

monoflow's metablobs Manual

vers. 1.01



Contents

- 1 Introduction** **1**

- 2 Requirements** **3**

- 3 Usage** **5**

- 4 Settings** **7**
 - 4.1 Targets 7
 - 4.2 Transparency & Blending 7
 - 4.3 Blobsize 8
 - 4.4 Border 9
 - 4.5 Posterize 11
 - 4.6 Outline 12
 - 4.7 RenderTexture 13

- 5 Scriping API** **15**

- 6 History** **17**

- 7 Support** **19**

Chapter 1

Introduction

- **Monoflow's Metablobs (MMB)** brings the oldschool Metablob effect to Unity.
- A Metablob is defined as a point in space that has a circular influence area. A single Metablob looks like a normal smooth circle, but if you have several Metablobs you see the organic shape of the interacting influence areas.



Figure 1.1: Default Metablob

Depending on your settings you can achieve very interesting visual results. Often they remind to some Liquid light shows from the psychedelic era. You could use MMB to visualize some sort of liquids, but as of the high rendering cost the amount of particles that could be used will be too low for a convincing liquid simulation in most use cases.

- **MMB** defines a Metablob with position, size and color. The values for these properties come from normal GameObjects. It uses the positions (in camera space), the size and additionally the color from the material of the GameObject for the Metablob calculation.

Chapter 2

Requirements

- Unity PRO
- Microsoft Windows with a DX 11 capable graphics card

Chapter 3

Usage

- You can use **MMB** as a normal fullscreen camera effect or render to a RenderTexture to use the image later in your scene.
- For both scenarios it is recommended to use an orthographic camera for the Metablob rendering camera.
- Fullscreen Effect:
 - **MMB** renders as a normal image effect on top of the current camera image.
 - Depending on your settings you can also see the normal 3D objects from your scene, but the effect has only a 2D character.
- Render To Texture:
 - The advantage of this option is that you can use the image of the Metablobs and attach it to a quad in you scene, so you can create effects that mix your 3D objects with the Metablob effect.
 - It is crucial that the quad that displays the Metablob texture has the same ratio like your RenderTexture. Otherwise the blobs don't match with the position of the 3D GameObjects from your scene.

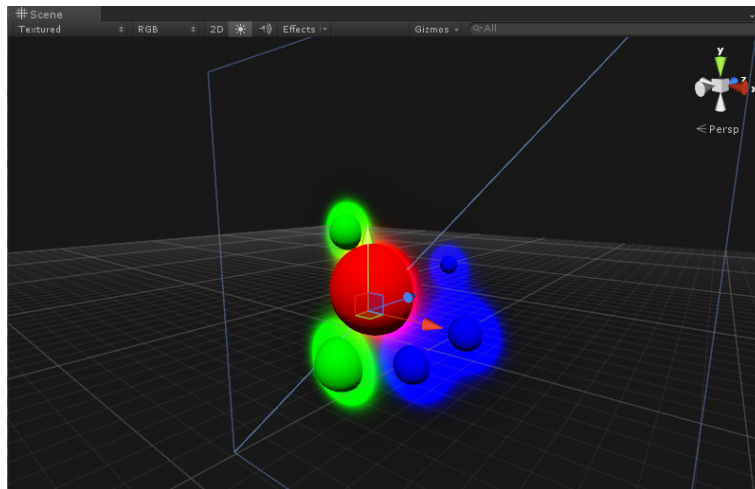


Figure 3.1: Quad with RenderTexture

Chapter 4

Settings

4.1 Targets

- To define a single Metablob you have to add a target (GameObject) to the TargetList.
- To make it easy you can define a special tag. **MMB** adds all GameObjects from your scene that has that tag attached to your TargetList.
- Later on when your app is running you can add/remove additional targets with a [scripting API](#).
- You can also toggle the visibility of the target GameObjects with the 'Show Targets' option.

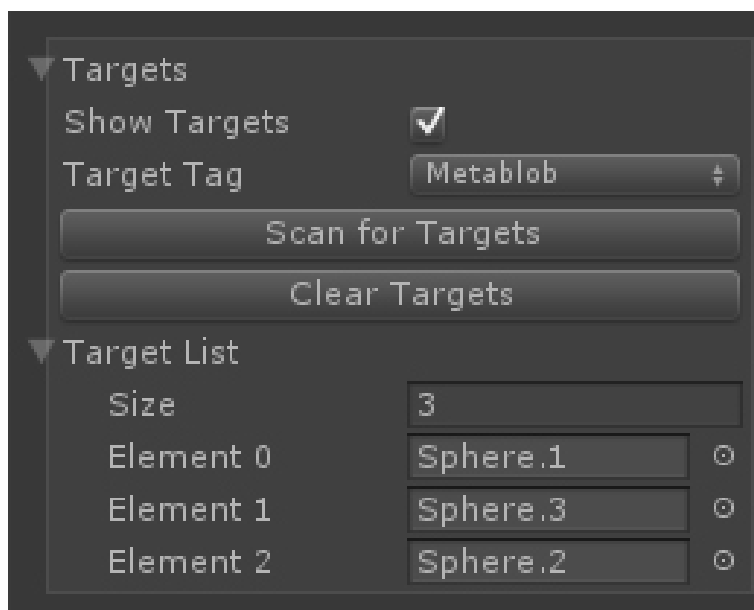


Figure 4.1: Targets

4.2 Transparency & Blending

- You can use the **MMB** with two different shaders. An opaque or a blending one.
- If you use the blending shader for transparency you get additional options.

- With the Blendmode/Blendop options you can change the way the shader blends the effect into the existing framebuffer. For more information look at the Unity documentation (<http://docs.unity3d.com/Manual/SL-Blend.html>). If you just want to change the alpha blending you should use a certain Blendmode and leave the BlendOp to “Add”.
- Amount is the overall transparency.(Only available when using an alpha blend mode)
- CutOff is a value where all pixels below a certain threshold are full transparent. You can invert the CutOff to create cut out effects. If you want to get smooth outlines it is recommended to use the [Outline](#) option instead.

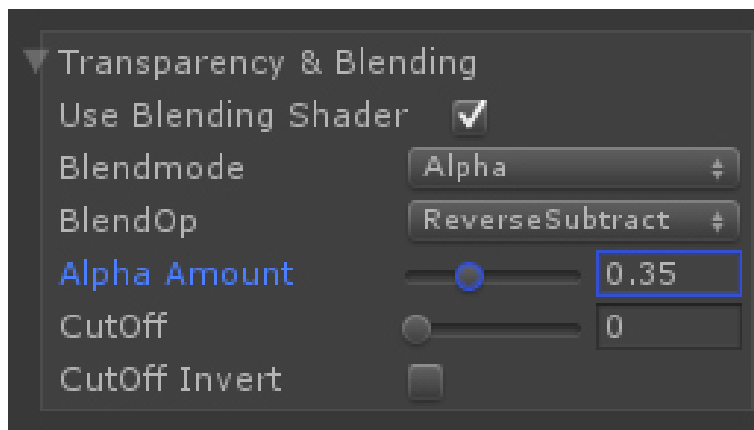


Figure 4.2: Blending

4.3 Blobsize

- Beside the local size of every Metablob that is defined with the scale of the target GameObject you can set an overall factor to change all blobs. (The local size is multiplied with this factor)

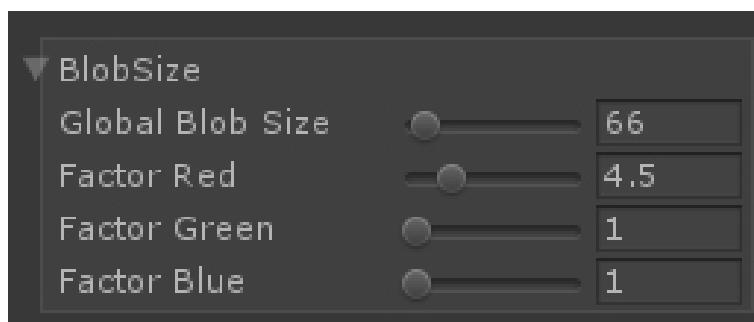


Figure 4.3: Blobsize

- For every color channel you can set an additional factor. With this you can change the size of the blob only in one color channel.



Figure 4.4: Size Factor Red



Figure 4.5: Size Factor Green

4.4 Border

- The border defines the size of the fall off of the influence area
- Larger values create a very smooth blob, smaller values create harder edges.

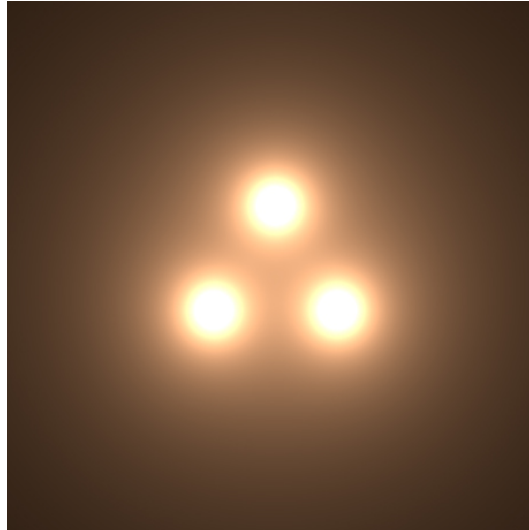


Figure 4.6: Border size 1

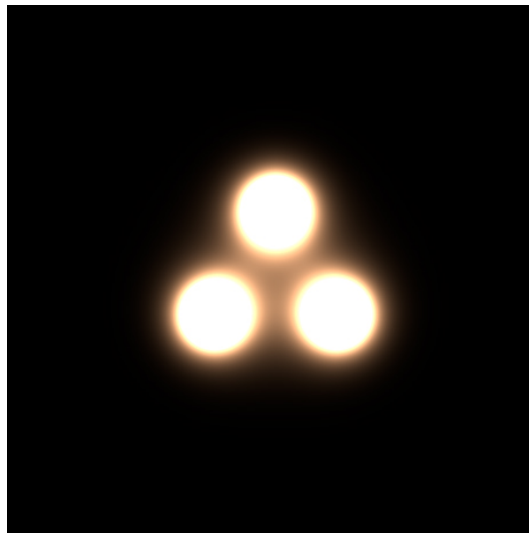


Figure 4.7: Border size 0.2

- For every color channel you can set an additional factor. With this you can create colorful edges.



Figure 4.8: Border Fac

4.5 Posterize

- If you want to create blobs that uses the colors of a texture you can use the posterize option. With this you specify a texture that should be sampled to get the final color.



Figure 4.9: Posterize settings

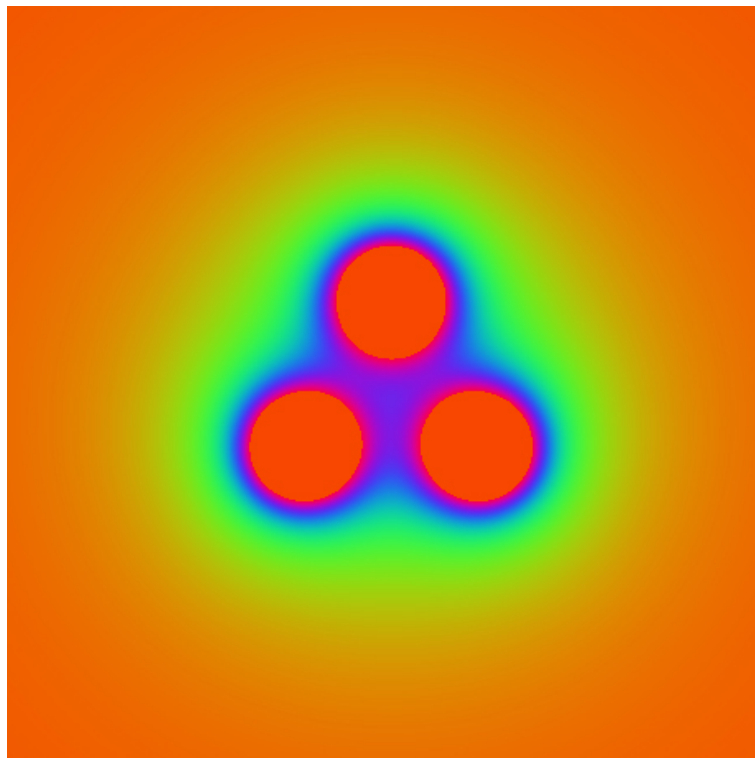


Figure 4.10: Posterize

- The gamma value is a factor for how many posterizing steps should be used. A lower value increase the posterizing effect as there are fewer steps used. But you can also achieve the same effect if you use a texture that is already posterized from image manipulation software.

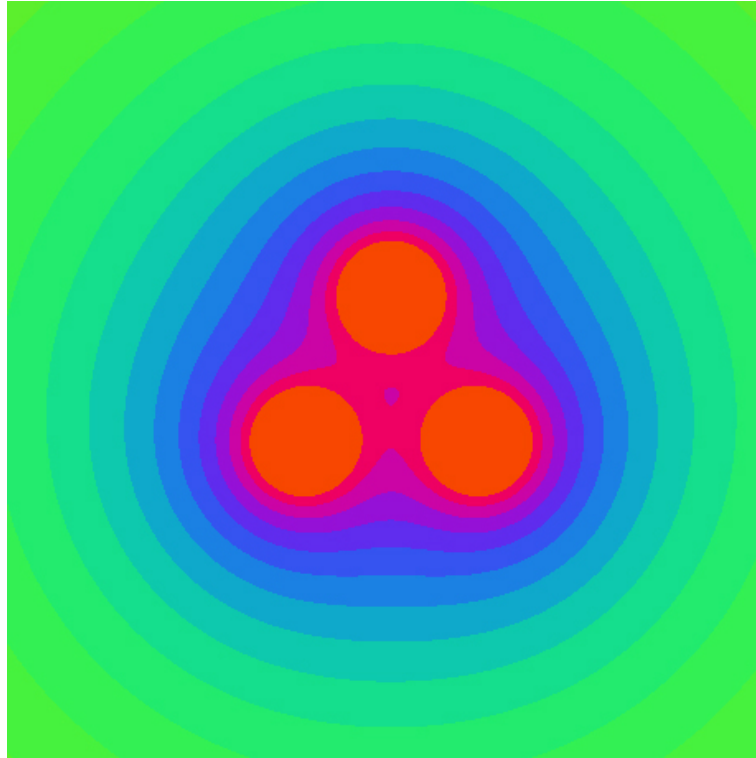


Figure 4.11: Posterize Gamma

- With the offset parameter you can shift the position of the texture sampling.
- To cycle through the different posterizing steps automatically you can set an offset speed. With this you change the offset property in an endless way and can create stunning visual effects.
- The posterizing effect also depends on the Wrapmode of your texture import settings. If you need an seamless color loop you should use Repeat as Wrapmode.

4.6 Outline

- If you only need the outline and don't want to hide the center of your blob you can use this option.

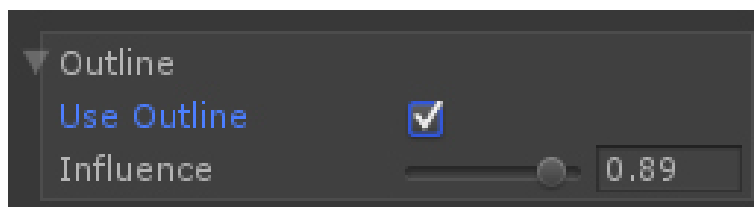


Figure 4.12: Outline settings

- This option works in conjunction with the Border size. You get the best results when the Global Border size is around 0.5 .

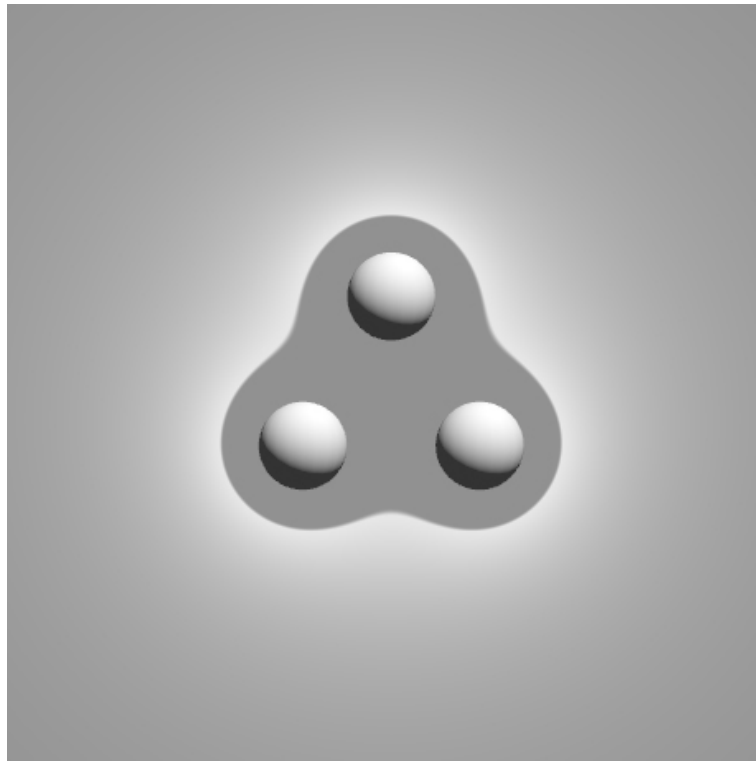


Figure 4.13: Outline

4.7 RenderTexture

- This option is only visible when the camera that **MMB** is attached to renders into a texture.

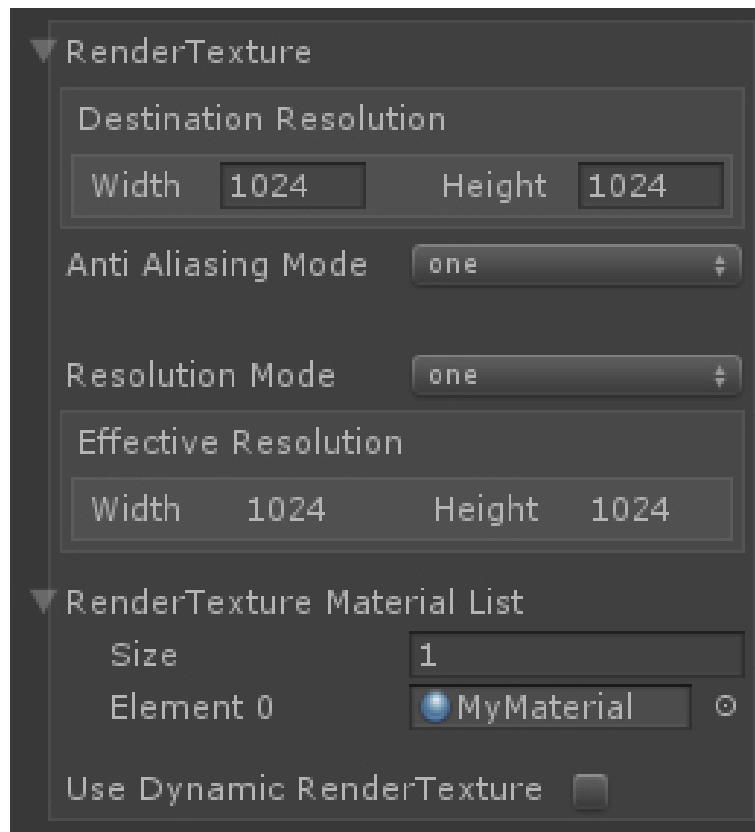


Figure 4.14: RenderTexture settings

- You can set the Resolution of your RenderTexture (Destination)
- Additionally you can select the Anti-Aliasing and Resolution Mode. This comes handy if you have performance problems and want to render at a lower resolution. Normally you can leave the Anti-Aliasing settings to 'One' (No Anti-Aliasing).
- The Effective Resolution shows the real dimension of the RenderTexture that is used.
- As the RenderTexture needs to be destroyed and recreated when you change your settings of the RenderTexture all materials that reference this RenderTexture are losing the reference to it. To auto update a materials with the new texture reference you have to add it to the RenderTexture Material List. **MMB** will update the materials for you.
- The **Use Dynamic RenderTexture** is just an additional option to get an instant feedback how the performance changes when you change the size of your scene. On every change of the GameView size the Rendertexture size is adjusted.(This option should be seen as an experimental feature)

Chapter 5

Scripting API

- All properties you see at the MMB Inspector are accessible via scripts.
- To add a new target you just call `MMB_instance.AddTarget(yourGameObject)`
- To remove a target you just call `MMB_instance.RemoveTarget(yourGameObject)`
- If you want to trigger a new scan for targets with the target tag you call `MMB_instance.SearchForTargets()`
- If you want to empty the whole target list you call `MMB_instance.ClearTargets()`
- If you want to change the used shader type you have to call `MMB_instance.UseAlpha = booleanValue`

Chapter 6

History

Version 1.01 - 2015.01.02

- Bugfix for RenderTexture feature:
 - RenderTexture Material List is now always available when the camera of MMB uses a Target Texture.
 - AntiAliasing mode has now the default value 'one'.

Version 1.0 - 2014.12.22

- Initial release

Chapter 7

Support

If you need support or have any question/suggestions please contact us.

- Website: <http://www.monoflow.org/unity3d-assets/monoflowss-metablobs/>
- Email: info@monoflow.org
- Unity Forum: <http://forum.unity3d.com/threads/286150>

Copyright © 2014 by

monoflow
new media experience