DTT
APPS WEB GAMES

SINCE 2010

Area Of Effect Regions

Documentation | V1.1.0 | 30-06-22

Table of Contents

1. Get started quickly 3

2. Introduction 5

3. Set-Up 6

4. Projectors 7

Mesh Indicators 7

Projector Indicators 7

5. API 8

6. Known Limitations 9

7. Support and feedback 10

1. Get started quickly

To get started you can check out the demo scene in "Assets/DTT/Area of Effect Regions/Demo/Mesh Indicators/Scenes/Demo" for the mesh based effects or "Assets/DTT/Area of Effect Regions/Demo/Projector Indicators/Scenes/Demo" for the projector based effects. Here you can see all the implementations of the different regions we support.

To use this on one of your own GameObjects you can add one of the prefabs in the "Assets/DTT/Area of Effect Regions/Demo/Mesh Indicators/Prefabs" for the mesh based prefabs, or in the "Assets/DTT/Area of Effect Regions/Demo/Projector Indicators/Prefabs" for the projector based prefabs.

Circle region

For the circle region you can make use of a radius to set the size of your circle, and an offset to move it relative from the GameObject.

Line region

The line region makes use of an angle so you can direct which way it should point, a length for how far it should reach out from the GameObject it's on, and a width for setting the thickness of the line region.

Arc region

To set up the arc region you have the option to set the arc in angles, you can adjust the angle to make it point in the direction you want it to, and you can tweak the radius of the arc.

Scattered Lines Region

The scattered lines region makes use of multiple **Line Regions**, by spreading the lines in an arc formation, which also has the option of changing the angle for it as well. Same as with the Line Region, you can change the width and length of the lines.

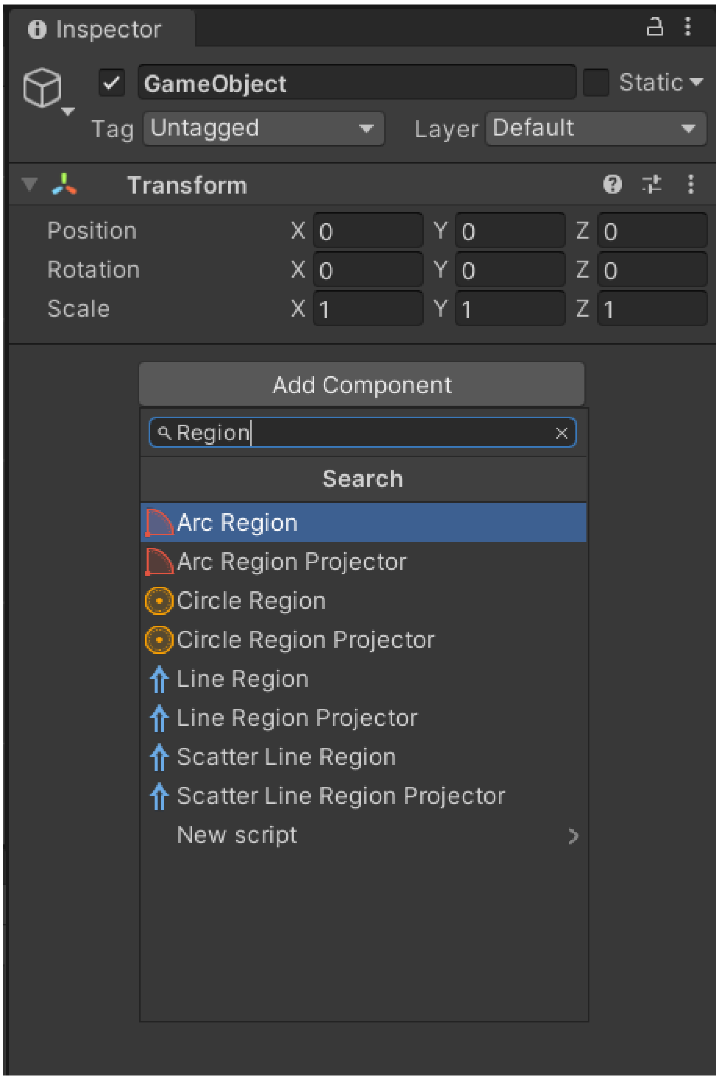
2. Introduction

Area of Effect Regions is an asset to help you define easier regions in your game/application. A common use case for this asset is to use it as indicators when you want to telegraph an attack or skill to your user in an MMO game. It's designed to be extendable by your own implementation so you can customize it as much as you see fit. It also contains an example implementation for every region we offer, so you can immediately see how it can be used.

3. Set-Up

To set up any of the region components, you can easily add them to your script using the "Add Component" menu.

1. Inspect your desired **GameObject** and open the "Add Component" menu.
2. In the menu, search for the region script you wish to add. You can either use the mesh or projector based regions.
3. Voilà! You've now successfully added a region component.



GameObject
Tag Untagged
Layer Default

Transform			
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Region

4. Projectors

Mesh Indicators

The mesh indicators, as mentioned in the name, are based on meshes to show an area of an effect. The advantages of the mesh indicators are that they work on every Unity rendering pipeline and can be rendered in the 'air', meaning that there is no need for a surface for them to be rendered on.

The later one, has its own shortcoming of being able to show the effects only at the position on which the mesh indicator game object is, therefore these effects cannot be used as projectors to be rendered onto flat surfaces. Nonetheless, these mesh effects do not work well on uneven terrain because of clipping with other meshes and Z fighting.

Projector Indicators

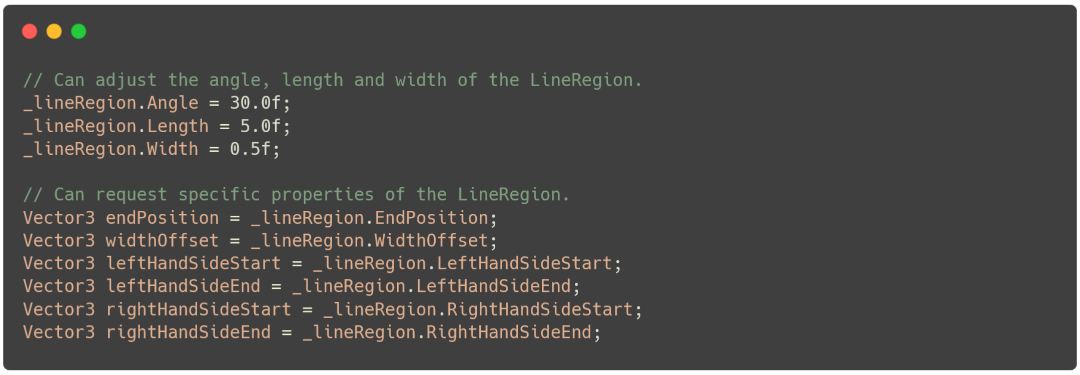
The projector indicators make use of the default Unity **Projector** component. This component only needs a material, which it can project onto whatever surface you want. However, the downside of these projector effects is that these are not supported in the URP and HDRP pipelines. Apart from this downside, these projector indicators work well on uneven terrain/surfaces, and they only show on the nearest surfaces at which they are pointed at. The projector has a near and far clip plane, in other words, rendering distance, which can be adjusted as well. The projectors have the option of choosing which layers to ignore or not.

Decal Projector Indicators (SRP/URP/HDRP)

The projector implementation supported by SRP based renderer is using the DecalProjector component from URP. This component is similar to the projector indicators. The only difference is that you have to set the depth of projection so the projector reaches your surface.

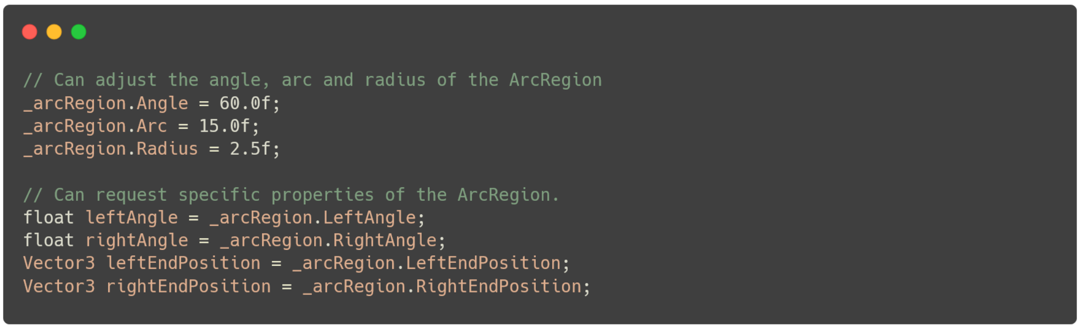
However the decalProjector component has been implemented in URP 12, the SRP implementation is therefore only available for unity version 2021.2 and higher.

5. API



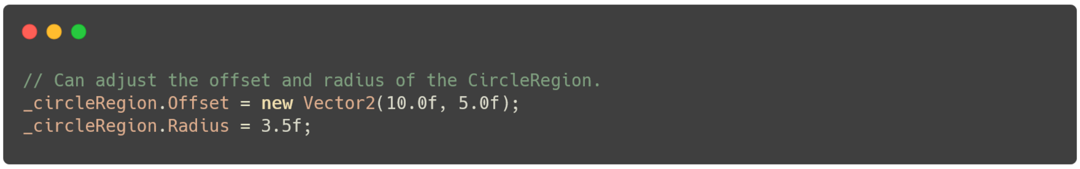
```
// Can adjust the angle, length and width of the LineRegion.
_lineRegion.Angle = 30.0f;
_lineRegion.Length = 5.0f;
_lineRegion.Width = 0.5f;

// Can request specific properties of the LineRegion.
Vector3 endPosition = _lineRegion.EndPosition;
Vector3 widthOffset = _lineRegion.WidthOffset;
Vector3 leftHandSideStart = _lineRegion.LeftHandSideStart;
Vector3 leftHandSideEnd = _lineRegion.LeftHandSideEnd;
Vector3 rightHandSideStart = _lineRegion.RightHandSideStart;
Vector3 rightHandSideEnd = _lineRegion.RightHandSideEnd;
```



```
// Can adjust the angle, arc and radius of the ArcRegion
_arcRegion.Angle = 60.0f;
_arcRegion.Arc = 15.0f;
_arcRegion.Radius = 2.5f;

// Can request specific properties of the ArcRegion.
float leftAngle = _arcRegion.LeftAngle;
float rightAngle = _arcRegion.RightAngle;
Vector3 leftEndPosition = _arcRegion.LeftEndPosition;
Vector3 rightEndPosition = _arcRegion.RightEndPosition;
```



```
// Can adjust the offset and radius of the CircleRegion.
_circleRegion.Offset = new Vector2(10.0f, 5.0f);
_circleRegion.Radius = 3.5f;
```

6. Known Limitations

1. The regions are designed to only use rotation on the y-axis, z and x rotations are not inherently supported.
2. The Projector based effects/regions are not supported in the URP and HDRP pipelines for unity version lower than 2021.2

7. Support and feedback

If you have any questions regarding the use of this asset, we are happy to help you out. Always feel free to contact us at:

unity-support@d-tt.nl

(We typically respond within 1-2 business days)

We are actively developing this asset, with many future updates and extensions already planned. We are eager to include feedback from our users in future updates, be they 'quality of life' improvements, new features, bug fixes or anything else that can help you improve your experience with this asset. You can reach us at the email above.

Reviews and ratings are very much appreciated as they help us raise awareness and to improve our assets.

DTT stands for Doing Things Together

DTT is an app, web and game development agency based in the centre of Amsterdam. Established in 2010, DTT has over a decade of experience in mobile, game, and web based technology.

Our game department primarily works in Unity where we put significant emphasis on the development of internal packages, allowing us to efficiently reuse code between projects. To support the Unity community, we are publishing a selection of our internal packages on the Asset Store, including this one.

More information about DTT (including our clients, projects and vacancies) can be found here:

https://www.d-tt.nl/en/