Radial Segmented Health Bars Guide and Reference

1 Contents

2 INTRODUCTION

Thank you for purchasing the Radial Segmented Health Bars (RSHB) Asset!

This document is a small guide to help you get started with the asset and contains some useful information for scripting.

This asset REQUIRES URP, so if you do not have URP installed, then please install it by going to Package Manager > Universal Rendering Pipeline and install and import the package into your project.

If you imported URP into a pre-existing project, some assets might not show correctly. Please refer to this video if that is the case: https://www.youtube.com/watch?v=ErsXwcb3n4c

If you encounter any bugs with this asset, please contact devorengegames@gmail.com with the details of the bug.

Check out my other assets if you like this one: https://assetstore.unity.com/publishers/49336

I also have a YouTube Channel: DevOrenge - YouTube

I hope you get everything you need out of this asset and out of this document!

3 FILES

The base Material can be found in the **Resources** folder of this asset. There you will also find a blank sprite. This sprite is used to ensure any SpriteRenderer you are trying to assign this asset to is able to be rendered correctly.

The **ShaderGraphs** folder contains the ShaderGraph.

The **Scripts** folder contains the scripts used to populate a given GameObject with the correct Material and which allow you to animate the properties of the RSHB Material.

The **Prefabs** folder is where you'll find finished GameObjects for UI (Image) or non-UI (SpriteRenderer) purposes which can simply be added to your scene.

4 USING THIS ASSET

This asset comes with an easy-to-use script which can be added to any GameObject which has a **SpriteRenderer** or an **Image** component attached to it.

Simply add the script **RadialSegmentedHealthBar** to your desired GameObject and you are good to go!

If something goes wrong, you'll get an error message in the console and the script will remove itself from the GameObject. Everything should work fine, provided you have a SpriteRenderer or an Image attached to your GameObject and you haven't removed any essential files from the asset.

The script has **six public properties** which can be used for animation or as an easy way to set the RSHB Material properties.

The shader also has **six public properties** which can be set like so:

```
material.SetFloat(Shader.PropertyToID("<ShaderProperty>"), <value>);
```

Following Notation:

<ScriptProperty> -> <**ShaderProperty**>

4.1 Data fields

SegmentCount -> **_SegmentCount**

> This is used to define how many segments the health bar has. This is a float value to allow for more flexibility, however, it is recommended to only use integers for this field to avoid funky looking health bars. Use a value of 1 to define a monolithic health bar with only one segment. Don't use 0…

RemovedSegments -> **_RemoveSegments**

> How many segments have been removed from the health bar's total segment count. This is also a float value, but unlike with the segment count field it is actually useful, here you can input something like 1.5 to remove 1.5 segments. It is also worth noting that if you have 1 set for the segment count this value acts like a percentage with 0 being 0% removed and 1 being 100% removed.

4.2 Appearance fields

Color -> **_Color**

> The color of the health bar.

Spacing -> **_SegmentSpacing**

> The amount of spacing between each segment.

Radius -> **_Radius**

> The relative size of the health bar. Values greater than 0.5 usually make the health bar exceed the bounds of the sprite.

Line Width -> **_LineWidth**

The thickness of the health bar. Setting a low value for the radius and cranking up the line width results in a pie looking health bar.

Rotation -> **_Rotation**

The rotation of the health bar.

5 Shader Reference

Following Notation:

<ScriptProperty> -> <**ShaderProperty**>

Color -> **_Color**

SegmentCount -> **_SegmentCount**

RemovedSegments -> **_RemoveSegments**

Spacing -> **_SegmentSpacing**

Radius -> **_Radius**

Line Width -> **_LineWidth**

Rotation -> **_Rotation**