WEREWOLF

STATUS INDICATORS

DEVELOPER'S GUIDE

unity

Contents

Quick Start

If you own the previous version “Spell Indicators”, first of all thank you for purchasing it as its popularity inspired me to make the upgrade.

Please be aware that there are many differences and it would be best to go through the manual in order to understand the changes.

After importing Status Indicators, have a go at one of the demos located under Werewolf > StatusIndicators > Demos > Example Usage.

If you would like to use this as a template, just remove the “Demo Progress Loop” components from each indicator.

Installing into your own project

Go to the Project window and find the “Examples” folder located under Werewolf > StatusIndicators > Prefabs.

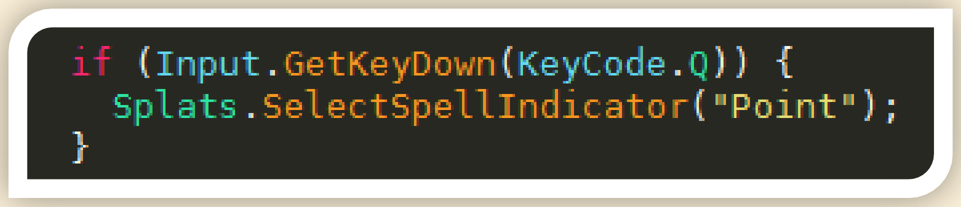
Drag “Example1” onto your scene.

The dummy character (capsule) that contains the “SplatManager” will have some example code that you can refer to in the “CharacterDemo3” script.

Press play and test it!

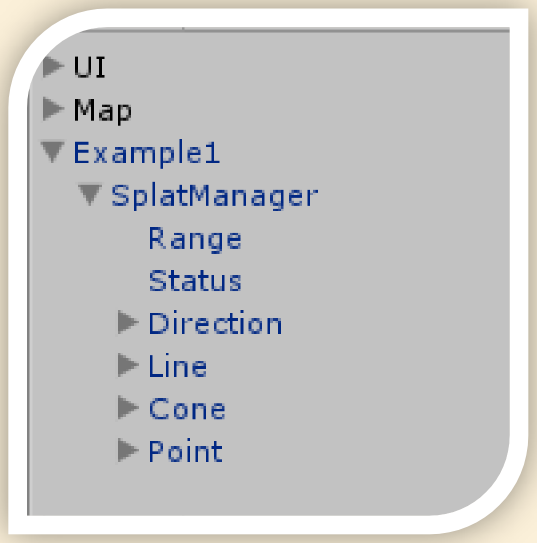
Expanding on the functionality

If you open "CharacterDemo3" you will notice some code that tells the Splat Manager which indicators to display depending on the keys pressed.



```
if (Input.GetKeyDown(KeyCode.Q)) {
  Splats.SelectSpellIndicator("Point");
}
```

The Splat Manager automatically aggregates all indicators nested inside the Splat Manager's hierarchy which you can then select with code. If the hierarchy has not been modified it should look like this.



```
▶ UI
▶ Map
▼ Example1
  ▼ SplatManager
      Range
      Status
    ▶ Direction
    ▶ Line
    ▶ Cone
    ▶ Point
```

It is very simple to tell the manager to select the appropriate indicator. After adding a new indicator to the manager's hierarchy, tell the Splat Manager to select that particular object by object name.

If it is a Spell Indicator use the following,

```
Splats.SelectSpellIndicator("Spell Indicator Name");
```

Or if it is a Status Indicator,

```
Splats.SelectStatusIndicator("Status Indicator Name");
```

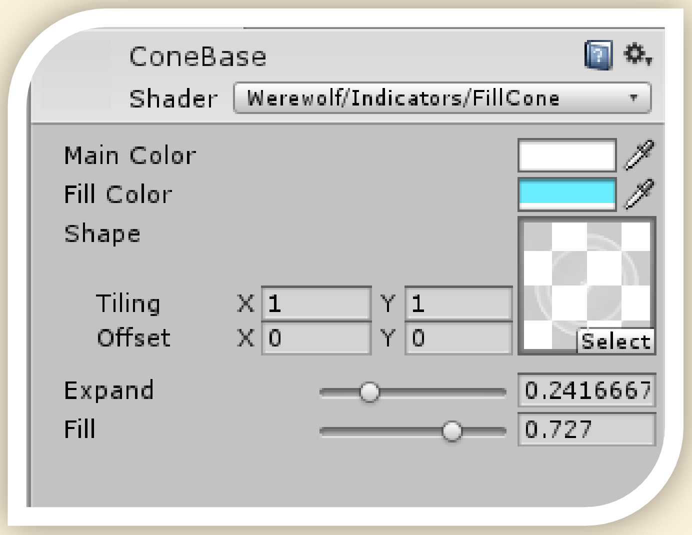
Or if it is a Range Indicator,

```
Splats.SelectRangeIndicator("Range Indicator Name");
```

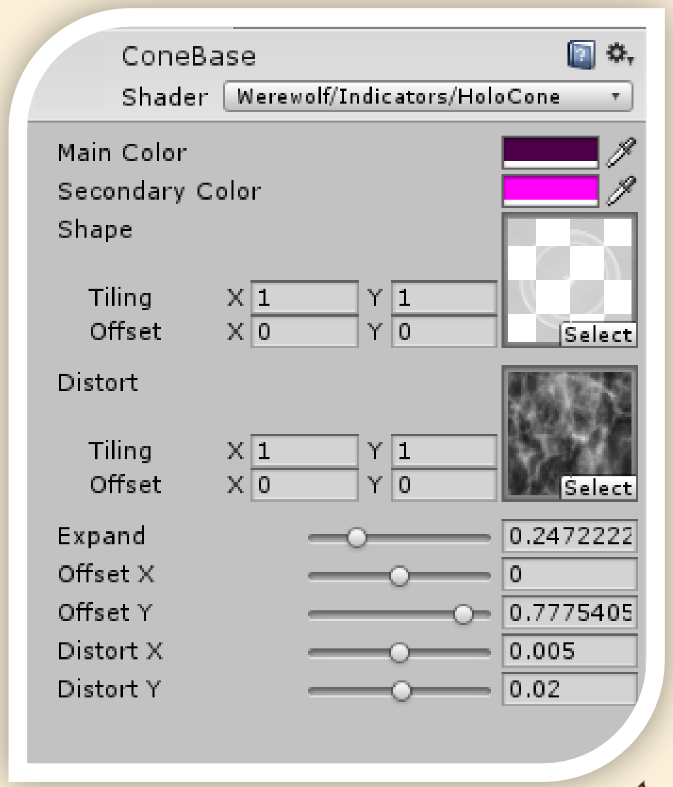
Basic Customization

Before changing the Material values such as colors, it is recommended to copy the prefab and the materials (and reference the materials to the new prefab) so that the original prefabs aren't modified. Otherwise adding the original prefabs later may have undesirable effects. You don't have to worry about it if it is not an issue.

There are 2 colors for each material, either containing Main/Secondary color or Main/Fill color. Fill Color refers to the "progress bar" fill color of a texture, whereas Secondary Color is used for the Distortion or Holo shaders for color scheme.



ConeBase
Shader Werewolf/Indicators/FillCone
Main Color
Fill Color
Shape
Tiling X 1 Y 1
Offset X 0 Y 0
Expand 0.2416667
Fill 0.727



ConeBase
Shader Werewolf/Indicators/HoloCone
Main Color
Secondary Color
Shape
Tiling X 1 Y 1
Offset X 0 Y 0
Distort
Tiling X 1 Y 1
Offset X 0 Y 0
Expand 0.2472222
Offset X 0
Offset Y 0.7775405
Distort X 0.005
Distort Y 0.02

Compatibility

If you are supporting low spec laptops or desktops, the Holo and Distort shaders use Shader Model 3.0 and may have performance or compatibility issues. The Holo shaders require most performance followed by Distort shaders and then the Fill shaders.

It is recommended (or at least have a graphic setting) to allow players to use the simpler, higher performing shaders in its place. For high graphics quality games this shouldn't be an issue;

Low Spec Shaders (Shader Model 2.0)

- Werewolf/Indicators/Basic
- Werewolf/Indicators/FillCone
- Werewolf/Indicators/FillLinear
- Werewolf/Indicators/FillRadial

Indicator Types

There are three general types of indicators. **Spell** Indicator, **Status** Indicator and **Range** Indicator.

Within Spell Indicators there are also four specialized subtypes. **AngleMissile**, **LineMissile**, **Cone** and **Point**.

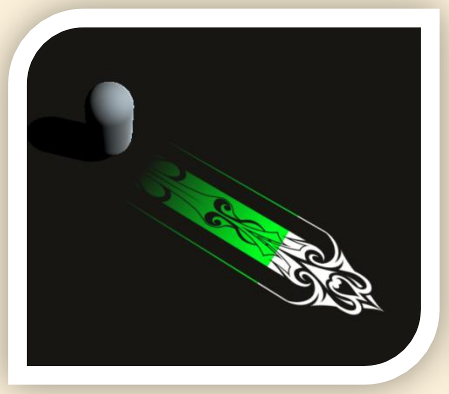
All variables are dynamic and can be modified during run-time. Allowing you to re-use the same indicators when abilities are upgraded, or between different abilities that use the same indicator.

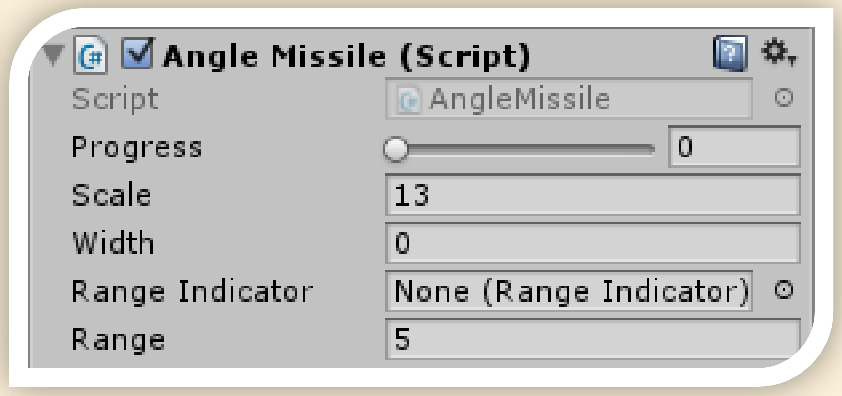
Spell Indicators

Spell Indicators are designed for skill shots and abilities to assist players with targeting and gauging the range of spells. Range Indicators can be used in tandem with Spell Indicators to add more granularity and control in assisting targeting. Spell Indicators also have a progress meter for spells that require a charge before firing.

Angle Missile

The Angle Missile component is for directional abilities with a fixed or infinite distance.





Angle Missile (Script)

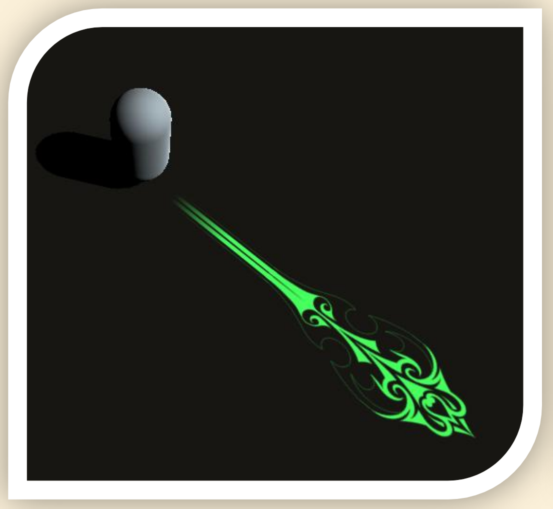
Script	AngleMissile
Progress	0
Scale	13
Width	0
Range Indicator	None (Range Indicator)
Range	5

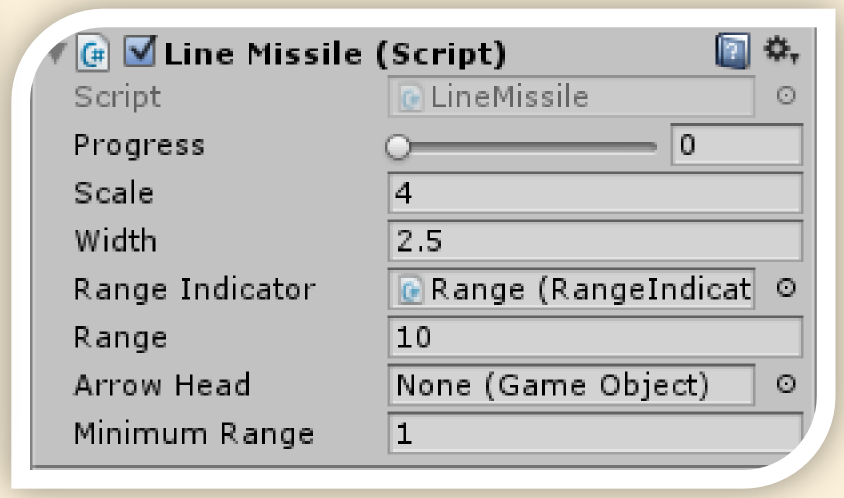
Angle Missile Attribute Description;

- **Progress:** Where 0 is empty and 1 is full. For power up abilities you can adjust the progress so that the splat fills up a different color accordingly. Color can be adjusted in material.
- **Scale:** The overall size of the splat.
- **Width:** Make the splat narrower or wider.
- **Range Indicator:** Automatically render a Range Indicator when the indicator is selected. The Range Indicator will automatically adjust its size according to the Range attribute below.
- **Range:** Refers to the size of the Range Indicator if displayed.

Line Missile

The Line Missile component behaves like the Angle Missile component, only that the length is dynamic and can be controlled by the player's mouse (stretches to the mouse from the character). This is useful for abilities where the distance needs to be controlled by the player.





Line Missile (Script)

Script	LineMissile
Progress	0
Scale	4
Width	2.5
Range Indicator	Range (RangeIndicat
Range	10
Arrow Head	None (Game Object)
Minimum Range	1

Line Missile Attribute Description;

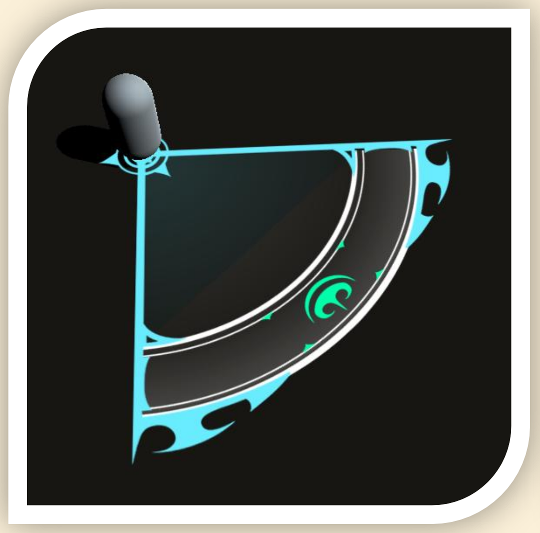
- **Progress:** While this can used, the progress component is not suitable for this component due to the dynamic length.
- **Scale:** The overall size of the splat.
- **Width:** Make the splat narrower or wider.
- **Range Indicator:** Automatically render a Range Indicator when the indicator is selected. The Range Indicator will

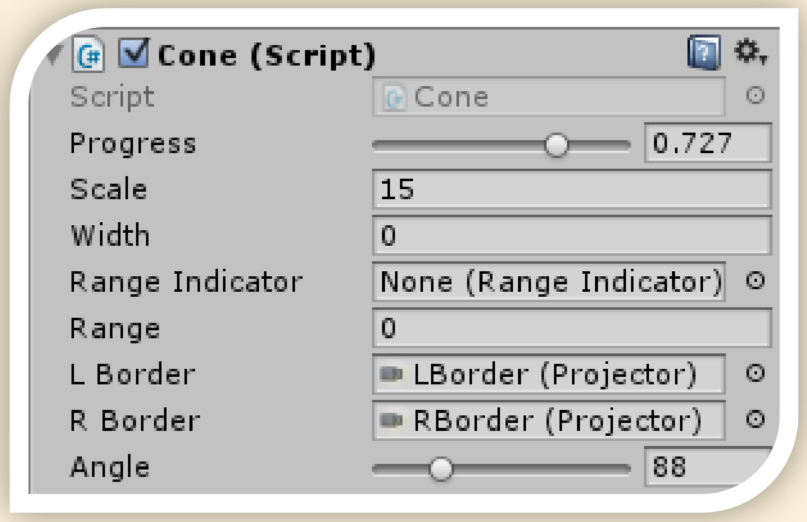
automatically adjust its size according to the Range attribute below.

- **Range:** Restricts the cursor or splat length to the range specified.
- **ArrowHead:** Refers to the object that will serve as the "Arrow Head" of the Line Missile component so that it doesn't get stretched.
- **Minimum Range:** Restricts the player from reducing the Line Missile component to a too smaller size.

Cone

The Cone component is for Area of Effect abilities that project outwards from the character. The Cone can be adjusted for a longer, shorter, wider or narrower arc.





```
Cone (Script)
Script              Cone
Progress            0.727
Scale               15
Width               0
Range Indicator     None (Range Indicator)
Range               0
L Border            LBorder (Projector)
R Border            RBorder (Projector)
Angle               88
```

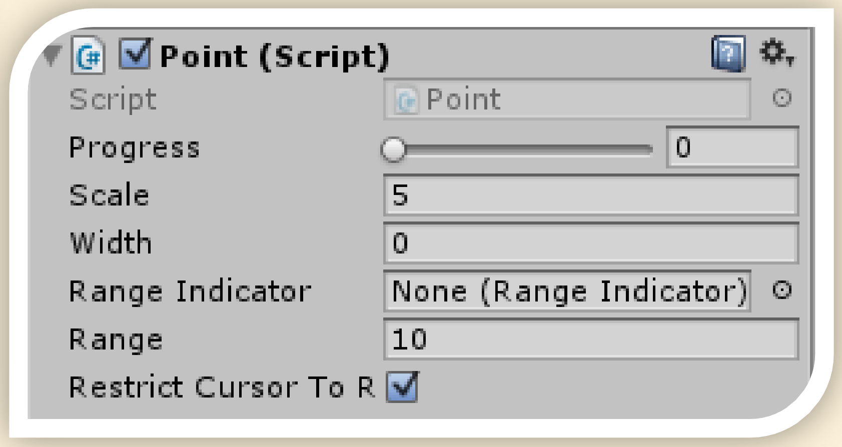
Cone Attribute Description;

- **Progress:** For power up abilities you can adjust the progress so that the splat fills up a different color accordingly.
- **Scale:** The overall size of the splat.
- **Width:** Make the splat narrower or wider.
- **Range Indicator:** Automatically render a Range Indicator when the indicator is selected. The Range Indicator will automatically adjust its size according to the Range attribute below. Not really necessary for a Cone.

- **Range:** Refers to the size of the Range Indicator if displayed.
- **LBorder, RBorder:** These are the borders of the cone that will wrap the graphic and rotate as the cone angle expands or shrinks.
- **Angle:** You can adjust the arc of the cone with the angle attribute and make it wider or narrower from 0 to 360 degrees.

Point

The Point component is for targeted abilities with no origin. It simply follows the mouse. It is normally used for targeting an area or individual character.



Point (Script)

Script	Point
Progress	0
Scale	5
Width	0
Range Indicator	None (Range Indicator)
Range	10
Restrict Cursor To R	✓

Point Attribute Description;

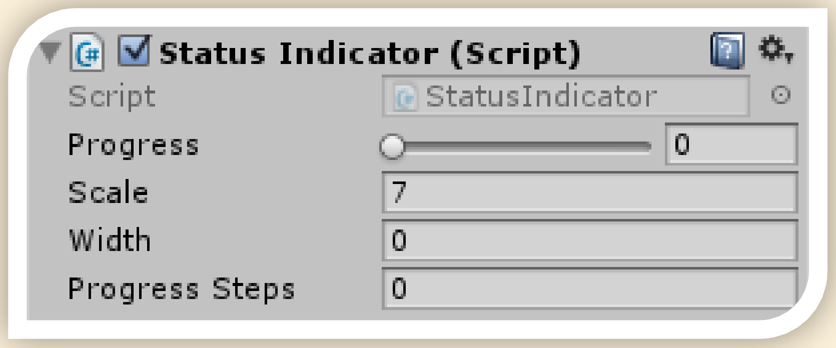
- **Progress:** For power up abilities you can adjust the progress so that the splat fills up a different color accordingly. For a Point this is normally a progress bar that rotates in a clockwise direction around the splat.
- **Scale:** The overall size of the splat.
- **Width:** Make the splat narrower or wider.
- **Range Indicator:** Automatically render a Range Indicator when the indicator is selected. The Range Indicator will

automatically adjust its size according to the Range attribute below.

- **Range:** Distance from origin ability can be used, affects the Range Indicator size and Restrict Cursor to Range function.
- **Restrict Cursor to Range:** Optionally restrict the cursor to the range specified.

Status Indicators

Status Indicators are designed to visually represent the attributes of the character, such as health, mana, or power up abilities. In contrast to Spell Indicators, Status Indicators do not have any specialized components and you simply use the base class.



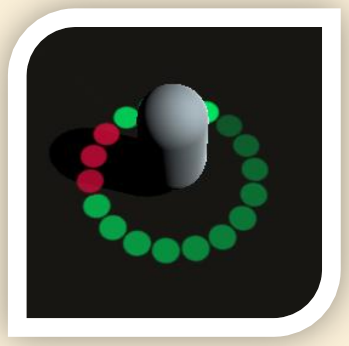
Status Indicator (Script)

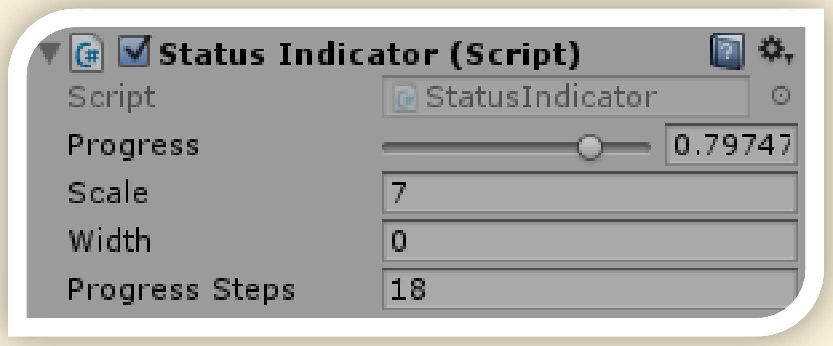
Script	StatusIndicator
Progress	0
Scale	7
Width	0
Progress Steps	0

Status Indicator Attribute Description;

- **Progress:** Normalized value that fills up the bar a different color. Can be used to indicate attributes such as health or mana, or a charge up meter for an ability. Color can be adjusted in material.
- **Scale:** The overall size of the splat.
- **Width:** Make the splat narrower or wider.
- **Progress Steps:** How many steps/increments does this progress bar have? 0 is smooth fill. Anything above 0 will round to the nearest step in a staggered format. 0 is ideal for

most formats, but sometimes you want a dotted format such as the example below.

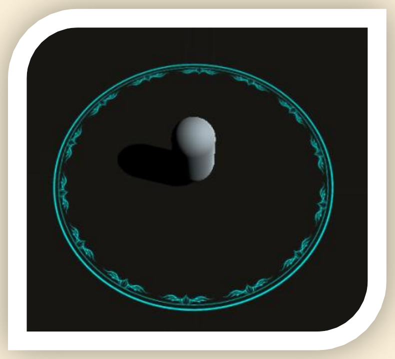


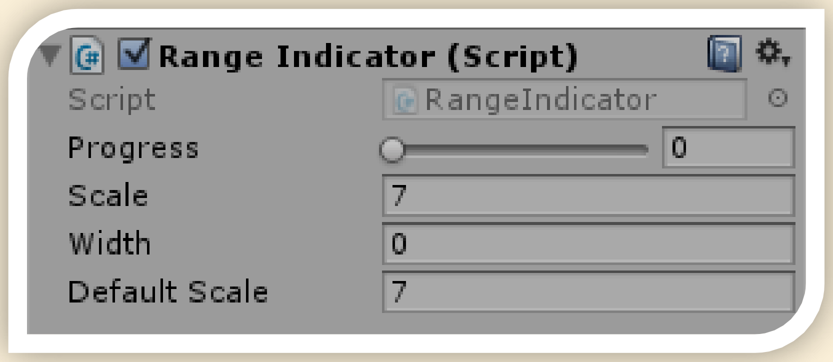


Script	StatusIndicator
Progress	0.79747
Scale	7
Width	0
Progress Steps	18

Range Indicators

Range Indicators represent boundaries of abilities or attacks. They can be associated with SpellIndicators by referencing them in the RangeIndicator property, but they can also be shown on its own for other reasons such as auto-attacks using SelectRangeIndicator().





Range Indicator (Script)	
Script	RangeIndicator
Progress	0
Scale	7
Width	0
Default Scale	7

Range Indicator Attribute Description;

- **Progress:** Normalized value that fills up the bar a different color. Can be used to indicate attributes such as health or mana, or a charge up meter for an ability. Color can be adjusted in material.
- **Scale:** The overall size of the splat.
- **Width:** Make the splat narrower or wider.
- **Default Scale:** Scale is dynamic when a Spell Indicator references a Range Indicator. This is what the Range Indicator Scale attribute will revert to when the Indicator is selected on its own.

Coding Guide

Initialization

Once you have placed the SplatManager in your character hierarchy. Use this format in your character component to get started.

```
public SplatManager Splats { get; set; }
void Start() {
  Splats = GetComponentInChildren<SplatManager>();
}
```

Splat Manager

Get position of indicator

#GetSpellCursorPosition()

Sometimes the raytraced 3d mouse position is not directly aligned with the indicator, such as when a Point indicator is restricted to the Range Indicator bounds. It is best to use GetSpellCursorPosition() to make sure that you always get the position of the spell indicator which makes more intuitive sense to the player.

Show an indicator

#Select Spell Indicator(string name)

#SelectStatusIndicator(string name)

#SelectRangeIndicator(string name)

```
Splats.SelectSpellIndicator("Spell Indicator Name");
```

You can select any Indicator as long as it is nested within the Splat Manager's hierarchy. Simply reference it by the object's name as it is in the hierarchy.

Hide the selected indicator

#CancelSpellIndicator()

#CancelStatusIndicator()

#CancelRangeIndicator()

Use this to hide the indicator that is currently showing. Note that hiding one of them does not hide all others. So you can show a Status Indicator while showing a Spell Indicator.

Retrieve an indicator

#CurrentSpellIndicator

#CurrentStatusIndicator

#CurrentRangeIndicator

#GetSpellIndicator(string name)

#GetStatusIndicator(string name)

#GetRangeIndicator(string name)

If at any point you need to retrieve an indicator to do specific tasks, you can do so by using the above methods.

Creating new Indicators

Create the image

First you will need an image to use as your Spell Splat. You can use one of the textures already supplied under Werewolf > StatusIndicators > SplatTextures, or you can roll your own.

Requirements

- Transparent PNG or similar
- Use White color only, to allow customization of color in game.
- The image should be square, non-square images will look deformed.
- For AngleMissile and Cone, you will need to place the image in the top half of the image and leave the bottom half empty.

Import the image into Unity. In the image settings set "Wrap Mode" to Clamped.

Create the Material

Create a new Material with, Right-Click -> Create -> Material

Select one of the Werewolf > Indicators > Shaders

In the new material select the Image that you want to use for your Indicator. Adjust colors to your preference. Some materials have a distortion component, select one of the distort textures under Werewolf > StatusIndicators > DistortTextures.

Add a Projector

Add a Projector to the scene and set the Material. Rotate the projector so it faces the ground. Adjust variables as necessary. Aspect Ratio and Scale are adjusted automatically by the Indicator component.

Add Indicator Component

Add one of the Indicator components to the object depending on what type of indicator you want. You can also create a base object with the component, and add several child objects with individual projectors. This will be necessary for some components like Cone and Line Missile, where there are moving objects.

Apply Splat to Scene

Create a "Splat Manager" if you haven't already, by creating a new GameObject and applying the SplatManager script. The Splat Manager will automatically reference to all child indicators inside it.

Drag the new Indicator you have created onto the Splat Manager, so that the Splat's parent is the Splat Manager.

You are now ready to test your new indicator.

Additional Components

Projector Rotate

This component provides a rotation effect on the projector for things like Range Indicators to make them more interesting. It rotates independently from its parent so if a character rotates it won’t rotate with it.

Projector Fixed Rotation

Fixed Rotation causes the projector rotation to stay the same despite parent rotation. It is useful for Status Indicators so that it doesn’t rotate with the character making it hard to see.

Linear Distort

Radial Distort is used to provide animation for the “HoloLinearDistort” shader effects. Usually used for non-circular indicators like Angle Missile and Cone.

Radial Distort

Radial Distort is used to provide animation for the “HoloRadialDistort” shader effects. Usually used for circular indicators like Point and Range.

Thank you

for purchasing Status Indicators