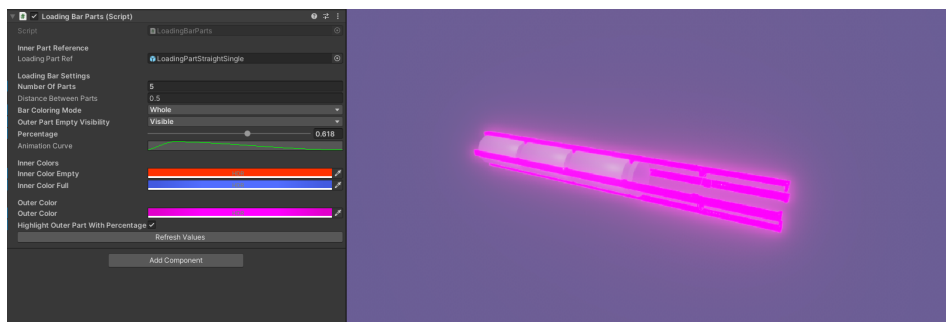
The **Refresh Values** button updates the previous values on the progress bar.

Setting the percentage

To set the percentage of the progress bars, create a script which holds a reference to the
LoadingBarStraight, *LoadingBarSegments* and *LoadingBarParts*. Either use the namespace in the
top of the class or specify the entire script name. Call the SetPercentage(float) of the object when
needed.

```csharp
using UnityEngine;
using Modularify.LoadingBars3D;
// Unity Script (1 asset reference) | 0 references
public class ExampleScript : MonoBehaviour
{
    [SerializeField]
    private LoadingBarStraight _loadingBarStraight;
    [SerializeField]
    private LoadingBarParts _loadingBarParts;
    [SerializeField]
    private LoadingBarSegments _loadingBarSegments;


    // Start is called before the first frame update
    // Unity Message | 0 references
    void Start()
    {
        _loadingBarStraight.SetPercentage(0.7f);
        _loadingBarSegments.SetPercentage(0.2f);
        _loadingBarParts.SetPercentage(0.5f);

    }


    // Update is called once per frame
    // Unity Message | 0 references
    void Update()
    {

    }
}
```