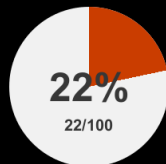
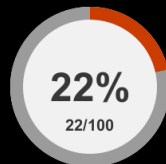
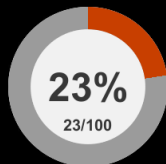


# ProgressCircle

A circular HUD tool asset pack

by Alper Yilmaz

[pm.alperyilmaz@gmail.com](mailto:pm.alperyilmaz@gmail.com)



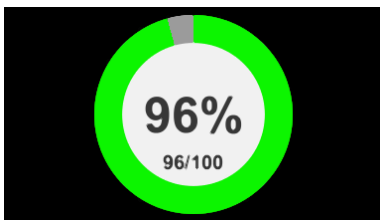
## What is ProgressCircle

ProgressCircle is a set of lightweight circular HUD tools that can be used for displaying progress, pools (HP, MP), XP for next level etc. All assets work by updating **CurrentValue** property. See “ProgressCircle/DemoScenes/DemoSceneScript.cs” script for example usage.

Currently included assets are:

- ProgressCircleThick
- ProgressCircleThin
- ProgressCircleFull
- HPCircleLinear
- HPCircleThick
- GrowingCircle

### ProgressCircleThick



ProgressCircleThick is a circular progress bar that can change color as its value changes. It has the following properties:

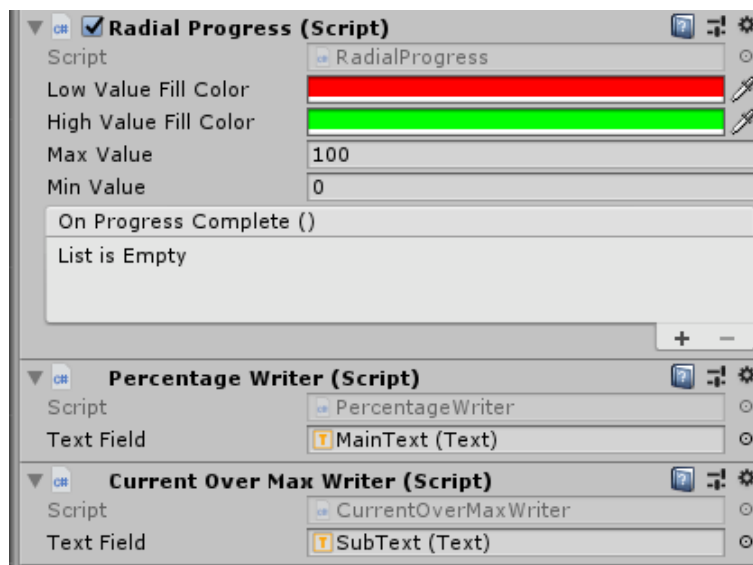
**Low Value Fill Color:** Color of the circle displaying progress when current value is equal to Min Value. Asset color linearly interpolates as current value changes.

**High Value Fill Color:** Color of the circle displaying progress when current value is equal to Max Value. Asset color linearly interpolates as current value changes.

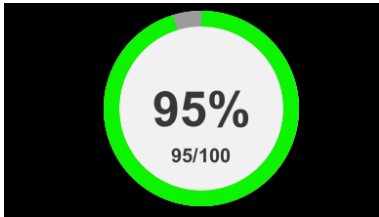
**Max Value:** Maximum value progress is expected to reach. Displays %100 when current value is equal to Max Value.

**Min Value:** Minimum value progress is expected to reach. Displays %0 when current value is equal to Min Value.

**On Progress Complete:** List of event handlers you can add to listen to progress complete event which is fired whenever current value is equal to Max Value.

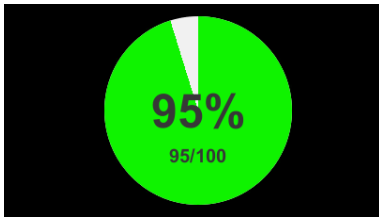


### *ProgressCircleThin*



ProgressCircleThin works the same way as ProgressCircleThick while utilizing a thinner circle.

### *ProgressCircleFull*



ProgressCircleFull works the same way as ProgressCircleThick while utilizing a non-hollow circle.

### *HPCircleLinear*



HPCircleLinear is a circular health bar that can change color as its value changes and includes a late-updating tracer which makes it easier to visualize damage taken. It has a gradient image backing it so lower parts of the circle will be effected more by its background color. It has the following properties:

**Low Value Fill Color:** Color of the circle displaying progress when current value is equal to Min Value. Asset color linearly interpolates as current value changes.

**High Value Fill Color:** Color of the circle displaying progress when current value is equal to Max Value. Asset color linearly interpolates as current value changes.

**Max Value:** Maximum value progress is expected to reach. Displays %100 when current value is equal to Max Value.

**Min Value:** Minimum value progress is expected to reach. Displays %0 when current value is equal to Min Value.

**Default Text Color:** Color of the center text value when current value is above the low value trigger percentage.

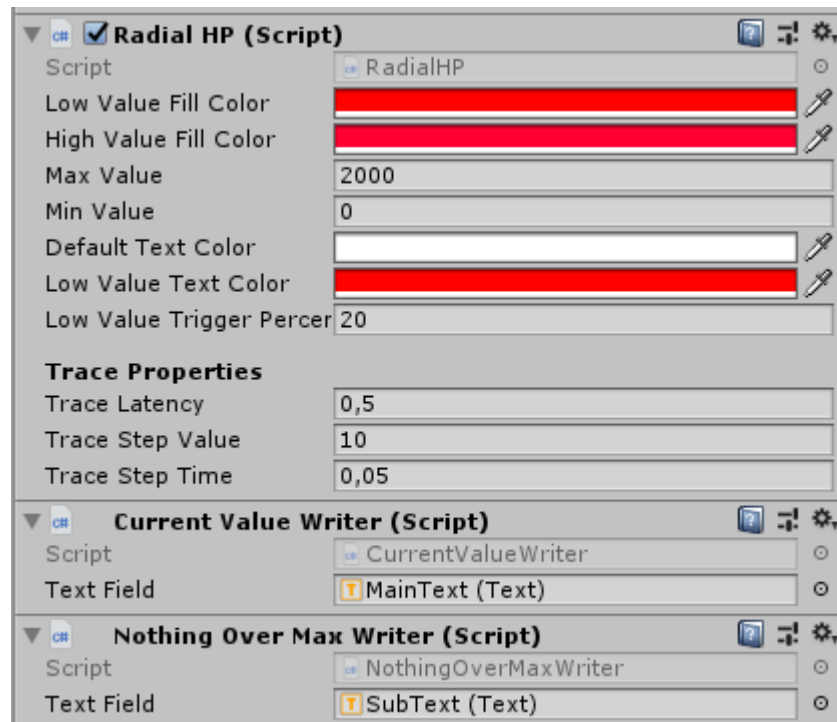
**Low Value Text Color:** Color of the center text value when current value drops below the low value trigger percentage.

**Low Value Trigger Percentage:** Percentage based threshold value that will trigger the text color change.

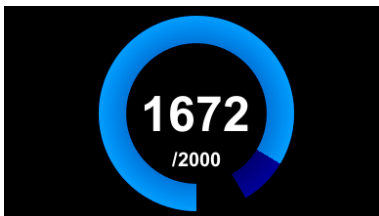
**Trace Latency:** Total time (in seconds) that the tracer circle will wait on its current value after the assets current value is reduced. If the assets current value is zero or greater than the tracer's current value the delay is ignored and tracing circle immediately takes the same value as the asset.

**Trace Step Value:** The amount of decrement tracing circle value will make with each trace tick.

**Trace Step Time:** Time (in seconds) each tracer tick takes. Assigning a too small value to this parameter will make the asset use unnecessarily larger amounts of processor power while a too large value may cause the tracing operation to be too slow. Finally a large step time and a large step value may cause this asset to look as if it is lagging.



### HPCircleThick



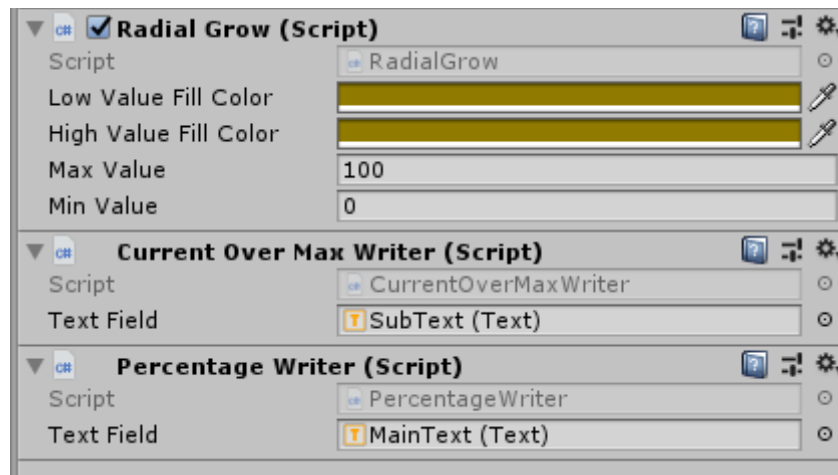
HPCircleThick works the same way as HPCircleLinear while utilizing a similar look and feel to ProgressCircleThick.

### GrowingCircle



GrowingCircle works in a similar way to ProgressCircleFull with the following differences:

- It utilizes a growing inner circle instead of a radially expanding one.
- It does not fire progress complete events and thus does not have the “On Progress Complete” parameter.



## Text Writers

Text writers are the script files that manage the logic of updating the **MainText** (the large ones in the center) and **SubText** (the smaller ones under the main ones) fields' values in ProgressCircle assets. The asset pack comes with the following text writers:

- Percentage Writer
- Current Over Max Writer
- Current Value Writer
- Nothing Over Max Writer

### Percentage Writer

This writer prints the result of following code on its text field:  
`(progressCircleInstance.FillPercentage * 100).ToString("0") + "%"`

### Current Over Max Writer

This writer prints the result of following code on its text field:  
`progressCircleInstance.CurrentValue.ToString("0") + "/" + progressCircleInstance.MaxValue`

### Current Value Writer

This writer prints the result of following code on its text field:  
`progressCircleInstance.CurrentValue.ToString("0")`

### Nothing Over Max Writer

This writer prints the result of following code on its text field:  
`"/" + progressCircleInstance.MaxValue`

## Extending ProgressCircle

Each ProgressCircle asset contains a core script (subclasses of AbstractRadial) and up to two text writers (subclasses of AbstractTextWriter). You can also extend these classes yourself and create similar assets that behave as you need them to. Svg files that are used for creating asset images are also kept with the asset pack in the "ProgressCircle/Images/svg/" folder. If you need a different look for these assets in your game and changing options in the Unity editor's inspector is not sufficient, you can easily modify the matching svg file with a vector graphics software (like Inkscape) and use the exported png image for your unique asset.