StarCluster Pack v3.0

Changelog

v1.0

- initial release

v2.0

- Lens effects
- Light attenuation
- VR support
- New textures
- New models

v3.0

new features

- stars twinkle effect
- procedural glow
- per particle orientation
- delete part of the stars

fixed

- bug with camera hdr mode
- sworling effect on big model scales
- partition models

Features overview

This model pack contains different star cluster models. This little guide will help you to understand how to use it. In this pack you can find 8 different types of star clusters:

- Crab
- Field
- Geometry
- Groups
- Partitions
- Regular
- Spherical
- Wires
- Twist

Each of these types has its own variations.
All these models should be used with StarCluster shader which ships with this pack.

Shader parameters

Shape texture

This texture controls the shape of each sprite.
You can use any texture from “StarClusterPack/Textures/Shapes”

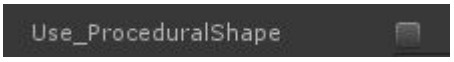
Color texture

This texture controls the color distribution across the sprites.
You can use any texture from “StarClusterPack/Textures/Color”

Size texture

This texture controls the sprites scale distribution across the model.
You can use any texture from “StarClusterPack/Textures/Size”. Inactive if *Use_ProceduralShape* enabled.

Use_ProceduralShape (new!)



Enables procedural glow calculation for stars shape. *Size* texture parameter becomes inactive. Parameters *ProceduralShape_Iterations*, *ProceduralShape_IterExp, ProceduralShape_OutExp* becomes active.

ProceduralShape_Iterations (new!)

Inactive by default. Becomes active if *Use_ProceduralShape* parameter checked. Controls the amount of iterations needed for shader to draw procedural glow.

ProceduralShape_IterExp (new!)

Inactive by default. Becomes active if *Use_ProceduralShape* parameter checked. Controls the exponent applied to glow on each draw iteration.

ProceduralShape_OutExp (new!)

Inactive by default. Becomes active if *Use_ProceduralShape* parameter checked. Controls the exponent applied to glow after all iterations completed.

Color_Tint

Color of all sprites will be multiplied by this color.

Color_Multiplier

Color of all sprites will be multiplied by this value.

Size_Multiplier

Size of all sprites will be multiplied by this value.

Variation_Shift

This parameter allows you to shift Size,Color,Shape textures in UV U direction.
This allows you to make more variations with these assets.

Use_Camera_Position

Enables custom look at mode for sprites. Becomes inactive if *PerParticleOrientation* parameter is not checked.

Camera_Position

Inactive by default. Becomes active if *Use_Camera_Position* parameter checked. This parameter will specify the position of point in space which all sprites will look at. Becomes inactive if *PerParticleOrientation* parameter is not checked.

Use_Attenuation

Enables decay mode for sprites. *Attenuation_Strength, Attenuation_Exponent* parameters becomes active.

Attenuation_Strength

Inactive by default. Becomes active if *Use_Attenuation* parameter checked. This parameter will specify the stars decay strength relative to the distance from the camera.

Attenuation_Exponent

Inactive by default. Becomes active if *Use_Attenuation* parameter checked. This parameter will specify the stars decay exponent.

Use_LensEffect

Enables LensEffect mode for sprites. *LensEffect_Distance, LensEffect_Distance, LensEffect_DistanceExponent* parameters becomes active.

LensEffect

Inactive by default. Becomes active if *Use_LensEffect* parameter checked.
This texture controls the lens effect shape of each sprite.
You can use any texture from “StarClusterPack/Textures/LensEffects”

LensEffect_Distance

Inactive by default. Becomes active if *Use_LensEffect* parameter checked.
This parameter will specify the minimum distance to the star for lens effect to appear.

LensEffect_DistanceExponent

Inactive by default. Becomes active if *Use_LensEffect* parameter checked.
This parameter will specify the size exponent of lens effect relative to the distance from the camera.

Use_Twinkle (new!)

Enables twinkle mode for sprites. *Twinkle_Ramp, Twinkle_Speed, Twinkle_Strength* parameters becomes active.

Twinkle_Ramp (new!)

Inactive by default. Becomes active if *Use_Twinkle* parameter checked.
This texture controls the type of stars twinkle effect.

Twinkle_Speed (new!)

Inactive by default. Becomes active if *Use_Twinkle* parameter checked.
This parameter controls twinkle speed.

Twinkle_Strength (new!)

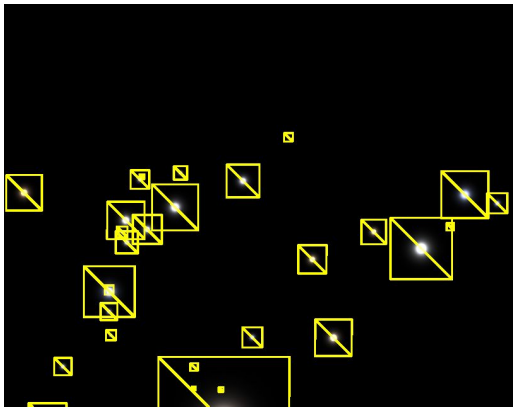
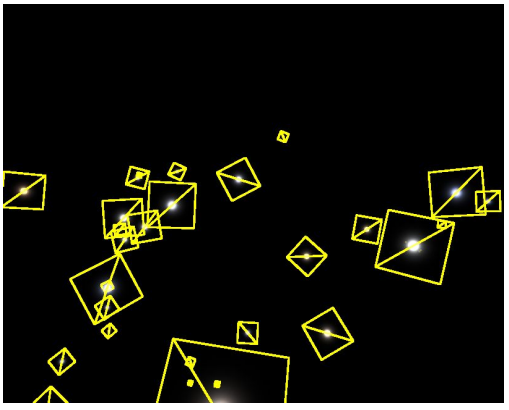
Inactive by default. Becomes active if *Use_Twinkle* parameter checked.
This parameter controls twinkle strength.

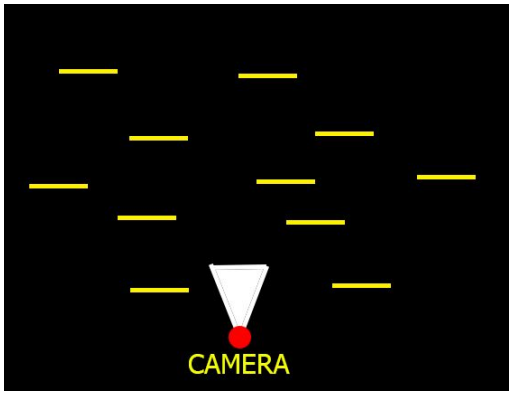
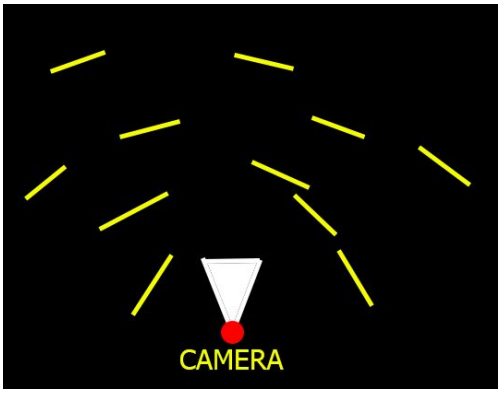
Percentage (new!)

This parameter controls the amount of stars visible. If it equals 1 then all stars is visible.
If it equals 0 then all stars is invisible. If it equals 0.5 then 50% of stars is visible.

PerParticleOrientation (new!)

Determines one of two algorithms for the orientation of sprites.

Disabled	Enabled
	

	
Particles are oriented in screen space. Better for VR. Better lens effects orientation. But if any particle will fly through a camera clipping plane then it will disappear.	Particles are oriented in world space using (Y+) as up vector. **Better for most cases**, because you never fly through a particle. And you can create some custom camera rigs. For example if you want to stitch 6 cameras for creating panoramic image. Enabling this function + using custom camera position controls you will avoid artifacts on the seams between cameras.

Using StarCluster Pack with VR

You should turn off *PerParticleOrientation* parameter. All particles will be oriented in screen space of each eye.

Using StarCluster Pack with custom spherical cameras

You should turn on *PerParticleOrientation* parameter. Enable *Use_Camera_Position* parameter and in *Camera_Position* parameter you need to input the position of camera rig center. That way particles will be properly oriented in world space around your rig.

Creating custom textures for StarCluster shader

This section describes what the color channels of each texture correspond to.

Shape texture
red: shape of the star
green: affects color. If you want very bright center, you should draw it here. It will be multiplied by 10 and will be added to main star shape. Helps to fake HDR effect.
blue: sprite size compensation value.
alpha: sprite intensity compensation value.
Shape intensity = (R+(G*10)+(A*5))

Color texture
red/green/blue: Color values for stars to sample.

Size texture
red: Sprite scale values for stars to sample.
green: not used
blue: not used

LensEffect texture
red: shape of the lens effect
green: affects color. Will be added to red channel.
blue: affects color. If you want very bright center, you should draw it here. It will be multiplied by 10 and will be added to main lens effect shape. Helps to fake HDR effect.
LensEffect intensity = (R+G+(B*10))

Twinkle_Ramp texture
red: Twinkle values for stars to sample. Sprite scale will be multiplied by twinkle value.
green: not used
blue: not used