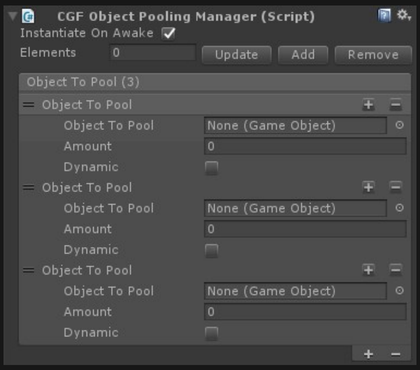
SYSTEMS

OBJECT TRANSFORM

OBJECT POOLING MANAGER

Files: CGFObjectPoolingManager.cs
Path: "CGF/Systems/ObjectTransform"

Manager that allows to the associated gameobject to manage the object pooling.



CGF Object Pooling Manager (Script)
Instantiate On Awake ☑
Elements 0

Object To Pool (3)
- Object To Pool
  - Object To Pool: None (Game Object)
  - Amount: 0
  - Dynamic: □
- Object To Pool
  - Object To Pool: None (Game Object)
  - Amount: 0
  - Dynamic: □
- Object To Pool
  - Object To Pool: None (Game Object)
  - Amount: 0
  - Dynamic: □

Instantiate On Awake – Instance all objects on Awake.

Object To Pool – Pooleable objects list

- **Object To Pool** – Pooleable object.
- **Amount** – Total amount of instances.
- **Dynamic** – Allows to exceed the total amount of instances if needed.

Use

Add the CGFObjectPoolingManager to an empty gameobject in the scene and add all objects to pool to the poolable object list and configure their settings.

Call the method InstantiatePoolObject to instance a poolable object from the poolable object list.

```
CGFObjectPoolingManager.Instance.InstantiatePoolObject(elementToSpawn.prefab, spawnPosition, Quaternion.identity) as GameObject;
```

In case the object must be destroyed, it must be disabled to return to the pool.

```
gameObject.setActive(false);
```

Call the method AddObject to add more objects to the pool in run time.

```
AddObject(GameObject objectToAdd, int amount, bool dynamic)
```

SHADERS

UNLIT

UNLIT PROJECTOR BEHAVIOR

Files: CGFUnlitProjectorBehavior.cs
Path: "CGF/Systems/Shaders/Unlit"

Behavior that allows to the attatched gameobject manage a material with with shader Unlit/Projector.

It has the same properties as the material.

Use

Add the CGFUnlitProjectorBehavior to the gameobject with Projector component configure their settings.

EDITOR

EDITOR UTILITIES

Files: CGFEditorUtilities.cs
Path: "CGF/Editor"

Editor Utilities is a set of methods and utilities to build inspector scripts fast and easily.

Compatible property types:

- Int
    - Generic
    - Positive
    - Negative
    - Slider
- Float
    - Generic
    - Positive
    - Negative
    - Ceil
    - Floor
    - Rounded
    - Slider
    - Slider Ranged
- Long
    - Generic
    - Positive
    - Negative
- Double
    - Generic
    - Positive
    - Negative
    - Ceil
    - Floor
    - Rounded
- Boolean
- Vector 2
    - Generic
    - Positive
    - Negative
- Vector 3
    - Generic
    - Positive
    - Negative
- Vector 4
    - Generic
    - Positive
    - Negative
- Color
- Enumeration
- Rect
- Text
- Animation curve
- Object <T>
- Fold Out
- List
    - Values
    - Classes
- Tags (From Tags and Layers settings)
- Layers (From Tags and Layers settings)
- Sorting Layers (From Tags and Layers settings)
- Inputs (From Input manager settings)
- Layer Mask

To create a custom inspector you should create an editor script with this attributes:

- [CustomEditor(typeof(SampleScript))] – Assign the type of the class that the editor script will represent in the inspector, in this case: *SampleScript.cs*.
- [CanEditMultipleObjects] – Allow support multi-object editing.
- Must inherit from "UnityEditor.Editor".

Step 1

Create "SerializedProperty" variables in the editor script (*SampleScriptEditor*) that reference the variables of the target Class.

```
[CustomEditor(typeof(SampleScript))]
[CanEditMultipleObjects]
public class SampleScriptEditor : UnityEditor.Editor
{
    private SerializedProperty _int;

}
```

Step 2

Inside "OnEnable()" method to initialize the variables of "SerializedProperty" type. Adding as a parameter of "FindProperty(string)" method the name of the referenced variable in the target script.

```
[CustomEditor(typeof(SampleScript))]
[CanEditMultipleObjects]
public class SampleScriptEditor : UnityEditor.Editor
{
    private SerializedProperty _int;

    void OnEnable()
    {
         _int = serializedObject.FindProperty("integer");
    }
}
```

In this case, the variable “integer” from *SampleScript* it’s being linked to the “_int” variable.

Step 3

The method “OnInspectorGUI()” creates the new inspector, allows draw and place the properties in inspector panel. This method works like the “Update()” method, and it must be overrided to work correctly.

Accessing to the static class “EditorGUILayout” you can create any field in the inspector. In this case, the method “IntField()” will show our “_int” variable and will set “Integer” as its name.

```
[CustomEditor(typeof(SampleScript))]
[CanEditMultipleObjects]
public class SampleScriptEditor : UnityEditor.Editor
{
    private SerializedProperty _int;

    void OnEnable()
    {
         _int = serializedObject.FindProperty("integer");
    }

    public override void OnInspectorGUI()
    {
        _int = EditorGUILayout.IntField("Integer", _int);
    }
}
```

In the Editor Utilities you use the “CGFEditorUtilities.BuildInt()” method instead of “EditorGUILayout.IntField()” method. You can add a description as a method parameter to show a tooltip from inspector.

```
[CustomEditor(typeof(SampleScript))]
[CanEditMultipleObjects]
public class SampleScriptEditor : UnityEditor.Editor
{
    private SerializedProperty _int;

    void OnEnable()
    {
         _int = serializedObject.FindProperty("integer");
    }

    public override void OnInspectorGUI()
    {
        CGFEditorUtilities.BuildInt("Integer", "All integer values", _int);
    }
}
```

Step 4

In order to update the inspector fields and values you use the “Update()” method, and to save changes to the variables we must use “ApplyModifiedProperties()”.

```
[CustomEditor(typeof(SampleScript))]
[CanEditMultipleObjects]
public class SampleScriptEditor : UnityEditor.Editor
{
    private SerializedProperty _int;

    void OnEnable()
    {
         _int = serializedObject.FindProperty("integer");
    }

    public override void OnInspectorGUI()
    {
        serializedObject.Update();
        CGFEditorUtilities.BuildInt("Integer", "All integer values", _int);

        serializedObject.ApplyModifiedProperties();
    }
}
```

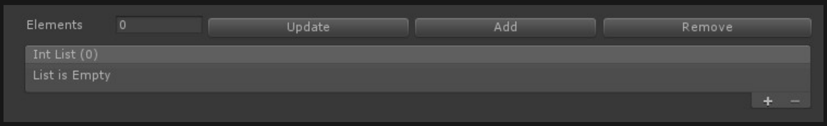
REORDERABLE LISTS

Unity manages list information as arrays which is uncomfortable for user. To improve usability, lists have been created that allows sort list elements just dragging them and additional features have been added.

The method “BuildListCustom()” allows you to create reorderable lists of classes. With the “params string[] properties” parameter, assign all the class properties for each list element.

```
[CustomEditor(typeof(SampleScript))]
[CanEditMultipleObjects]
public class SampleScriptEditor : UnityEditor.Editor
{
    private SerializedProperty _listProperty;

    private ReorderableList _reorderableList;

    private int _newListSize;

    void OnEnable()
    {
         _listProperty = serializedObject.FindProperty("list");

         _reorderableList = new ReorderableList(serializedObject, _listProperty, true, true, true, true);
    }

    public override void OnInspectorGUI()
    {
        serializedObject.Update();

        _newListSize = CGFEditorUtilities.BuildListButtons(_listProperty, _newListSize);

        _reorderableList = CGFEditorUtilities.BuildListCustom(_listProperty, _reorderableList, "List Name", bool vertical
, int[] propertySpace, params string[] properties);

        serializedObject.ApplyModifiedProperties();
    }
}
```

To make easier the add and remove elements actions from the list you have two buttons. The “BuildListButtons()” method adds these buttons.



Elements 0

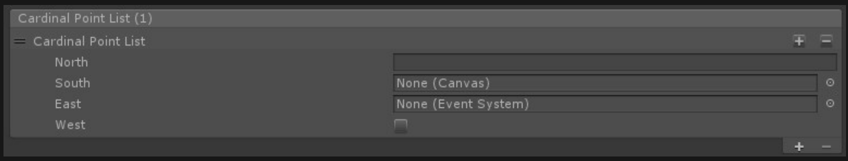
Int List (0)

List is Empty

An additional button has been created that replaces the total number of elements in the list with a new total. This action is activated by matching the *int* value of the new quantity of elements to the “BuildListButtons()” method.

There are two reorderable list layouts, vertical (default) and horizontal. “bool vertical” parameter determine the layout to use.

In the vertical layout element fields appear one below the other.



Cardinal Point List (1)

Cardinal Point List

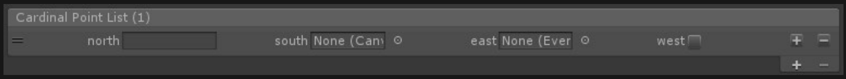
North

South None (Canvas)

East None (Event System)

West

In the horizontal layout element fields of elements appear next to each other.



Cardinal Point List (1)

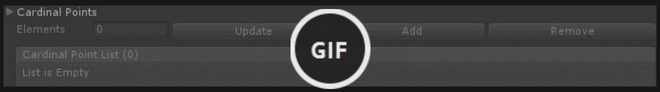
north south None (Canv east None (Ever west

You can specify width of each element field with “int[] propertySpace” parameter. You enter a value between 1 and 12, the total value of all the array elements must be 12.

FOLD OUT

Boolean value that manages if the objects inside the FoldOut must be shown or not.

As the FoldOut boolean value in the inspector is selected, it changes its value to show or not the objects that contains.

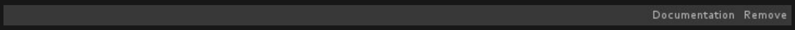


Use

```
[CustomEditor(typeof(SampleScript))]
[CanEditMultipleObjects]
public class SampleScriptEditor : UnityEditor.Editor
{
    private bool _foldOut;

    private SerializedProperty _vector3;

    public void OnEnable()
    {
         _vector3 = serializedObject.FindProperty("vector3");
    }

    public override void OnInspectorGUI()
    {
        serializedObject.Update();

        _foldOut = CGFEditorUtilities.BuildFoldOut("Cube5", "Description", _foldOut);

        if(_foldOut)
        {
               CGFEditorUtilities.BuildVector3("Position", "Description", _vector3);
        }

        serializedObject.ApplyModifiedProperties();
    }
}
```

COMPONENT TOOLS

Adds shortcuts to actions related to the component.



Documentation – Shortcut to component documentation.

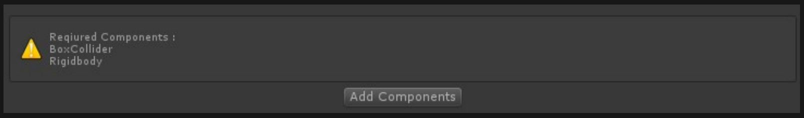
Remove – Shortcut to delete the component.

Use

```
CGFEditorUtilities.BuildComponentTools(string documentationURL, SerializedObject component)
```

REQUIRED COMPONENTS

Warn what components are required gameobject and allows them to be added easily.



Required Components :
BoxCollider
Rigidbody

Add Components – Add the required components that are not present in the current gameobject.

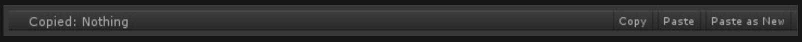
Use

```
CGFEditorUtilities.BuildRequiredComponents(params Type[] components
```

```
CGFEditorUtilities.BuildRequiredComponents(typeof(BoxCollider), typeof(Rigidbody));
```

MANAGE COMPONENT VALUES

Copy, paste or paste as new component the values from a component.


Copied: Nothing

Copied – Shows if copied values exists.

Copy – Copy the component values.

Paste – Paste the component values on the current component.

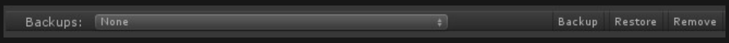
Paste as New – Add a new component to the gameobject with the copied values.

Use

```
CGFEditorUtilities.ManageComponentValues<T>()
```

BACK UP COMPONENT VALUES

Allow create component back ups and restore this back ups saved in the project. Works in “Play” and “Edit” mode.


Backups: None

Backups – Dropdown menu with all available backups.

Backup – Create a back up of the component.

Restore – Replace the current component values for the values saved in the selected back up from dropdown menu.

Remove – Delete the selected backup from the dropdown menu.

Use

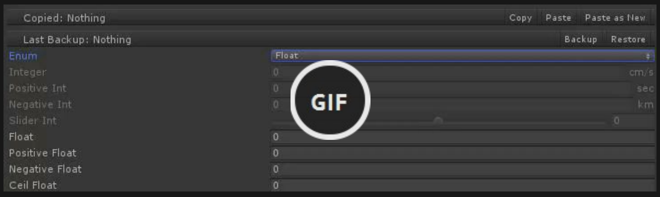
```
CGFEditorUtilities.BackUpComponentValues<SampleScript>(serializedObject)
```

The parameter of this method is the type *<T>* of the target scrip to back up and the “SerializedObject”.

LOCKING PROPERTIES

ENUMERATION

Allows enable or disable fields from inspector according to a selected enumeration value.


Copied: Nothing

Last Backup: Nothing

Enum Float

Integer 0 cm/s

Positive Int 0 sec

Negative Int 0 km

Slider Int 0

Float 0

Positive Float 0

Negative Float 0

Ceil Float 0

Every property construction method has an overload with two additional parameters to add the enumeration and the enumeration index of values that enable or disable the field.

Use

In this example each filed method receives an "_enum" variable as an enumeration parameter and a *int* value as other parameter that designates wich values from "_enum" enables the field.

```
public override void OnInspectorGUI()
{
     CGFEditorUtilities.BuildEnum("Enum", "Enumaration of different values", _enum);


     CGFEditorUtilities.BuildInt("Integer", "All integer values", _int, "cm/s", _enum, 1);

     CGFEditorUtilities.BuildIntPositive("Positive Int", "Only positive integer values", _positiveInt, "sec", _enum, 1);


     CGFEditorUtilities.BuildFloat("Float", "All float values", _float, _enum, 2);

     CGFEditorUtilities.BuildFloatPositive("Positive Float", "Only positive float values", _positiveFloat, _enum, 2);
}
```

BOOLEAN

Allows enable or disable fields from the inspector according to a selected boolean property.

Every property construction method has an overload with an additional parameter to add a boolean that will lock/unlock the property in the inspector.

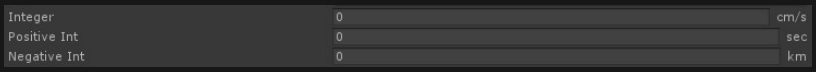
Use

In this example the property "_bool" will be the property locker, as we see in the code, every method adds as a parameter the the property "bool".

```
public override void OnInspectorGUI()
{
     CGFEditorUtilities.BuildBool("Boolean", "Locker Boolean", _bool);


     CGFEditorUtilities.BuildInt("Integer", "All integer values", _int, "cm/s", _bool);

     CGFEditorUtilities.BuildIntPositive("Positive Int", "Only positive integer values", _positiveInt, "sec", _bool);


     CGFEditorUtilities.BuildFloat("Float", "All float values", _float, _bool);

     CGFEditorUtilities.BuildFloatPositive("Positive Float", "Only positive float values", _positiveFloat, _bool);
}
```

VALUE UNITS

A unit can be added to the filds from inspector.



Integer	0	cm/s
Positive Int	0	sec
Negative Int	0	km

Every field construction method has an overload with an string parameter which will set the units.

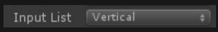
Use

In this example you can see that the last parameter of each method is the units of each field.

```
public override void OnInspectorGUI()
{
     CGFEditorUtilities.BuildInt("Integer", "All integer values", _int, "cm/s");

     CGFEditorUtilities.BuildIntPositive("Positive Int", "Only positive integer values", _positiveInt, "sec");

     CGFEditorUtilities.BuildIntNegative("Negative Int", "Only negative integer values", _negativeInt, "km");
}
public class SampleScriptEditor : UnityEditor.Editor
{
private bool _foldOut;

private SerializedProperty _vector3;

public void OnEnable()
{
_vector3 = serializedObject.FindProperty("vector3");
}

public override void OnInspectorGUI()
{
serializedObject.Update();

_foldOut = CGFEditorUtilities.BuildContentFoldOut("Cube5", "Description", _foldOut);

if(_foldOut)
{
CGFEditorUtilities.BuildVector3("Position", "Description", _vector3);
}

serializedObject.ApplyModifiedProperties();
}
}
```

INPUT LIST

Allows to show the inputs from Input Manager pof Unity with a popup.


Input List Vertical

Use

The method “BuildInputList()” allows to get a enumeration (Popup) with all the inputs that manages Unity, to call this method you should use this method before “ReadInputs()”.

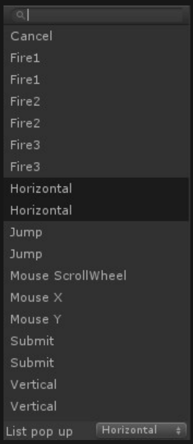
In this example you can see the use.

```
public void OnEnable()
{

    CGFEditorUtilitiesClass.ReadInputs();

}

public void OnInspectorGUI()
{

    CGFEditorUtilitiesClass.BuildInputList(enumerationName, serializedProperty);

}
```

Is strongly recommended don’t use the “ReadInputs()” method on recursively methods such “Update()”, affects vegatively in performance, it is recommended to use it in “Awake()” or in the initialization of the editor class that is being used.

ENUM LIST POP UP WINDOW

Allows to search fields in a list from inspector with a popup.



List pop up Horizontal

Use

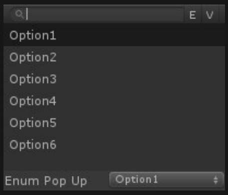
The method “BuildEnumListPopUpWindow()” allows to build a popup with a search field, this field will filter the elemements of string list and assigns the selected element to the property.

In this example you can see the use.

```
public override void OnInspectorGUI()
{

   CGFEditorUtilitiesClass.BuildEnumPopUpWindow("Label Name", "Description", rectPosition, serializedProperty, EnumType);

}
```

ENUM POP UP WINDOW

Allows to search fields in a enumeration from inspector with a popup.



- **E/A –** Sorting Method. Enumeration o Alphabetically.
- **V –** Shows the index of the enumeration element.

Use

The method “BuildEnumPopUpWindow()” allows to build a popup with a search field, this field will filter the elemements of a enumeration and assigns the selected element to the property.

In this example you can see the use.

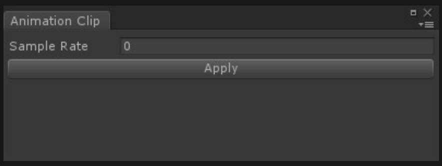
```
public override void OnInspectorGUI()
{

    CGFEditorUtilitiesClass.BuildEnumListPopUpWindow("Label Name", "Description", rectPosition, serializedProperty, listNames);

}
```

TOOLS

ANIMATION CLIP SETTINGS

Files: CGFAnimationClipSettingsTool.cs
Path: “CGF/Editor/Tools”
Menu: “Window/Chloroplast Games Framework/Animation Clip Settings Tool”

Tool that allows change the sample rate from one or multiple animation clips.



Sample Rate – New sample rate.

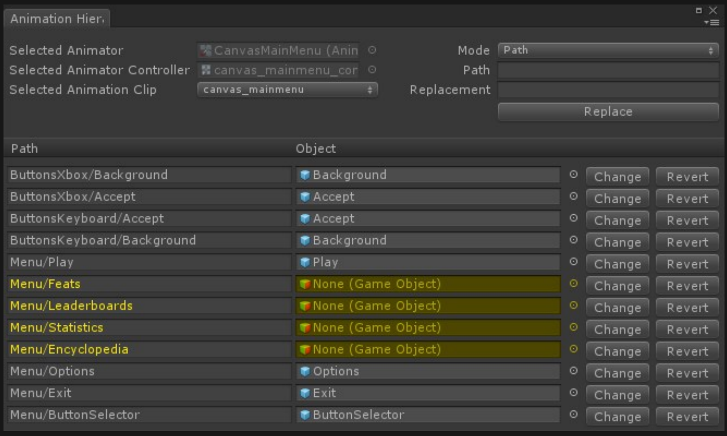
Use

Select a Animation Clip or multiple Animation Clips to change their settings.

ANIMATION HIEARCHY EDITOR

Files: CGFAnimationHierarchyEditorTool.cs
Path: “CGF/Editor/Tools”
Menu: “Window/Chloroplast Games Framework/Animation Hierarchy Editor Tool”

Tool that allows change the path of hierarchy from a animation from an animation clip.



Selected Animator: CanvasMainMenu (Anin
Selected Animator Controller: canvas_mainmenu_cor
Selected Animation Clip: canvas_mainmenu
Mode: Path
Path:
Replacement:

Path	Object		
ButtonsXbox/Background	Background	Change	Revert
ButtonsXbox/Accept	Accept	Change	Revert
ButtonsKeyboard/Accept	Accept	Change	Revert
ButtonsKeyboard/Background	Background	Change	Revert
Menu/Play	Play	Change	Revert
Menu/Feats	None (Game Object)	Change	Revert
Menu/Leaderboards	None (Game Object)	Change	Revert
Menu/Statistics	None (Game Object)	Change	Revert
Menu/Encyclopedia	None (Game Object)	Change	Revert
Menu/Options	Options	Change	Revert
Menu/Exit	Exit	Change	Revert
Menu/ButtonSelector	ButtonSelector	Change	Revert

Selected Animator – Selected Animator.

Selected Animator Controller – Selected Animator Controller.

Selected Animation Clip – Selected Animation Clip or Animation clips of the selected gameobject or Animator Controller.

Mode – Bulk replace mode.

- **Path –** Text replacement mode.
- **GameObject –** Gameobject replacement mode.

Path/GameObject – Original text or gameobject.

Replacement – Text or gameobject replacement.

Path – Hierarchy path of the animated gameobject.

Object – Animated gameobject or prefab.

Use

Select a gameobject or prefab with a Animator or Animator Controller, a Animator Controller or a Animation Clip.

Then you can change the hierarchy path of the animation changing the string (Path) and press the Change button or changing the animated gameobject or prefab (Object). To undo the change press the Reverse button.

If the selected gameobject or Animator Controller have multiple Animation Clips you can switch the current Animation Clip using the Selected Animation Clip drop down menu.

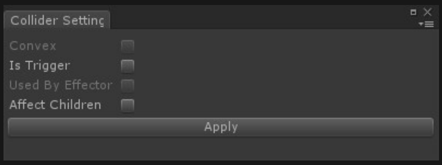
When some property becomes yellow mens that the hierarchy path is incorrect because the referenced animated gameobject is missing.

You can replace the hierarchy path of multiple animatios with the bulk replacement system.

COLLIDER SETTINGS

Files: CGFColliderSettingsTool.cs
Path: "CGF/Editor/Tools"
Menu: "Window/Chloroplast Games Framework/Collider Settings Tool"

Tool that allows change the settings of a all colliders from a GameObject or prefab.



Convex □
Is Trigger □
Used By Effector □
Affect Children □

Convex – Enable or disable the Convex property. Only for MeshCollider.

Is Trigger – Enable or disable the Is Trigger property.

Used By Effector – Enable or disable the Used By Effector property. Only for 2D Collider.

Affect Children – Enable or disable the properties of the colliders of the childs from the selected gameobject.

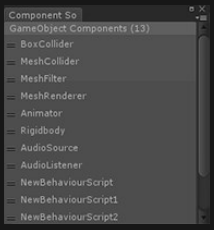
Use

Select a gameobject or prefab or multiple gameobjects and prefabs with a collider or multiple colliders of the same type to to change their collider properties.

COMPONENT SORTER

Files: CGFComponentSorterTool.cs
Path: “CGF/Editor/Tools”
Menu: “Window/Chloroplast Games Framework/Component Sorter Tool”

Tool that allows sort the components from gameobject or prefab.



GameObject Components (13)
- BoxCollider
- MeshCollider
- MeshFilter
- MeshRenderer
- Animator
- Rigidbody
- AudioSource
- AudioListener
- NewBehaviourScript
- NewBehaviourScript1
- NewBehaviourScript2

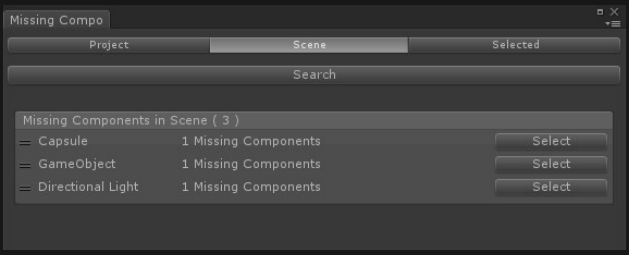
Use

Select a gameobject or prefab to sort (drag and drop) their components.

MISSING COMPONENT FINDER

Files: CGFMissingComponentFinderTool.cs
Path: “CGF/Editor/Tools”
Menu: “Window/Chloroplast Games Framework/Missing Component Finder Tool”

Tool that allows search for missing components of gameobjects from a scene, project or selected gameobjects and folders.



Missing Components in Scene (3)

Capsule	1 Missing Components	Select
GameObject	1 Missing Components	Select
Directional Light	1 Missing Components	Select

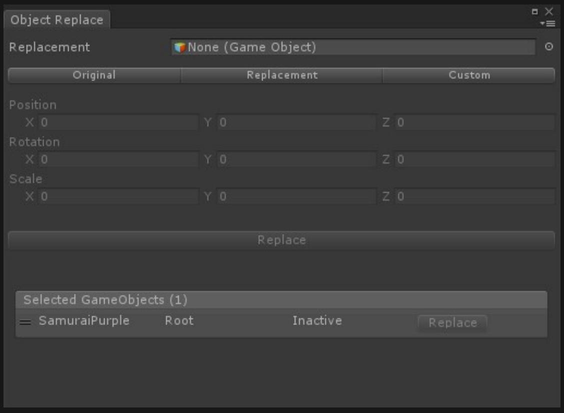
Three ways to search missing components from gameobjects.

- Project – Search missing components from prefabs and childrens inside project folder.
- Scene – Search missing components from gameobjects and childrens from the current scene.
- Selected – Search missing components from selected elements (gameobjects, prefabs or folders).

OBJECT REPLACER

Files: CGFObjectReplacerTool.cs
Path: “CGF/Editor/Tools”
Menu: “Window/Chloroplast Games Framework/Object Replacer Tool”

Tool that allows replace a gameobject on the scene with other gameobject or a prefab.



Replacement None (Game Object)

Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 0

Selected GameObjects (1)

SamuraiPurple	Root	Inactive	Replace

Replacement – Replacement gameobject or prefab that replaces the selected gameobject.

Transform Replacement – Original, keep the transform values from original gameobject or prefab. Replacement, use the transform values from replacement gameobject or prefab. Custom, use the values from Position, Rotation and Scale properties.

Position – New position. Only for Custom mode.

Rotation – New rotatión. Only for Custom mode.

Scale – New scale. Only for Custom mode.

Selected GameObjects – Selected gameobject list.

Use

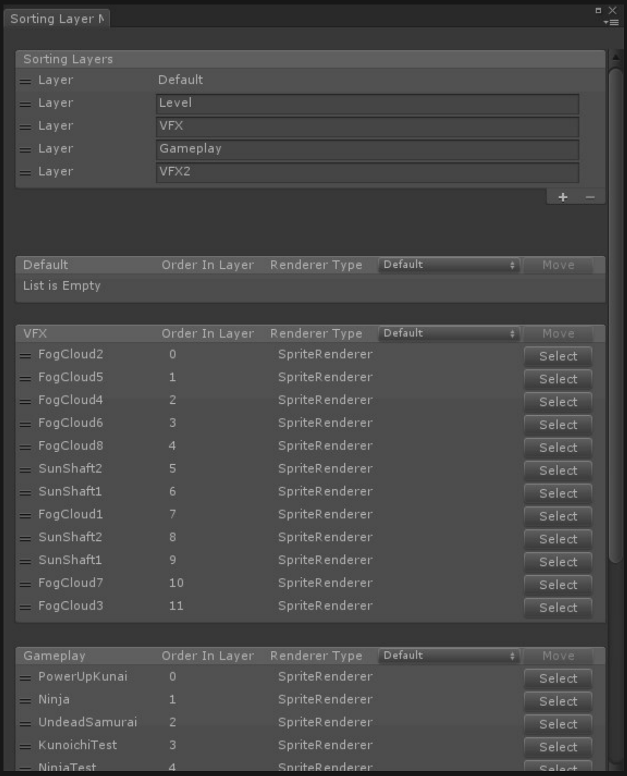
Drag a gameobject or prefaf into the Replacement property, configure the transform mode and press the Replace button.

You can replace each selected gameobject one by one.

SORTING LAYER MANAGER

Files: CGFSortingLayerManagerTool.cs
Path: "CGF/Editor/Tools"
Menu: "Window/Chloroplast Games Framework/Sorting Layer Manager Tool"

Tool that allows sort comfortably the sorting layers and order in layer of the elements from de current scene.



Sorting Layers	
Layer	Default
Layer	Level
Layer	VFX
Layer	Gameplay
Layer	VFX2

Default	Order In Layer	Renderer Type	Default
List is Empty			

VFX	Order In Layer	Renderer Type	Default
FogCloud2	0	SpriteRenderer	Select
FogCloud5	1	SpriteRenderer	Select
FogCloud4	2	SpriteRenderer	Select
FogCloud6	3	SpriteRenderer	Select
FogCloud8	4	SpriteRenderer	Select
SunShaft2	5	SpriteRenderer	Select
SunShaft1	6	SpriteRenderer	Select
FogCloud1	7	SpriteRenderer	Select
SunShaft2	8	SpriteRenderer	Select
SunShaft1	9	SpriteRenderer	Select
FogCloud7	10	SpriteRenderer	Select
FogCloud3	11	SpriteRenderer	Select

Gameplay	Order In Layer	Renderer Type	Default
PowerUpKunai	0	SpriteRenderer	Select
Ninja	1	SpriteRenderer	Select
UndeadSamurai	2	SpriteRenderer	Select
KunoichiTest	3	SpriteRenderer	Select
NinjaTest	4	SpriteRenderer	Select

Use

With the top reorderable list you can sort (drag and drop), create and remove sorting layers.

With the bottom reorderable lists you can sort (drag and drop) the order in layer from each gameobject. Show all types of renderers (SpriteRenderer, ParticleSystem (ParticleRenderer), LineRenderer, TrailRenderer) less the MeshRenderer.

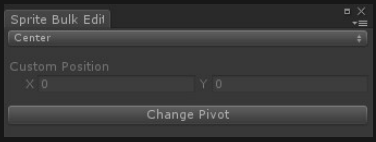
If you remove a sortign layers, all gameobjects from this sorting layer are moved to sorting layer "Default".

⚠ Unity Sorting Layer Manager and Sorting Layer Manager Tool should not be used at the same time because it can cause errors.

SPRITE BULK EDITOR

Files: CGFSpriteBulkEditorTool.cs
Path: "CGF/Editor/Tools"
Menu: "Window/Chloroplast Games Framework/Sprite Bulk Editor Tool"

Tool that allows modify the pivot of selected sprites.


Sprite Bulk Edit
Center
Custom Position
X 0 Y 0
Change Pivot

Dropdown menu – Position presets.

Custom Position – Custom pivot position. Only if the selected option from the dropdown menu is Custom.

Use

Select a sprite or miltiple sprites.

If Sprite mode is set to Multiple, the changes affects all the sprite slices.

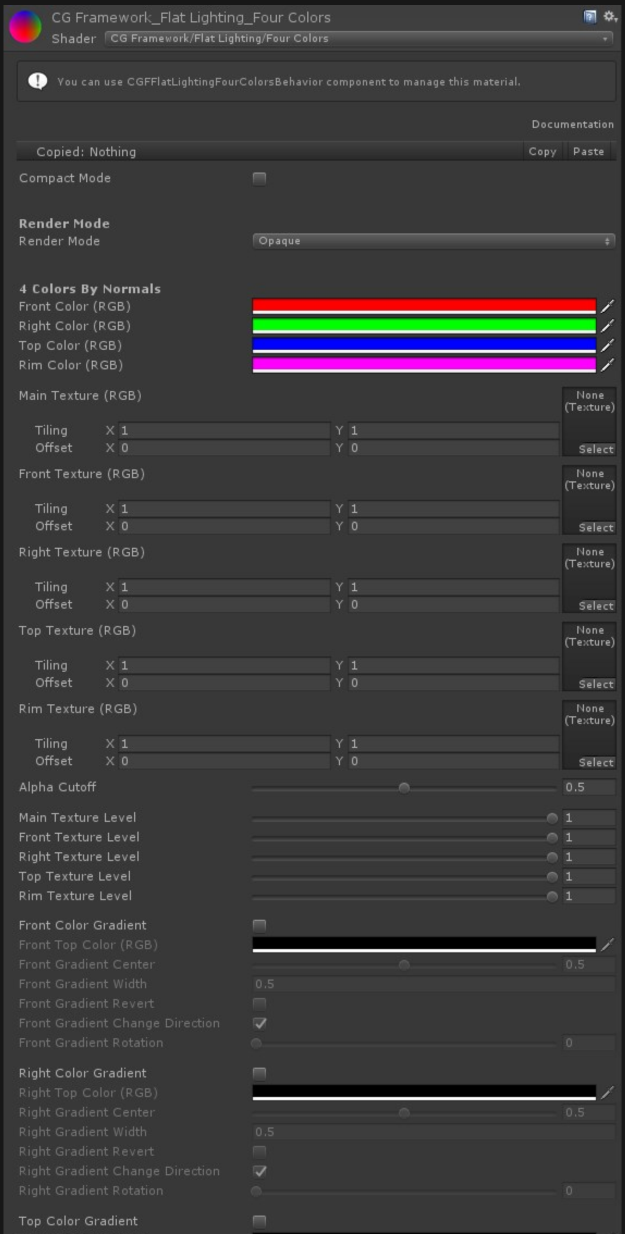
SHADERS

FLAT LIGHTING

FOUR COLORS

Files: FourColors.shader
Path: "CGF/Shaders/FlatLighting"
Shader menu: "CG Framework/Flat Lighting/Four Colors"

Shader that applies four colors based on the normals of the mesh.


CG Framework_Flat Lighting_Four Colors
Shader: CG Framework/Flat Lighting/Four Colors

You can use CGFFlatLightingFourColorsBehavior component to manage this material.

Documentation
Copied: Nothing — Copy | Paste

Compact Mode □

Render Mode
Render Mode: Opaque

4 Colors By Normals
Front Color (RGB) 🟥
Right Color (RGB) 🟩
Top Color (RGB) 🟦
Rim Color (RGB) 🟪

Main Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Front Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Right Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Top Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Rim Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

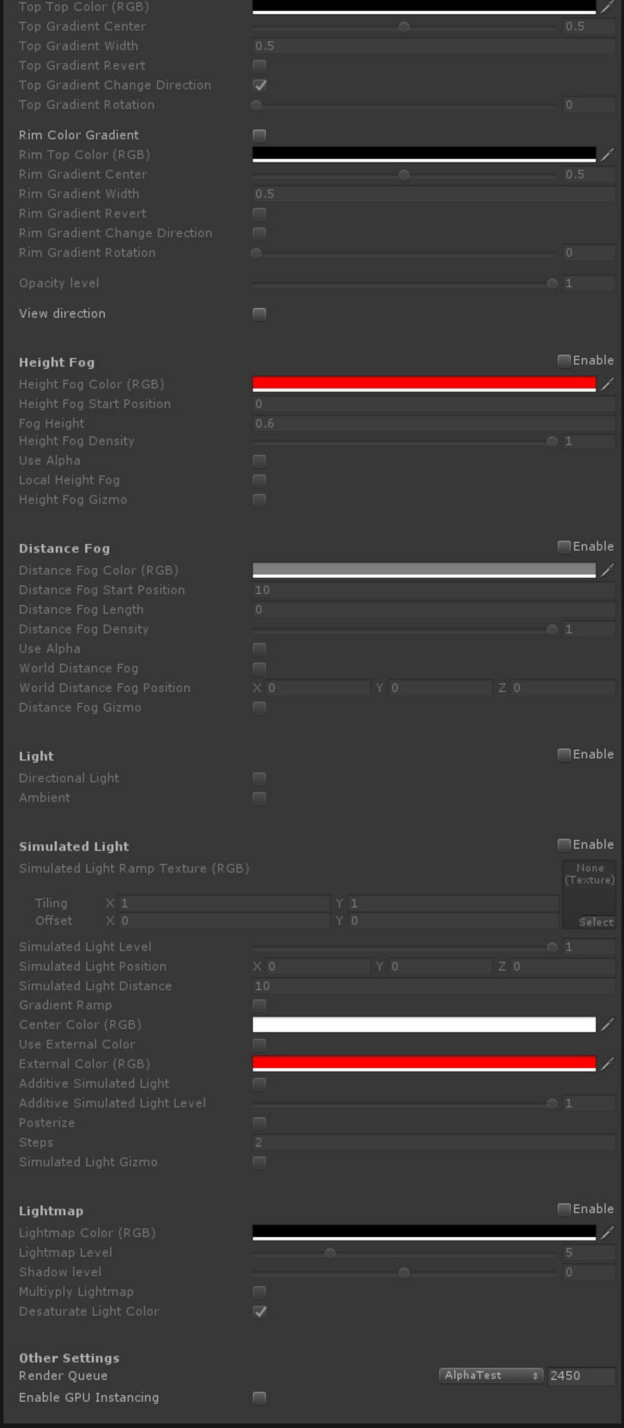
Alpha Cutoff 0.5

Main Texture Level 1
Front Texture Level 1
Right Texture Level 1
Top Texture Level 1
Rim Texture Level 1

Front Color Gradient □
Front Top Color (RGB) ■
Front Gradient Center 0.5
Front Gradient Width 0.5
Front Gradient Revert □
Front Gradient Change Direction ☑
Front Gradient Rotation 0

Right Color Gradient □
Right Top Color (RGB) ■
Right Gradient Center 0.5
Right Gradient Width 0.5
Right Gradient Revert □
Right Gradient Change Direction ☑
Right Gradient Rotation 0

Top Color Gradient □



Top Top Color (RGB)
Top Gradient Center 0.5
Top Gradient Width 0.5
Top Gradient Revert □
Top Gradient Change Direction ☑
Top Gradient Rotation 0

Rim Color Gradient □
Rim Top Color (RGB)
Rim Gradient Center 0.5
Rim Gradient Width 0.5
Rim Gradient Revert □
Rim Gradient Change Direction □
Rim Gradient Rotation 0

Opacity level 1

View direction □

Height Fog □ Enable
Height Fog Color (RGB)
Height Fog Start Position 0
Fog Height 0.6
Height Fog Density 1
Use Alpha □
Local Height Fog □
Height Fog Gizmo □

Distance Fog □ Enable
Distance Fog Color (RGB)
Distance Fog Start Position 10
Distance Fog Length 0
Distance Fog Density 1
Use Alpha □
World Distance Fog □
World Distance Fog Position X 0 Y 0 Z 0
Distance Fog Gizmo □

Light □ Enable
Directional Light □
Ambient □

Simulated Light □ Enable
Simulated Light Ramp Texture (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0
Simulated Light Level 1
Simulated Light Position X 0 Y 0 Z 0
Simulated Light Distance 10
Gradient Ramp □
Center Color (RGB)
Use External Color □
External Color (RGB)
Additive Simulated Light □
Additive Simulated Light Level 1
Posterize □
Steps 2
Simulated Light Gizmo □

Lightmap □ Enable
Lightmap Color (RGB)
Lightmap Level 5
Shadow level 0
Multiyply Lightmap □
Desaturate Light Color ☑

Other Settings
Render Queue AlphaTest 2450
Enable GPU Instancing □

Render Mode – Manages the rendering mode.

- **Opaque –** Solid objects without transparent areas.
- **Transparent –** Partial or total transparent objects.
- **Cutout –** Objects with a hard edge between transparents and opaque areas. The object can not have semi-transparent areas.
- **Background –** Solid objects without transparents areas that are rendered behind any other object.

Front Color – Color of the front normals. Use the RGB or RGBA channels according to the selected rendering mode.

Right Color –Color of the right normals. Use the RGB or RGBA channels according to the selected rendering mode.

Top Color – Color of the top normals. Use the RGB or RGBA channels according to the selected rendering mode.

Rim Color – Color of the rim normals. Use the RGB or RGBA channels according to the selected rendering mode.

Main Texture – Texture of all normals. Use the RGB or RGBA channels according to the selected rendering mode.

Front Texture – Texture of the front normals. Use the RGB or RGBA channels according to the selected rendering mode.

Right Texture – Texture of the right normals. Use the RGB or RGBA channels according to the selected rendering mode.

Top Texture – Texture of the top normals. Use the RGB or RGBA channels according to the selected rendering mode.

Rim Texture – Texture of the rim normals. Use the RGB or RGBA channels according to the selected rendering mode.

Alpha cutoff – Alpha Cutoff value. Enabled if the selected rendering mode is Cutout.

Main Texture Level – Level of main texture in relation the source color.

Front Texture Level – Level of front texture in relation the source color.

Right Texture Level – Level of right texture in relation the source color.

Top Texture Level – Level of top texture in relation the source color.

Rim Texture Level – Level of rim texture in relation the source color.

Front Color Gradient – Front color gradient.

Front Top Color – Color of the top part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Front Gradient Center – Gradient center.

Front Gradient Width – Gradient width.

Front Gradient Revert – Revert the ortientation of the gradient.

Front Gradient Change Direction – Change direction of the gradient.

Front Gradient Rotation – Gradient rotation.

Right Color Gradient – Right color gradient.

Right Top Color – Color of the top part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Right Gradient Center – Gradient center.

Right Gradient Width – Gradient width.

Right Gradient Revert – Revert the ortientation of the gradient.

Right Gradient Change Direction – Change direction of the gradient.

Right Gradient Rotation – Gradient rotation.

Top Color Gradient – Top color gradient.

Top Top Color – Color of the top part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Top Gradient Center – Gradient center.

Top Gradient Width – Gradient width.

Top Gradient Revert – Revert the ortientation of the gradient.

Top Gradient Change Direction – Change direction of the gradient.

Top Gradient Rotation – Gradient rotation.

Rim Color Gradient – Right color gradient.

Rim Top Color – Color of the top part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Rim Gradient Center – Gradient center.

Rim Gradient Width – Gradient width.

Rim Gradient Revert – Revert the ortientation of the gradient.

Rim Gradient Change Direction – Change direction of the gradient.

Rim Gradient Rotation – Gradient rotation.

Opacity – Overall opacity value. Enabled if the selected rendering mode is Transparent or Cutout.

View Mode – If enabled the color is applied based on the view direction.

Height Fog – Fog by vertex height.

Height Fog Color – Color of the fog.

Height Fog Start Position – Start point of the fog.

Fog Height – Height of the fog.

Height Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn’t affect the transparent parts of the source color.

Local Height Fog – If enabled the fog is created based on the center of the mesh.

Height Fog Gizmo – If enabled show height fog gizmo.

Distance Fog – Fog by camera distance.

Distance Fog Color – Color of the fog.

Distance Fog Start Position – Start point of the fog.

Distance Length – Length of the fog.

Distance Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn’t affect the transparent parts of the source color.

World Distance Fog – If enabled the fog is created based on a position of the world.

World Distance Fog Position – Posición en el mundo de la niebla.

Distance Fog Gizmo – World position of the distance fog.

Light – Light and Ambient light.

Directional Light – If enabled main directional light affect to the source mesh.

Ambient – If enabled ambient light affect to the source mesh.

Simulated Light – Simulated Light.

Simulated Light Ramp Texture – Color ramp of the simulated light based on a texture. The top part of the texture is the center of the simulated light and the bottom part is the external part of the simulated light.

Simulated Light Level – Level of simulated light in relation the source color.

Simulated Light Position – World position of the simulated light.

Simulated Light Distance – Simulated light circunference diameter.

Gradient Ramp – If enabled uses a gradient ramp between two colors instead a ramp texture.

Center Color – Color of the center of the simulated light if gradient ramp is enabled.

Use External Color – If enabled uses a color for the external part of the light instead de source color.

External Color – Color of the expernal part of the simulated light if gradient ramp is enabled.

Additive Simulated Light – If enabled adds the simulated light color to the source color.

Additive Simulated Light Level – Level of simulated light addition in relation the source color.

Posterize – If enabled converts the ramp texture or the gradient ramp to multiple regions of fewer tones.

Steps – Color steps of the posterization.

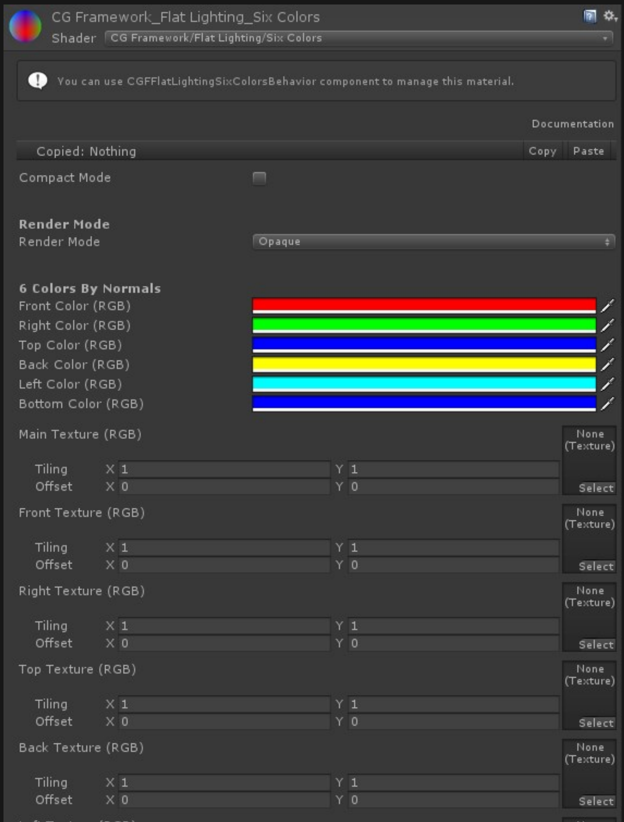
Simulated Light Gizmo – If enabled show simulated light gizmo.

Use

Create a material with this shader and assign the material to the Material property of the MeshRenderer component.

Files: SixColors.shader
Path: "CGF/Shaders/FlatLighting"
Shader menu: "CG Framework/Flat Lighting/Six Colors"

Shader that applies six colors based on the normals of the mesh.



CG Framework_Flat Lighting_Six Colors
Shader CG Framework/Flat Lighting/Six Colors

You can use CGFFlatLightingSixColorsBehavior component to manage this material.

Documentation

Copied: Nothing Copy Paste

Compact Mode □

Render Mode
Render Mode Opaque

6 Colors By Normals
Front Color (RGB) 🟥
Right Color (RGB) 🟩
Top Color (RGB) 🟦
Back Color (RGB) 🟨
Left Color (RGB) 🟦
Bottom Color (RGB) 🟦

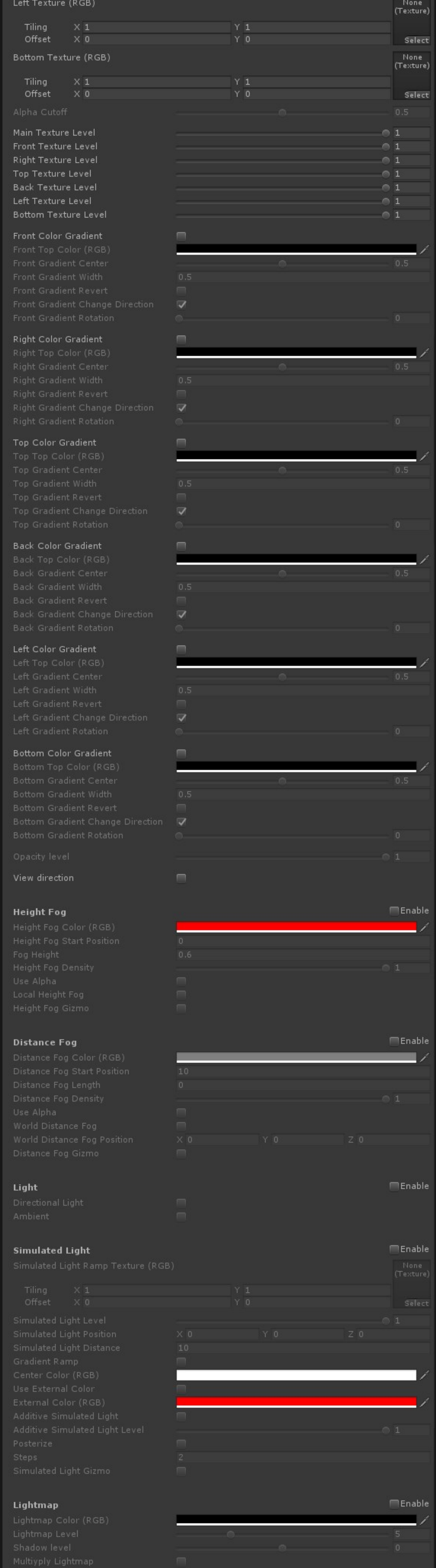
Main Texture (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Front Texture (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Right Texture (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Top Texture (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Back Texture (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0



Left Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Bottom Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Alpha Cutoff 0.5

Main Texture Level 1
Front Texture Level 1
Right Texture Level 1
Top Texture Level 1
Back Texture Level 1
Left Texture Level 1
Bottom Texture Level 1

Front Color Gradient □
Front Top Color (RGB)
Front Gradient Center 0.5
Front Gradient Width 0.5
Front Gradient Revert □
Front Gradient Change Direction ☑
Front Gradient Rotation 0

Right Color Gradient □
Right Top Color (RGB)
Right Gradient Center 0.5
Right Gradient Width 0.5
Right Gradient Revert □
Right Gradient Change Direction ☑
Right Gradient Rotation 0

Top Color Gradient □
Top Top Color (RGB)
Top Gradient Center 0.5
Top Gradient Width 0.5
Top Gradient Revert □
Top Gradient Change Direction ☑
Top Gradient Rotation 0

Back Color Gradient □
Back Top Color (RGB)
Back Gradient Center 0.5
Back Gradient Width 0.5
Back Gradient Revert □
Back Gradient Change Direction ☑
Back Gradient Rotation 0

Left Color Gradient □
Left Top Color (RGB)
Left Gradient Center 0.5
Left Gradient Width 0.5
Left Gradient Revert □
Left Gradient Change Direction ☑
Left Gradient Rotation 0

Bottom Color Gradient □
Bottom Top Color (RGB)
Bottom Gradient Center 0.5
Bottom Gradient Width 0.5
Bottom Gradient Revert □
Bottom Gradient Change Direction ☑
Bottom Gradient Rotation 0

Opacity level 1

View direction □

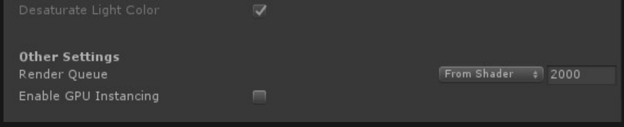
Height Fog □ Enable
Height Fog Color (RGB) 🟥
Height Fog Start Position 0
Fog Height 0.6
Height Fog Density 1
Use Alpha □
Local Height Fog □
Height Fog Gizmo □

Distance Fog □ Enable
Distance Fog Color (RGB)
Distance Fog Start Position 10
Distance Fog Length 0
Distance Fog Density 1
Use Alpha □
World Distance Fog □
World Distance Fog Position X 0 Y 0 Z 0
Distance Fog Gizmo □

Light □ Enable
Directional Light □
Ambient □

Simulated Light □ Enable
Simulated Light Ramp Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0
Simulated Light Level 1
Simulated Light Position X 0 Y 0 Z 0
Simulated Light Distance 10
Gradient Ramp □
Center Color (RGB)
Use External Color □
External Color (RGB) 🟥
Additive Simulated Light □
Additive Simulated Light Level 1
Posterize □
Steps 2
Simulated Light Gizmo □

Lightmap □ Enable
Lightmap Color (RGB)
Lightmap Level 5
Shadow level 0
Multiyply Lightmap □



Desaturate Light Color ☑

Other Settings

Render Queue From Shader 2000

Enable GPU Instancing □

Render Mode – Manages the rendering mode.

- **Opaque –** Solid objects without transparent areas.
- **Transparent –** Partial or total transparent objects.
- **Cutout –** Objects with a hard edge between transparents and opaque areas. The object can not have semi-transparent areas.
- **Background –** Solid objects without transparents areas that are rendered behind any other object.

Front Color – Color of the front normals. Use the RGB or RGBA channels according to the selected rendering mode.

Right Color – Color of the right normals. Use the RGB or RGBA channels according to the selected rendering mode.

Top Color – Color of the top normals. Use the RGB or RGBA channels according to the selected rendering mode.

Back Color – Color of the back normals. Use the RGB or RGBA channels according to the selected rendering mode.

Left Color – Color of the left normals. Use the RGB or RGBA channels according to the selected rendering mode.

Bottom Color – Color of the bottom normals. Use the RGB or RGBA channels according to the selected rendering mode.

Main Texture – Texture of all normals. Use the RGB or RGBA channels according to the selected rendering mode.

Front Texture – Texture of the front normals. Use the RGB or RGBA channels according to the selected rendering mode.

Right Texture – Texture of the right normals. Use the RGB or RGBA channels according to the selected rendering mode.

Top Texture – Texture of the top normals. Use the RGB or RGBA channels according to the selected rendering mode.

Back Texture – Texture of the back normals. Use the RGB or RGBA channels according to the selected rendering mode.

Left Texture – Texture of the left normals. Use the RGB or RGBA channels according to the selected rendering mode.

Bottom Texture – Texture of the bottom normals. Use the RGB or RGBA channels according to the selected rendering mode.

Alpha cutoff – Alpha Cutoff value. Enabled if the selected rendering mode is Cutout.

Main Texture Level – Level of main texture in relation the source color.

Front Texture Level – Level of front texture in relation the source color.

Right Texture Level – Level of right texture in relation the source color.

Top Texture Level – Level of top texture in relation the source color.

Back Texture Level – Level of back texture in relation the source color.

Left Texture Level – Level of left texture in relation the source color.

Bottom Texture Level – Level of bottom texture in relation the source color.

Front Color Gradient – Front color gradient.

Front Top Color – Color of the top part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Front Gradient Center – Gradient center.

Front Gradient Width – Gradient width.

Front Gradient Revert – Revert the ortientation of the gradient.

Front Gradient Change Direction – Change direction of the gradient.

Front Gradient Rotation – Gradient rotation.

Right Color Gradient – Right color gradient.

Right Top Color – Color of the top part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Right Gradient Center – Gradient center.

Right Gradient Width – Gradient width.

Right Gradient Revert – Revert the ortientation of the gradient.

Right Gradient Change Direction – Change direction of the gradient.

Right Gradient Rotation – Gradient rotation.

Top Color Gradient – Top color gradient.

Top Top Color – Color of the top part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Top Gradient Center – Gradient center.

Top Gradient Width – Gradient width.

Top Gradient Revert – Revert the ortientation of the gradient.

Top Gradient Change Direction – Change direction of the gradient.

Top Gradient Rotation – Gradient rotation.

Back Color Gradient – Back color gradient.

Back Top Color – Color of the back part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Back Gradient Center – Gradient center.

Back Gradient Width – Gradient width.

Back Gradient Revert – Revert the ortientation of the gradient.

Back Gradient Change Direction – Change direction of the gradient.

Back Gradient Rotation – Gradient rotation.

Left Color Gradient – Left color gradient.

Left Top Color – Color of the left part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Left Gradient Center – Gradient center.

Left Gradient Width – Gradient width.

Left Gradient Revert – Revert the ortientation of the gradient.

Left Gradient Change Direction – Change direction of the gradient.

Left Gradient Rotation – Gradient rotation.

Bottom Color Gradient – Bottom color gradient.

Bottom Top Color – Color of the bottom part of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Bottom Gradient Center – Gradient center.

Bottom Gradient Width – Gradient width.

Bottom Gradient Revert – Revert the ortientation of the gradient.

Bottom Gradient Change Direction – Change direction of the gradient.

Bottom Gradient Rotation – Gradient rotation.

Opacity – Overall opacity value. Enabled if the selected rendering mode is Transparent or Cutout.

View Mode – If enabled the color is applied based on the view direction.

Height Fog – Fog by vertex height.

Height Fog Color – Color of the fog.

Height Fog Start Position – Start point of the fog.

Fog Height – Height of the fog.

Height Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn’t affect the transparent parts of the source color.

Local Height Fog – If enabled the fog is created based on the center of the mesh.

Height Fog Gizmo – If enabled show height fog gizmo.

Distance Fog – Fog by camera distance.

Distance Fog Color – Color of the fog.

Distance Fog Start Position – Start point of the fog.

Distance Length – Length of the fog.

Distance Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn’t affect the transparent parts of the source color.

World Distance Fog – If enabled the fog is created based on a position of the world.

World Distance Fog Position – Posición en el mundo de la niebla.

Distance Fog Gizmo – World position of the distance fog.

Light – Light and Ambient light.

Directional Light – If enabled main directional light affect to the source mesh.

Ambient – If enabled ambient light affect to the source mesh.

Simulated Light – Simulated Light.

Simulated Light Ramp Texture – Color ramp of the simulated light based on a texture. The top part of the texture is the center of the simulated light and the bottom part is the external part of the simulated light.

Simulated Light Level – Level of simulated light in relation the source color.

Simulated Light Position – World position of the simulated light.

Simulated Light Distance – Simulated light circunference diameter.

Gradient Ramp – If enabled uses a gradient ramp between two colors instead a ramp texture.

Center Color – Color of the center of the simulated light if gradient ramp is enabled.

Use External Color – If enabled uses a color for the external part of the light instead de source color.

External Color – Color of the expernal part of the simulated light if gradient ramp is enabled.

Additive Simulated Light – If enabled adds the simulated light color to the source color.

Additive Simulated Light Level – Level of simulated light addition in relation the source color.

Posterize – If enabled converts the ramp texture or the gradient ramp to multiple regions of fewer tones.

Steps – Color steps of the posterization.

Simulated Light Gizmo – If enabled show simulated light gizmo.

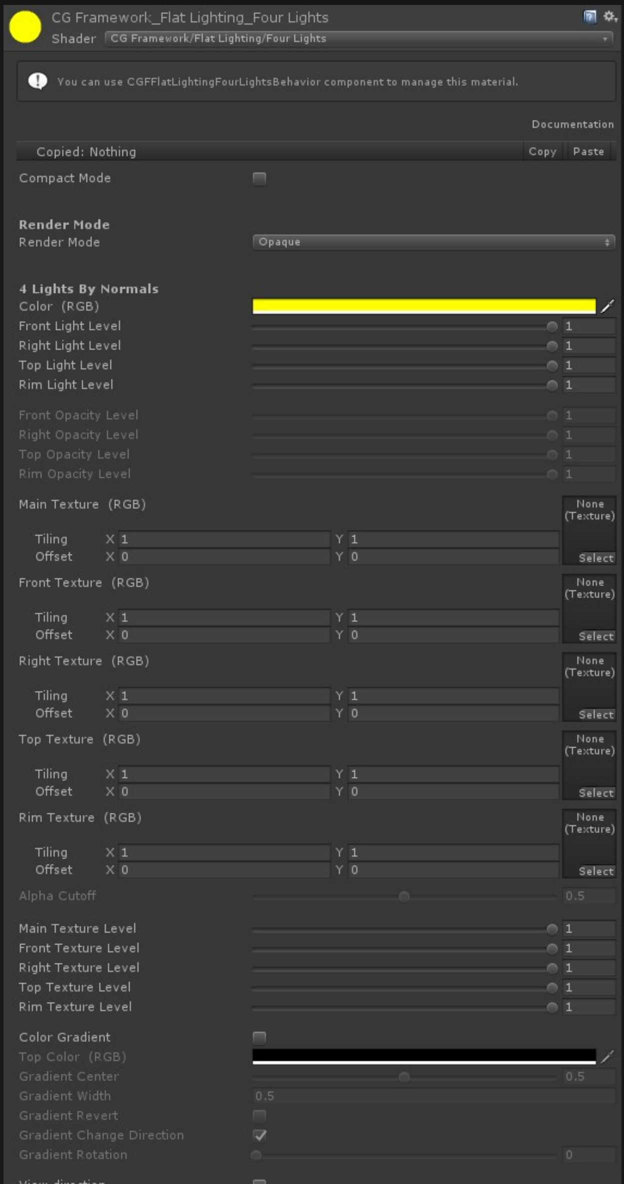
Use

Create a material with this shader and assign the material to the Material property of the MeshRenderer component.

FOUR LIGHTS

Files: FourLights.shader
Path: "CGF/Shaders/FlatLighting"
Shader menu: "CG Framework/Flat Lighting/Four Lights"

Shader that applies four lights based on the normals of the mesh.



CG Framework_Flat Lighting_Four Lights
Shader CG Framework/Flat Lighting/Four Lights

You can use CGFFlatLightingFourLightsBehavior component to manage this material.

Documentation

Copied: Nothing | Copy | Paste

Compact Mode ☐

Render Mode
Render Mode: Opaque

4 Lights By Normals
Color (RGB)
Front Light Level 1
Right Light Level 1
Top Light Level 1
Rim Light Level 1

Front Opacity Level 1
Right Opacity Level 1
Top Opacity Level 1
Rim Opacity Level 1

Main Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Front Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Right Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Top Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

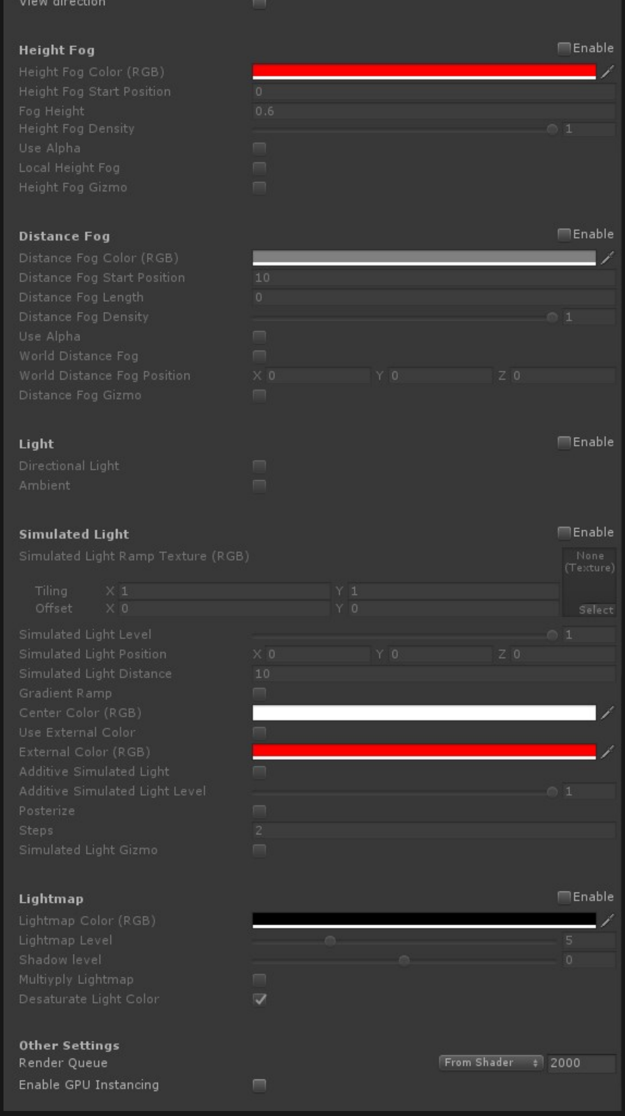
Rim Texture (RGB) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Alpha Cutoff 0.5

Main Texture Level 1
Front Texture Level 1
Right Texture Level 1
Top Texture Level 1
Rim Texture Level 1

Color Gradient ☐
Top Color (RGB)
Gradient Center 0.5
Gradient Width 0.5
Gradient Revert ☐
Gradient Change Direction ☑
Gradient Rotation 0

View direction ☐



View direction □

Height Fog □ Enable

Height Fog Color (RGB) 🟥

Height Fog Start Position 0

Fog Height 0.6

Height Fog Density 1

Use Alpha □

Local Height Fog □

Height Fog Gizmo □

Distance Fog □ Enable

Distance Fog Color (RGB)

Distance Fog Start Position 10

Distance Fog Length 0

Distance Fog Density 1

Use Alpha □

World Distance Fog □

World Distance Fog Position X 0 Y 0 Z 0

Distance Fog Gizmo □

Light □ Enable

Directional Light □

Ambient □

Simulated Light □ Enable

Simulated Light Ramp Texture (RGB) None (Texture) Select

Tiling X 1 Y 1

Offset X 0 Y 0

Simulated Light Level 1

Simulated Light Position X 0 Y 0 Z 0

Simulated Light Distance 10

Gradient Ramp □

Center Color (RGB)

Use External Color □

External Color (RGB) 🟥

Additive Simulated Light □

Additive Simulated Light Level 1

Posterize □

Steps 2

Simulated Light Gizmo □

Lightmap □ Enable

Lightmap Color (RGB)

Lightmap Level 5

Shadow level 0

Multiyply Lightmap □

Desaturate Light Color ☑

Other Settings

Render Queue From Shader 2000

Enable GPU Instancing □

Render Mode – Manages the rendering mode.

- **Opaque –** Solid objects without transparent areas.
- **Transparent –** Partial or total transparent objects.
- **Cutout –** Objects with a hard edge between transparents and opaque areas. The object can not have semi-transparent areas.
- **Background –** Solid objects without transparents areas that are rendered behind any other object.

Color – Color of the normals. Use the RGB or RGBA channels according to the selected rendering mode.

Front Light Level – Brightness of the light of the front normals.

Right Light Level – Brightness of the light of the right normals.

Top Light Level – Brightness of the light of the top normals.

Rim Light Level – Brightness of the light of the rim normals.

Front Opacity Level – Opacity of the color of the front normals.

Right Opacity Level – Opacity of the color of the right normals.

Top Opacity Level – Opacity of the color of the top normals.

Rim Opacity Level – Opacity of the color of the rim normals.

Main Texture – Texture of all normals. Use the RGB or RGBA channels according to the selected rendering mode.

Front Texture – Texture of the front normals. Use the RGB or RGBA channels according to the selected rendering mode.

Right Texture – Texture of the right normals. Use the RGB or RGBA channels according to the selected rendering mode.

Top Texture – Texture of the top normals. Use the RGB or RGBA channels according to the selected rendering mode.

Rim Texture – Texture of the rim normals. Use the RGB or RGBA channels according to the selected rendering mode.

Alpha cutoff – Alpha Cutoff value. Enabled if the selected rendering mode is Cutout.

Main Texture Level – Level of main texture in relation the source color.

Front Texture Level – Level of front texture in relation the source color.

Right Texture Level – Level of right texture in relation the source color.

Top Texture Level – Level of top texture in relation the source color.

Rim Texture Level – Level of rim texture in relation the source color.

Color Gradient – Color gradient.

Top Color – Color of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Gradient Center – Gradient center.

Gradient Width – Gradient width.

Gradient Revert – Revert the ortientation of the gradient.

Gradient Change Direction – Change direction of the gradient.

Gradient Rotation – Gradient rotation.

View Mode – If enabled the color is applied based on the view direction.

Height Fog – Fog by vertex height.

Height Fog Color – Color of the fog.

Height Fog Start Position – Start point of the fog.

Fog Height – Height of the fog.

Height Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

Local Height Fog – If enabled the fog is created based on the center of the mesh.

Height Fog Gizmo – If enabled show height fog gizmo.

Distance Fog – Fog by camera distance.

Distance Fog Color – Color of the fog.

Distance Fog Start Position – Start point of the fog.

Distance Length – Length of the fog.

Distance Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

World Distance Fog – If enabled the fog is created based on a position of the world.

World Distance Fog Position – Posición en el mundo de la niebla.

Distance Fog Gizmo – World position of the distance fog.

Light – Light and Ambient light.

Directional Light – If enabled main directional light affect to the source mesh.

Ambient – If enabled ambient light affect to the source mesh.

Simulated Light – Simulated Light.

Simulated Light Ramp Texture – Color ramp of the simulated light based on a texture. The top part of the texture is the center of the simulated light and the bottom part is the external part of the simulated light.

Simulated Light Level – Level of simulated light in relation the source color.

Simulated Light Position – World position of the simulated light.

Simulated Light Distance – Simulated light circunference diameter.

Gradient Ramp – If enabled uses a gradient ramp between two colors instead a ramp texture.

Center Color – Color of the center of the simulated light if gradient ramp is enabled.

Use External Color – If enabled uses a color for the external part of the light instead de source color.

External Color – Color of the expernal part of the simulated light if gradient ramp is enabled.

Additive Simulated Light – If enabled adds the simulated light color to the source color.

Additive Simulated Light Level – Level of simulated light addition in relation the source color.

Posterize – If enabled converts the ramp texture or the gradient ramp to multiple regions of fewer tones.

Steps – Color steps of the posterization.

Simulated Light Gizmo – If enabled show simulated light gizmo.

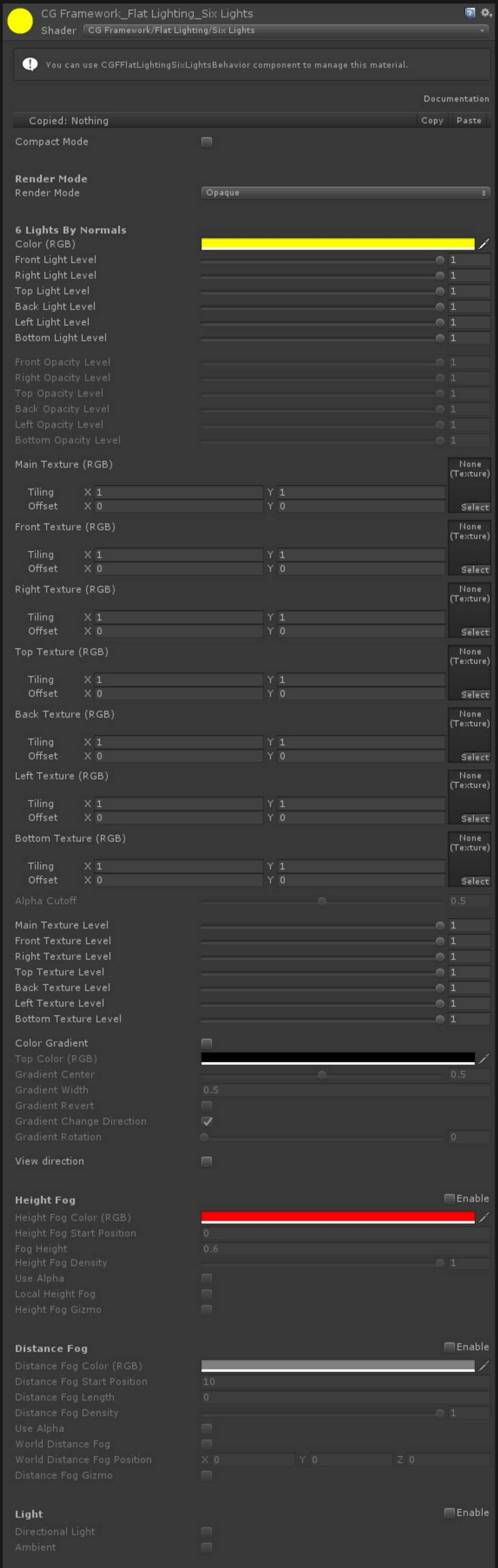
Use

Create a material with this shader and assign the material to the Material property of the MeshRenderer component.

SIX LIGHTS

Files: SixLights.shader
Path: “CGF/Shaders/FlatLighting”
Shader menu: “CG Framework/Flat Lighting/Six Lights”

Shader that applies six lights based on the normals of the mesh.



CG Framework_Flat Lighting_Six Lights
Shader: CG Framework/Flat Lighting/Six Lights

You can use CGFFlatLightingSixLightsBehavior component to manage this material.

Compact Mode ☐

Render Mode
Render Mode: Opaque

6 Lights By Normals
Color (RGB)
Front Light Level 1
Right Light Level 1
Top Light Level 1
Back Light Level 1
Left Light Level 1
Bottom Light Level 1

Front Opacity Level 1
Right Opacity Level 1
Top Opacity Level 1
Back Opacity Level 1
Left Opacity Level 1
Bottom Opacity Level 1

Main Texture (RGB): None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Front Texture (RGB): None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Right Texture (RGB): None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Top Texture (RGB): None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Back Texture (RGB): None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Left Texture (RGB): None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Bottom Texture (RGB): None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Alpha Cutoff 0.5

Main Texture Level 1
Front Texture Level 1
Right Texture Level 1
Top Texture Level 1
Back Texture Level 1
Left Texture Level 1
Bottom Texture Level 1

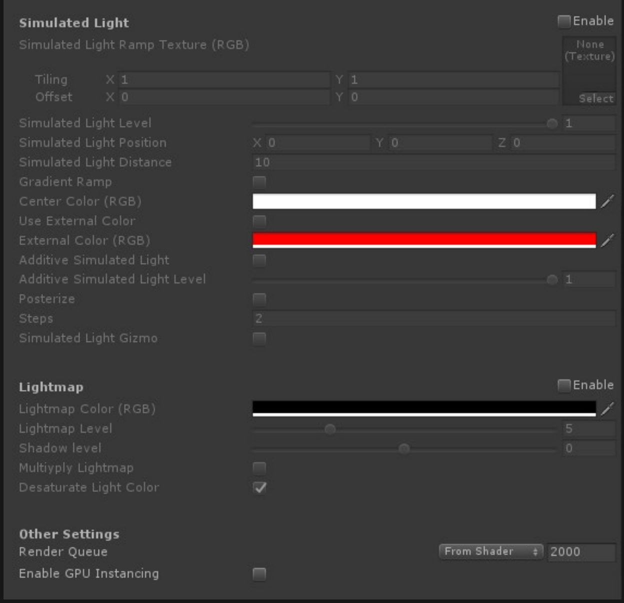
Color Gradient ☐
Top Color (RGB)
Gradient Center 0.5
Gradient Width 0.5
Gradient Revert ☐
Gradient Change Direction ☑
Gradient Rotation 0

View direction ☐

Height Fog ☐ Enable
Height Fog Color (RGB)
Height Fog Start Position 0
Fog Height 0.6
Height Fog Density 1
Use Alpha ☐
Local Height Fog ☐
Height Fog Gizmo ☐

Distance Fog ☐ Enable
Distance Fog Color (RGB)
Distance Fog Start Position 10
Distance Fog Length 0
Distance Fog Density 1
Use Alpha ☐
World Distance Fog ☐
World Distance Fog Position X 0 Y 0 Z 0
Distance Fog Gizmo ☐

Light ☐ Enable
Directional Light ☐
Ambient ☐



Simulated Light □ Enable

Simulated Light Ramp Texture (RGB)			None (Texture) Select
Tiling	X 1	Y 1	
Offset	X 0	Y 0	
Simulated Light Level	1		
Simulated Light Position	X 0	Y 0	Z 0
Simulated Light Distance	10		
Gradient Ramp	□		
Center Color (RGB)	⬜		
Use External Color	□		
External Color (RGB)	🟥		
Additive Simulated Light	□		
Additive Simulated Light Level	1		
Posterize	□		
Steps	2		
Simulated Light Gizmo	□		

Lightmap □ Enable

Lightmap Color (RGB)	⬛
Lightmap Level	5
Shadow level	0
Multiyply Lightmap	□
Desaturate Light Color	☑

Other Settings

Render Queue	From Shader 2000
Enable GPU Instancing	□

Render Mode – Manages the rendering mode.

- **Opaque –** Solid objects without transparent areas.
- **Transparent –** Partial or total transparent objects.
- **Cutout –** Objects with a hard edge between transparents and opaque areas. The object can not have semi-transparent areas.
- **Background –** Solid objects without transparents areas that are rendered behind any other object.

Color – Color of the normals. Use the RGB or RGBA channels according to the selected rendering mode.

Front Light Level – Brightness of the light of the front normals.

Right Light Level – Brightness of the light of the right normals.

Top Light Level – Brightness of the light of the top normals.

Back Light Level – Brightness of the light of the back normals.

Left Light Level – Brightness of the light of the left normals.

Bottom Light Level – Brightness of the light of the bottom normals.

Front Opacity Level – Opacity of the color of the front normals.

Right Opacity Level – Opacity of the color of the right normals.

Top Opacity Level – Opacity of the color of the top normals.

Back Opacity Level – Opacity of the color of the back normals.

Left Opacity Level – Opacity of the color of the left normals.

Bottom Opacity Level – Opacity of the color of the bottom normals.

Main Texture – Texture of all normals. Use the RGB or RGBA channels according to the selected rendering mode.

Front Texture – Texture of the front normals. Use the RGB or RGBA channels according to the selected rendering mode.

Right Texture – Texture of the right normals. Use the RGB or RGBA channels according to the selected rendering mode.

Top Texture – Texture of the top normals. Use the RGB or RGBA channels according to the selected rendering mode.

Back Texture – Texture of the back normals. Use the RGB or RGBA channels according to the selected rendering mode.

Left Texture – Texture of the left normals. Use the RGB or RGBA channels according to the selected rendering mode.

Bottom Texture – Texture of the bottom normals. Use the RGB or RGBA channels according to the selected rendering mode.

Alpha cutoff – Alpha Cutoff value. Enabled if the selected rendering mode is Cutout.

Main Texture Level – Level of main texture in relation the source color.

Front Texture Level – Level of front texture in relation the source color.

Right Texture Level – Level of right texture in relation the source color.

Top Texture Level – Level of top texture in relation the source color.

Back Texture Level – Level of back texture in relation the source color.

Left Texture Level – Level of left texture in relation the source color.

Bottom Texture Level – Level of bottom texture in relation the source color.

Color Gradient – Color gradient.

Top Color – Color of the gradient. Use the RGB or RGBA channels according to the selected rendering mode.

Gradient Center – Gradient center.

Gradient Width – Gradient width.

Gradient Revert – Revert the ortientation of the gradient.

Gradient Change Direction – Change direction of the gradient.

Gradient Rotation – Gradient rotation.

View Mode – If enabled the color is applied based on the view direction.

Height Fog – Fog by vertex height.

Height Fog Color – Color of the fog.

Height Fog Start Position – Start point of the fog.

Fog Height – Height of the fog.

Height Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

Local Height Fog – If enabled the fog is created based on the center of the mesh.

Height Fog Gizmo – If enabled show height fog gizmo.

Distance Fog – Fog by camera distance.

Distance Fog Color – Color of the fog.

Distance Fog Start Position – Start point of the fog.

Distance Length – Length of the fog.

Distance Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

World Distance Fog – If enabled the fog is created based on a position of the world.

World Distance Fog Position – Posición en el mundo de la niebla.

Distance Fog Gizmo – World position of the distance fog.

Light – Light and Ambient light.

Directional Light – If enabled main directional light affect to the source mesh.

Ambient – If enabled ambient light affect to the source mesh.

Simulated Light – Simulated Light.

Simulated Light Ramp Texture – Color ramp of the simulated light based on a texture. The top part of the texture is the center of the simulated light and the bottom part is the external part of the simulated light.

Simulated Light Level – Level of simulated light in relation the source color.

Simulated Light Position – World position of the simulated light.

Simulated Light Distance – Simulated light circunference diameter.

Gradient Ramp – If enabled uses a gradient ramp between two colors instead a ramp texture.

Center Color – Color of the center of the simulated light if gradient ramp is enabled.

Use External Color – If enabled uses a color for the external part of the light instead de source color.

External Color – Color of the expernal part of the simulated light if gradient ramp is enabled.

Additive Simulated Light – If enabled adds the simulated light color to the source color.

Additive Simulated Light Level – Level of simulated light addition in relation the source color.

Posterize – If enabled converts the ramp texture or the gradient ramp to multiple regions of fewer tones.

Steps – Color steps of the posterization.

Simulated Light Gizmo – If enabled show simulated light gizmo.

Use

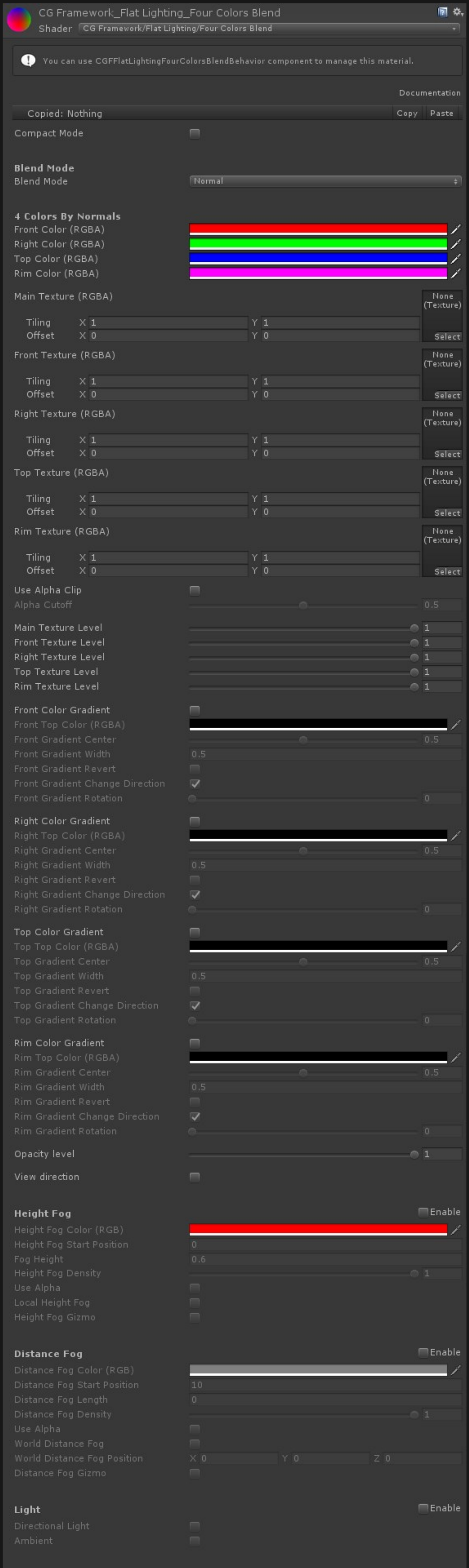
Create a material with this shader and assign the material to the Material property of the MeshRenderer component.

FOUR COLORS BLEND

Files: FourColorsBlend.shader

Path: “CGF/Shaders/FlatLighting”
Shader menu: “CG Framework/Flat Lighting/Four Colors Blend”

Shader that applies four colors based on the normals of the mesh with a blend mode.



CG Framework_Flat Lighting_Four Colors Blend
Shader CG Framework/Flat Lighting/Four Colors Blend

You can use CGFFlatLightingFourColorsBlendBehavior component to manage this material.

Documentation

Copied: Nothing Copy Paste

Compact Mode ☐

Blend Mode
Blend Mode Normal

4 Colors By Normals
Front Color (RGBA) 🟥
Right Color (RGBA) 🟩
Top Color (RGBA) 🟦
Rim Color (RGBA) 🟪

Main Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Front Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Right Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Top Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Rim Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Use Alpha Clip ☐
Alpha Cutoff 0.5

Main Texture Level 1
Front Texture Level 1
Right Texture Level 1
Top Texture Level 1
Rim Texture Level 1

Front Color Gradient ☐
Front Top Color (RGBA) ⬛
Front Gradient Center 0.5
Front Gradient Width 0.5
Front Gradient Revert ☐
Front Gradient Change Direction ☑
Front Gradient Rotation 0

Right Color Gradient ☐
Right Top Color (RGBA) ⬛
Right Gradient Center 0.5
Right Gradient Width 0.5
Right Gradient Revert ☐
Right Gradient Change Direction ☑
Right Gradient Rotation 0

Top Color Gradient ☐
Top Top Color (RGBA) ⬛
Top Gradient Center 0.5
Top Gradient Width 0.5
Top Gradient Revert ☐
Top Gradient Change Direction ☑
Top Gradient Rotation 0

Rim Color Gradient ☐
Rim Top Color (RGBA) ⬛
Rim Gradient Center 0.5
Rim Gradient Width 0.5
Rim Gradient Revert ☐
Rim Gradient Change Direction ☑
Rim Gradient Rotation 0

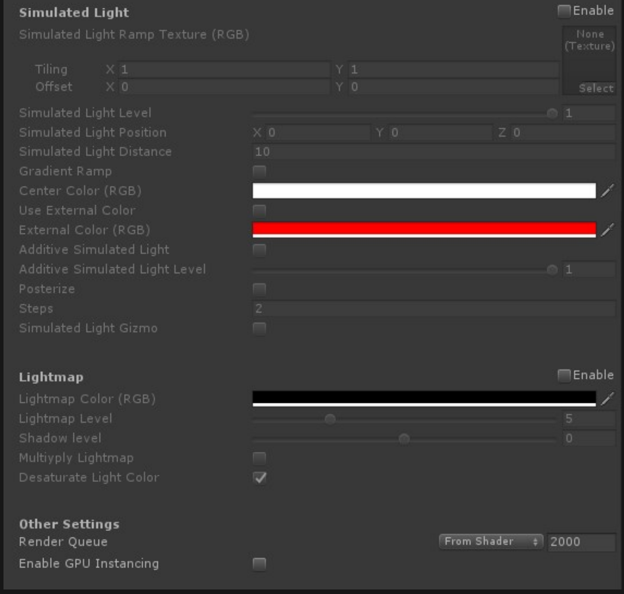
Opacity level 1

View direction ☐

Height Fog ☐ Enable
Height Fog Color (RGB) 🟥
Height Fog Start Position 0
Fog Height 0.6
Height Fog Density 1
Use Alpha ☐
Local Height Fog ☐
Height Fog Gizmo ☐

Distance Fog ☐ Enable
Distance Fog Color (RGB) ⬜
Distance Fog Start Position 10
Distance Fog Length 0
Distance Fog Density 1
Use Alpha ☐
World Distance Fog ☐
World Distance Fog Position X 0 Y 0 Z 0
Distance Fog Gizmo ☐

Light ☐ Enable
Directional Light ☐
Ambient ☐



Simulated Light □ Enable

Simulated Light Ramp Texture (RGB)		None (Texture) Select
Tiling	X 1	Y 1
Offset	X 0	Y 0
Simulated Light Level	1	
Simulated Light Position	X 0 Y 0 Z 0	
Simulated Light Distance	10	
Gradient Ramp	□	
Center Color (RGB)	⬜	
Use External Color	□	
External Color (RGB)	🟥	
Additive Simulated Light	□	
Additive Simulated Light Level	1	
Posterize	□	
Steps	2	
Simulated Light Gizmo	□	

Lightmap □ Enable

Lightmap Color (RGB)	⬛
Lightmap Level	5
Shadow level	0
Multiyply Lightmap	□
Desaturate Light Color	☑

Other Settings

Render Queue	From Shader 2000
Enable GPU Instancing	□

Blend Mode – Manages de blending mode.

- **Normal –** Without blending mode.
- **Darken –** Looks at the color information in each channel and selects the base or blend color—whichever is darker—as the result color. Pixels lighter than the blend color are replaced, and pixels darker than the blend color do not change.
- **Multiply –** Looks at the color information in each channel and multiplies the base color by the blend color. The result color is always a darker color. Multiplying any color with black produces black. Multiplying any color with white leaves the color unchanged. When you're painting with a color other than black or white, successive strokes with a painting tool produce progressively darker colors. The effect is similar to drawing on the image with multiple marking pens.
- **Color Burn –** Looks at the color information in each channel and darkens the base color to reflect the blend color by increasing the contrast between the two. Blending with white produces no change.
- **Linear Burn –** Looks at the color information in each channel and darkens the base color to reflect the blend color by decreasing the brightness. Blending with white produces no change.
- **Darker Color –** Compares the total of all channel values for the blend and base color and displays the lower value color. Darker Color does not produce a third color, which can result from the Darken blend, because it chooses the lowest channel values from both the base and the blend color to create the result color.
- **Lighten –** Looks at the color information in each channel and selects the base or blend color—whichever is lighter—as the result color. Pixels darker than the blend color are replaced, and pixels lighter than the blend color do not change.
- **Screen –** Looks at each channel's color information and multiplies the inverse of the blend and base colors. The result color is always a lighter color. Screening with black leaves the color unchanged. Screening with white produces white. The effect is similar to projecting multiple photographic slides on top of each other.
- **Color Dodge –** Looks at the color information in each channel and brightens the base color to reflect the blend color by decreasing contrast between the two. Blending with black produces no change.
- **Linear Dodge Or Additive –** Looks at the color information in each channel and brightens the base color to reflect the blend color by increasing the brightness. Blending with black produces no change.
- **Lighter Color –** Compares the total of all channel values for the blend and base color and displays the higher value color. Lighter Color does not produce a third color, which can result from the Lighten blend, because it chooses the highest channel values from both the base and blend color to create the result color.
- **Overlay –** Multiplies or screens the colors, depending on the base color. Patterns or colors overlay the existing pixels while preserving the highlights and shadows of the base color. The base color is not replaced, but mixed with the blend color to reflect the lightness or darkness of the original color.
- **Soft Light –** Darkens or lightens the colors, depending on the blend color. The effect is similar to shining a diffused spotlight on the image. If the blend color (light source) is lighter than 50% gray, the image is lightened as if it were dodged. If the blend color is darker than 50% gray, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.
- **Hard Light –** Multiplies or screens the colors, depending on the blend color. The effect is similar to shining a harsh spotlight on the image. If the blend color (light source) is lighter than 50% gray, the image is lightened, as if it were screened. This is useful for adding highlights to an image. If the blend color is darker than 50% gray, the image is darkened, as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.
- **Vivid Light –** Burns or dodges the colors by increasing or decreasing the contrast, depending on the blend color. If the blend color (light source) is lighter than 50% gray, the image is lightened by decreasing the contrast. If the blend color is darker than 50% gray, the image is darkened by increasing the contrast.
- **Linear Light –** Burns or dodges the colors by decreasing or increasing the brightness, depending on the blend color. If the blend color (light source) is lighter than 50% gray, the image is lightened by increasing the brightness. If the blend color is darker than 50% gray, the image is darkened by decreasing the brightness.
- **Pin Light –** Replaces the colors, depending on the blend color. If the blend color (light source) is lighter than 50% gray, pixels darker than the blend color are replaced, and pixels lighter than the blend color do not change. If the blend color is darker than 50% gray, pixels lighter than the blend color are replaced, and pixels darker than the blend color do not change. This is useful for adding special effects to an image.
- **Hard Mix-** Adds the red, green and blue channel values of the blend color to the RGB values of the base color. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if less than 255, a value of 0. Therefore, all blended pixels have red, green, and blue channel values of either 0 or 255. This changes all pixels to primary additive colors (red, green, or blue), white, or black.
- **Difference –** Looks at the color information in each channel and subtracts either the blend color from the base color or the base color from the blend color, depending on which has the greater brightness value. Blending with white inverts the base color values; blending with black produces no change.
- **Exclusion –** Creates an effect similar to but lower in contrast than the Difference mode. Blending with white inverts the base color values. Blending with black produces no change.
- **Subtract –** Looks at the color information in each channel and subtracts the blend color from the base color. In 8- and 16-bit images, any resulting negative values are clipped to zero.
- **Divide –** Looks at the color information in each channel and divides the blend color from the base color.
- **Hue –** Creates a result color with the luminance and saturation of the base color and the hue of the blend color.
- **Saturation –** Creates a result color with the luminance and hue of the base color and the saturation of the blend color. Painting with this mode in an area with no (0) saturation (gray) causes no change.
- **Color –** Creates a result color with the luminance of the base color and the hue and saturation of the blend color. This preserves the gray levels in the image and is useful for coloring monochrome images and for tinting color images.
- **Luminosity –** Creates a result color with the hue and saturation of the base color and the luminance of the blend color. This mode creates the inverse effect of Color mode.

Front Color – Color of the front normals. Use the RGBA channels.

Right Color – Color of the right normals. Use the RGBA channels.

Top Color – Color of the top normals. Use the RGBA channels.

Rim Color – Color of the rim normals. Use the RGBA channels.

Main Texture – Texture of all normals. Use the RGBA channels.

Front Texture – Texture of the front normals. Use the RGBA channels.

Right Texture – Texture of the right normals. Use the RGBA channels.

Top Texture – Texture of the top normals. Use the RGBA channels.

Rim Texture – Texture of the rim normals. Use the RGBA channels.

Use Alpha Clip – Enables Alpha Clip.

Alpha cutoff – Alpha Cutoff value. Enabled if the selected rendering mode is Cutout.

Main Texture Level – Level of main texture in relation the source color.

Front Texture Level – Level of front texture in relation the source color.

Right Texture Level – Level of right texture in relation the source color.

Top Texture Level – Level of top texture in relation the source color.

Rim Texture Level – Level of rim texture in relation the source color.

Front Color Gradient – Front color gradient.

Front Top Color – Color of the top part of the gradient. Use the RGBA channels.

Front Gradient Center – Gradient center.

Front Gradient Width – Gradient width.

Front Gradient Revert – Revert the ortientation of the gradient.

Front Gradient Change Direction – Change direction of the gradient.

Front Gradient Rotation – Gradient rotation.

Right Color Gradient – Right color gradient.

Right Top Color – Color of the top part of the gradient. Use the RGBA channels.

Right Gradient Center – Gradient center.

Right Gradient Width – Gradient width.

Right Gradient Revert – Revert the ortientation of the gradient.

Right Gradient Change Direction – Change direction of the gradient.

Right Gradient Rotation – Gradient rotation.

Top Color Gradient – Top color gradient.

Top Top Color – Color of the top part of the gradient. Use the RGBA channels.

Top Gradient Center – Gradient center.

Top Gradient Width – Gradient width.

Top Gradient Revert – Revert the ortientation of the gradient.

Top Gradient Change Direction – Change direction of the gradient.

Top Gradient Rotation – Gradient rotation.

Rim Color Gradient – Right color gradient.

Rim Top Color – Color of the top part of the gradient. Use the RGBA channels.

Rim Gradient Center – Gradient center.

Rim Gradient Width – Gradient width.

Rim Gradient Revert – Revert the ortientation of the gradient.

Rim Gradient Change Direction – Change direction of the gradient.

Rim Gradient Rotation – Gradient rotation.

Opacity – Overall opacity value. Enabled if the selected rendering mode is Transparent or Cutout.

View Mode – If enabled the color is applied based on the view direction.

Height Fog – Fog by vertex height.

Height Fog Color – Color of the fog.

Height Fog Start Position – Start point of the fog.

Fog Height – Height of the fog.

Height Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

Local Height Fog – If enabled the fog is created based on the center of the mesh.

Height Fog Gizmo – If enabled show height fog gizmo.

Distance Fog – Fog by camera distance.

Distance Fog Color – Color of the fog.

Distance Fog Start Position – Start point of the fog.

Distance Length – Length of the fog.

Distance Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

World Distance Fog – If enabled the fog is created based on a position of the world.

World Distance Fog Position – Posición en el mundo de la niebla.

Distance Fog Gizmo – World position of the distance fog.

Light – Light and Ambient light.

Directional Light – If enabled main directional light affect to the source mesh.

Ambient – If enabled ambient light affect to the source mesh.

Simulated Light – Simulated Light.

Simulated Light Ramp Texture – Color ramp of the simulated light based on a texture. The top part of the texture is the center of the simulated light and the bottom part is the external part of the simulated light.

Simulated Light Level – Level of simulated light in relation the source color.

Simulated Light Position – World position of the simulated light.

Simulated Light Distance – Simulated light circunference diameter.

Gradient Ramp – If enabled uses a gradient ramp between two colors instead a ramp texture.

Center Color – Color of the center of the simulated light if gradient ramp is enabled.

Use External Color – If enabled uses a color for the external part of the light instead de source color.

External Color – Color of the expernal part of the simulated light if gradient ramp is enabled.

Additive Simulated Light – If enabled adds the simulated light color to the source color.

Additive Simulated Light Level – Level of simulated light addition in relation the source color.

Posterize – If enabled converts the ramp texture or the gradient ramp to multiple regions of fewer tones.

Steps – Color steps of the posterization.

Simulated Light Gizmo – If enabled show simulated light gizmo.

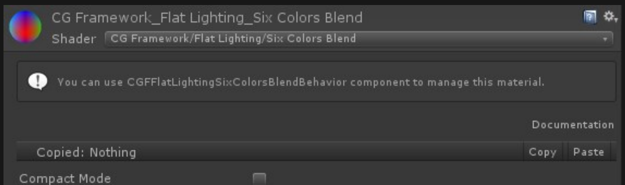
Use

Create a material with this shader and assign the material to the Material property of the MeshRenderer component.

SIX COLORS BLEND

Files: SixColorsBlend.shader
Path: "CGF/Shaders/FlatLighting"
Shader menu: "CG Framework/Flat Lighting/Six Colors Blend"

Shader that applies six colors based on the normals of the mesh with a blend mode.



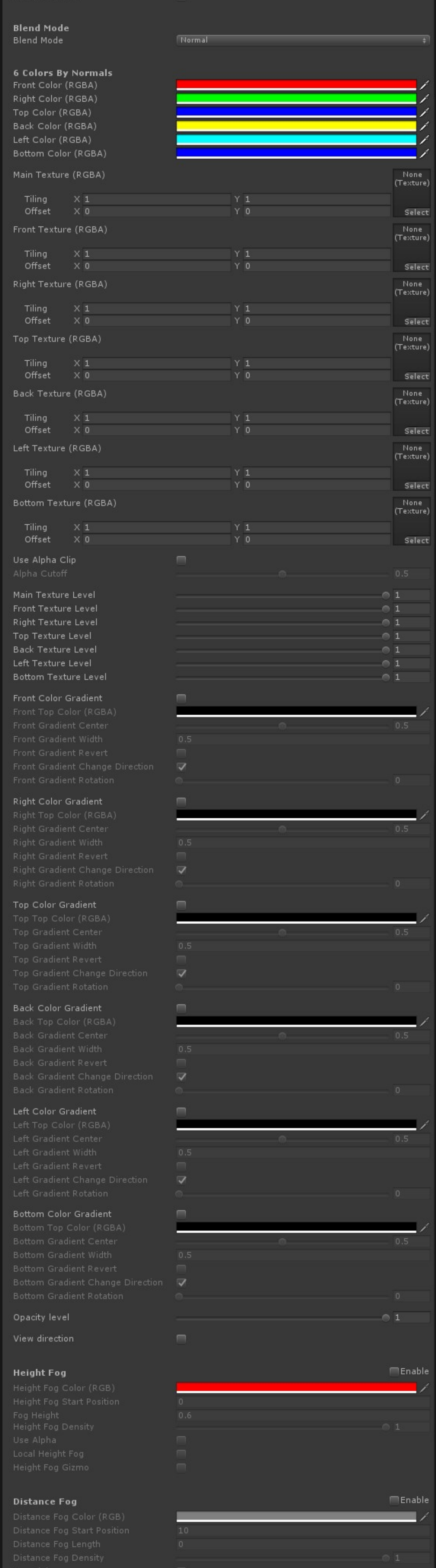
CG Framework_Flat Lighting_Six Colors Blend

Shader CG Framework/Flat Lighting/Six Colors Blend

You can use CGFFlatLightingSixColorsBlendBehavior component to manage this material.

Copied: Nothing

Compact Mode □



Blend Mode

Blend Mode: Normal

6 Colors By Normals

Front Color (RGBA) 🟥

Right Color (RGBA) 🟩

Top Color (RGBA) 🟦

Back Color (RGBA) 🟨

Left Color (RGBA) 🟦

Bottom Color (RGBA) 🟦

Main Texture (RGBA) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Front Texture (RGBA) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Right Texture (RGBA) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Top Texture (RGBA) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Back Texture (RGBA) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Left Texture (RGBA) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Bottom Texture (RGBA) — None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Use Alpha Clip □
Alpha Cutoff 0.5

Main Texture Level 1
Front Texture Level 1
Right Texture Level 1
Top Texture Level 1
Back Texture Level 1
Left Texture Level 1
Bottom Texture Level 1

Front Color Gradient □
Front Top Color (RGBA) ⬛
Front Gradient Center 0.5
Front Gradient Width 0.5
Front Gradient Revert □
Front Gradient Change Direction ☑
Front Gradient Rotation 0

Right Color Gradient □
Right Top Color (RGBA) ⬛
Right Gradient Center 0.5
Right Gradient Width 0.5
Right Gradient Revert □
Right Gradient Change Direction ☑
Right Gradient Rotation 0

Top Color Gradient □
Top Top Color (RGBA) ⬛
Top Gradient Center 0.5
Top Gradient Width 0.5
Top Gradient Revert □
Top Gradient Change Direction ☑
Top Gradient Rotation 0

Back Color Gradient □
Back Top Color (RGBA) ⬛
Back Gradient Center 0.5
Back Gradient Width 0.5
Back Gradient Revert □
Back Gradient Change Direction ☑
Back Gradient Rotation 0

Left Color Gradient □
Left Top Color (RGBA) ⬛
Left Gradient Center 0.5
Left Gradient Width 0.5
Left Gradient Revert □
Left Gradient Change Direction ☑
Left Gradient Rotation 0

Bottom Color Gradient □
Bottom Top Color (RGBA) ⬛
Bottom Gradient Center 0.5
Bottom Gradient Width 0.5
Bottom Gradient Revert □
Bottom Gradient Change Direction ☑
Bottom Gradient Rotation 0

Opacity level 1

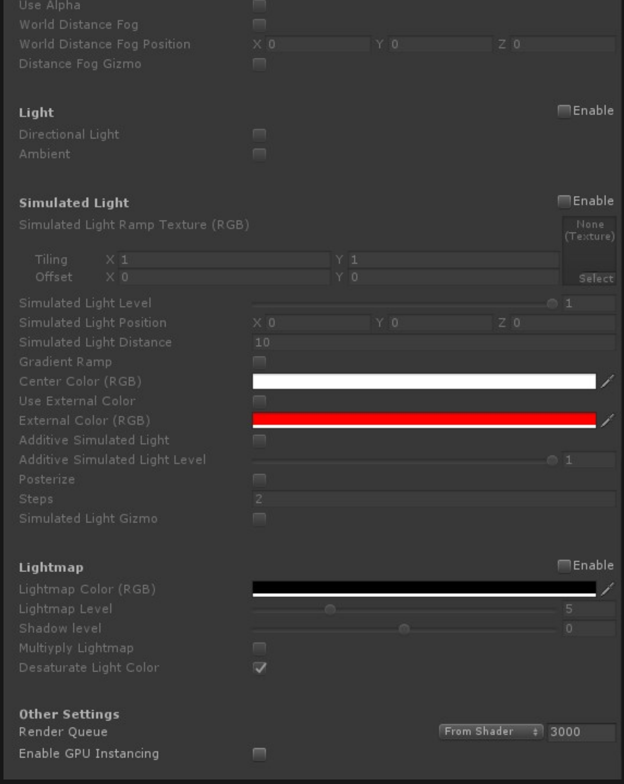
View direction □

Height Fog □ Enable

Height Fog Color (RGB) 🟥
Height Fog Start Position 0
Fog Height 0.6
Height Fog Density 1
Use Alpha □
Local Height Fog □
Height Fog Gizmo □

Distance Fog □ Enable

Distance Fog Color (RGB) ⬜
Distance Fog Start Position 10
Distance Fog Length 0
Distance Fog Density 1



```
Use Alpha                            □
World Distance Fog                   □
World Distance Fog Position          X 0          Y 0          Z 0
Distance Fog Gizmo                   □

Light                                                         □Enable
Directional Light                    □
Ambient                              □

Simulated Light                                               □Enable
Simulated Light Ramp Texture (RGB)                            None
                                                              (Texture)
  Tiling        X 1                        Y 1
  Offset        X 0                        Y 0                Select
Simulated Light Level                                         1
Simulated Light Position             X 0          Y 0          Z 0
Simulated Light Distance             10
Gradient Ramp                        □
Center Color (RGB)                   ⬜
Use External Color                   □
External Color (RGB)                 🟥
Additive Simulated Light             □
Additive Simulated Light Level                                1
Posterize                            □
Steps                                2
Simulated Light Gizmo                □

Lightmap                                                      □Enable
Lightmap Color (RGB)                 ⬛
Lightmap Level                                                5
Shadow level                                                  0
Multiyply Lightmap                   □
Desaturate Light Color               ☑

Other Settings
Render Queue                                   From Shader ⬍  3000
Enable GPU Instancing                □
```

Blend Mode – Manages de blending mode.

- **Normal –** Without blending mode.
- **Darken –** Looks at the color information in each channel and selects the base or blend color—whichever is darker—as the result color. Pixels lighter than the blend color are replaced, and pixels darker than the blend color do not change.
- **Multiply –** Looks at the color information in each channel and multiplies the base color by the blend color. The result color is always a darker color. Multiplying any color with black produces black. Multiplying any color with white leaves the color unchanged. When you're painting with a color other than black or white, successive strokes with a painting tool produce progressively darker colors. The effect is similar to drawing on the image with multiple marking pens.
- **Color Burn –** Looks at the color information in each channel and darkens the base color to reflect the blend color by increasing the contrast between the two. Blending with white produces no change.
- **Linear Burn –** Looks at the color information in each channel and darkens the base color to reflect the blend color by decreasing the brightness. Blending with white produces no change.
- **Darker Color –** Compares the total of all channel values for the blend and base color and displays the lower value color. Darker Color does not produce a third color, which can result from the Darken blend, because it chooses the lowest channel values from both the base and the blend color to create the result color.
- **Lighten –** Looks at the color information in each channel and selects the base or blend color—whichever is lighter—as the result color. Pixels darker than the blend color are replaced, and pixels lighter than the blend color do not change.
- **Screen –** Looks at each channel's color information and multiplies the inverse of the blend and base colors. The result color is always a lighter color. Screening with black leaves the color unchanged. Screening with white produces white. The effect is similar to projecting multiple photographic slides on top of each other.
- **Color Dodge –** Looks at the color information in each channel and brightens the base color to reflect the blend color by decreasing contrast between the two. Blending with black produces no change.
- **Linear Dodge Or Additive –** Looks at the color information in each channel and brightens the base color to reflect the blend color by increasing the brightness. Blending with black produces no change.
- **Lighter Color –** Compares the total of all channel values for the blend and base color and displays the higher value color. Lighter Color does not produce a third color, which can result from the Lighten blend, because it chooses the highest channel values from both the base and blend color to create the result color.
- **Overlay –** Multiplies or screens the colors, depending on the base color. Patterns or colors overlay the existing pixels while preserving the highlights and shadows of the base color. The base color is not replaced, but mixed with the blend color to reflect the lightness or darkness of the original color.
- **Soft Light –** Darkens or lightens the colors, depending on the blend color. The effect is similar to shining a diffused spotlight on the image. If the blend color (light source) is lighter than 50% gray, the image is lightened as if it were dodged. If the blend color is darker than 50% gray, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.
- **Hard Light –** Multiplies or screens the colors, depending on the blend color. The effect is similar to shining a harsh spotlight on the image. If the blend color (light source) is lighter than 50% gray, the image is lightened, as if it were screened. This is useful for adding highlights to an image. If the blend color is darker than 50% gray, the image is darkened, as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.
- **Vivid Light –** Burns or dodges the colors by increasing or decreasing the contrast, depending on the blend color. If the blend color (light source) is lighter than 50% gray, the image is lightened by decreasing the contrast. If the blend color is darker than 50% gray, the image is darkened by increasing the contrast.
- **Linear Light –** Burns or dodges the colors by decreasing or increasing the brightness, depending on the blend color. If the blend color (light source) is lighter than 50% gray, the image is lightened by increasing the brightness. If the blend color is darker than 50% gray, the image is darkened by decreasing the brightness.
- **Pin Light –** Replaces the colors, depending on the blend color. If the blend color (light source) is lighter than 50% gray, pixels darker than the blend color are replaced, and pixels lighter than the blend color do not change. If the blend color is darker than 50% gray, pixels lighter than the blend color are replaced, and pixels darker than the blend color do not change. This is useful for adding special effects to an image.
- **Hard Mix-** Adds the red, green and blue channel values of the blend color to the RGB values of the base color. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if less than 255, a value of 0. Therefore, all blended pixels have red, green, and blue channel values of either 0 or 255. This changes all pixels to primary additive colors (red, green, or blue), white, or black.
- **Difference –** Looks at the color information in each channel and subtracts either the blend color from the base color or the base color from the blend color, depending on which has the greater brightness value. Blending with white inverts the base color values; blending with black produces no change.
- **Exclusion –** Creates an effect similar to but lower in contrast than the Difference mode. Blending with white inverts the base color values. Blending with black produces no change.
- **Subtract –** Looks at the color information in each channel and subtracts the blend color from the base color. In 8- and 16-bit images, any resulting negative values are clipped to zero.
- **Divide –** Looks at the color information in each channel and divides the blend color from the base color.

- **Hue –** Creates a result color with the luminance and saturation of the base color and the hue of the blend color.
- **Saturation –** Creates a result color with the luminance and hue of the base color and the saturation of the blend color. Painting with this mode in an area with no

- **Luminosity –** Creates a result color with the hue and saturation of the base color and the luminance of the blend color. This mode creates the inverse effect of Color mode.

Front Color – Color of the front normals. Use the RGBA channels.

Right Color – Color of the right normals. Use the RGBA channels.

Top Color – Color of the top normals. Use the RGBA channels.

Back Color – Color of the back normals. Use the RGBA channels.

Left Color – Color of the left normals. Use the RGBA channels.

Bottom Color – Color of the bottom normals. Use the RGBA channels.

Main Texture – Texture of all normals. Use the RGBA channels.

Front Texture – Texture of the front normals. Use the RGBA channels.

Right Texture – Texture of the right normals. Use the RGBA channels.

Top Texture – Texture of the top normals. Use the RGBA channels.

Back Texture – Texture of the back normals. Use the RGBA channels.

Left Texture – Texture of the left normals. Use the RGBA channels.

Bottom Texture – Texture of the bottom normals. Use the RGBA channels.

Use Alpha Clip – Enables Alpha Clip.

Alpha cutoff – Alpha Cutoff value. Enabled if the selected rendering mode is Cutout.

Main Texture Level – Level of main texture in relation the source color.

Front Texture Level – Level of front texture in relation the source color.

Right Texture Level – Level of right texture in relation the source color.

Top Texture Level – Level of top texture in relation the source color.

Back Texture Level – Level of back texture in relation the source color.

Left Texture Level – Level of left texture in relation the source color.

Bottom Texture Level – Level of bottom texture in relation the source color.

Front Color Gradient – Front color gradient.

Front Top Color – Color of the top part of the gradient. Use the RGBA channels.

Front Gradient Center – Gradient center.

Front Gradient Width – Gradient width.

Front Gradient Revert – Revert the ortientation of the gradient.

Front Gradient Change Direction – Change direction of the gradient.

Front Gradient Rotation – Gradient rotation.

Right Color Gradient – Right color gradient.

Right Top Color – Color of the top part of the gradient. Use the RGBA channels.

Right Gradient Center – Gradient center.

Right Gradient Width – Gradient width.

Right Gradient Revert – Revert the ortientation of the gradient.

Right Gradient Change Direction – Change direction of the gradient.

Right Gradient Rotation – Gradient rotation.

Top Color Gradient – Top color gradient.

Top Top Color – Color of the top part of the gradient. Use the RGBA channels.

Top Gradient Center – Gradient center.

Top Gradient Width – Gradient width.

Top Gradient Revert – Revert the ortientation of the gradient.

Top Gradient Change Direction – Change direction of the gradient.

Top Gradient Rotation – Gradient rotation.

Back Color Gradient – Back color gradient.

Back Top Color – Color of the back part of the gradient. Use the RGBA channels.

Back Gradient Center – Gradient center.

Back Gradient Width – Gradient width.

Back Gradient Revert – Revert the ortientation of the gradient.

Back Gradient Change Direction – Change direction of the gradient.

Back Gradient Rotation – Gradient rotation.

Left Color Gradient – Left color gradient.

Left Top Color – Color of the left part of the gradient. Use the RGBA channels.

Left Gradient Center – Gradient center.

Left Gradient Width – Gradient width.

Left Gradient Revert – Revert the ortientation of the gradient.

Left Gradient Change Direction – Change direction of the gradient.

Left Gradient Rotation – Gradient rotation.

Bottom Color Gradient – Bottom color gradient.

Bottom Top Color – Color of the bottom part of the gradient. Use the RGBA channels.

Bottom Gradient Center – Gradient center.

Bottom Gradient Width – Gradient width.

Bottom Gradient Revert – Revert the ortientation of the gradient.

Bottom Gradient Change Direction – Change direction of the gradient.

Bottom Gradient Rotation – Gradient rotation.

Opacity – Overall opacity value. Enabled if the selected rendering mode is Transparent or Cutout.

View Mode – If enabled the color is applied based on the view direction.

Height Fog – Fog by vertex height.

Height Fog Color – Color of the fog.

Height Fog Start Position – Start point of the fog.

Fog Height – Height of the fog.

Height Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

Local Height Fog – If enabled the fog is created based on the center of the mesh.

Height Fog Gizmo – If enabled show height fog gizmo.

Distance Fog – Fog by camera distance.

Distance Fog Color – Color of the fog.

Distance Fog Start Position – Start point of the fog.

Distance Length – Length of the fog.

Distance Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

World Distance Fog – If enabled the fog is created based on a position of the world.

World Distance Fog Position – Posición en el mundo de la niebla.

Distance Fog Gizmo – World position of the distance fog.

Light – Light and Ambient light.

Directional Light – If enabled main directional light affect to the source mesh.

Ambient – If enabled ambient light affect to the source mesh.

Simulated Light – Simulated Light.

Simulated Light Ramp Texture – Color ramp of the simulated light based on a texture. The top part of the texture is the center of the simulated light and the bottom part is the external part of the simulated light.

Simulated Light Level – Level of simulated light in relation the source color.

Simulated Light Position – World position of the simulated light.

Simulated Light Distance – Simulated light circunference diameter.

Gradient Ramp – If enabled uses a gradient ramp between two colors instead a ramp texture.

Center Color – Color of the center of the simulated light if gradient ramp is enabled.

Use External Color – If enabled uses a color for the external part of the light instead de source color.

External Color – Color of the expernal part of the simulated light if gradient ramp is enabled.

Additive Simulated Light – If enabled adds the simulated light color to the source color.

Additive Simulated Light Level – Level of simulated light addition in relation the source color.

Posterize – If enabled converts the ramp texture or the gradient ramp to multiple regions of fewer tones.

Steps – Color steps of the posterization.

Simulated Light Gizmo – If enabled show simulated light gizmo.

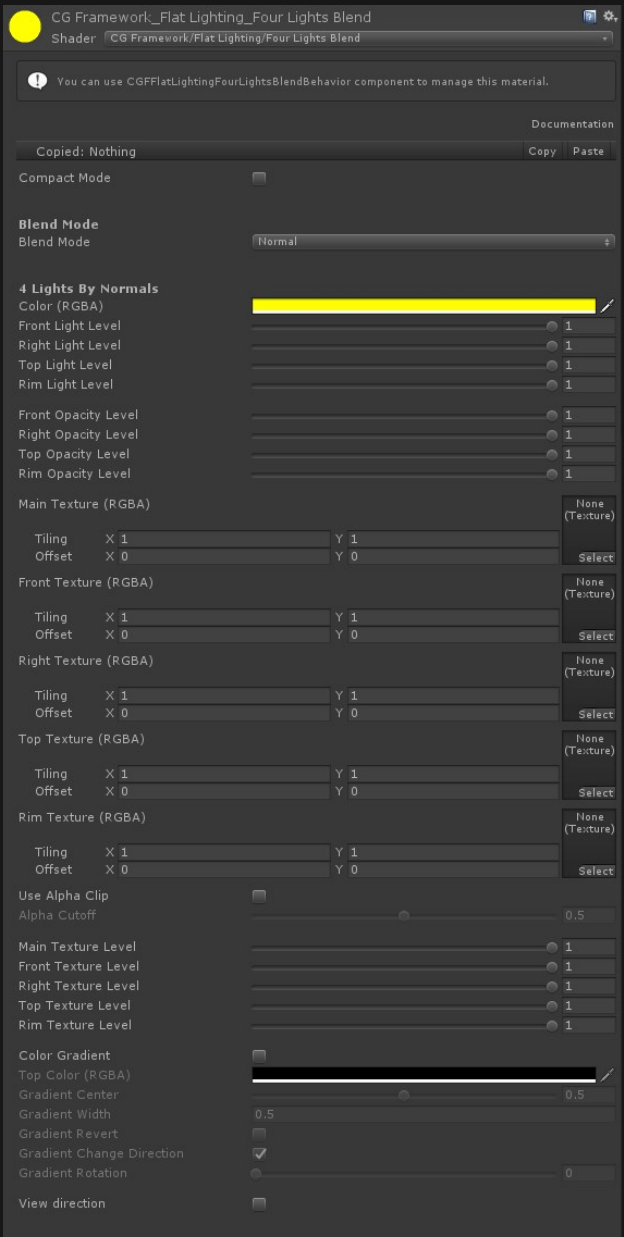
Use

Create a material with this shader and assign the material to the Material property of the MeshRenderer component.

FOUR LIGHTS BLEND

Files: FourLightsBlend.shader
Path: “CGF/Shaders/FlatLighting”
Shader menu: “CG Framework/Flat Lighting/Four Lights Blend”

Shader that applies four lights based on the normals of the mesh with a blend mode.



CG Framework_Flat Lighting_Four Lights Blend
Shader: CG Framework/Flat Lighting/Four Lights Blend

You can use CGFFlatLightingFourLightsBlendBehavior component to manage this material.

Documentation

Copied: Nothing — Copy — Paste

Compact Mode ☐

Blend Mode
Blend Mode: Normal

4 Lights By Normals
Color (RGBA)
Front Light Level: 1
Right Light Level: 1
Top Light Level: 1
Rim Light Level: 1

Front Opacity Level: 1
Right Opacity Level: 1
Top Opacity Level: 1
Rim Opacity Level: 1

Main Texture (RGBA): None (Texture) — Select
Tiling X 1 Y 1
Offset X 0 Y 0

Front Texture (RGBA): None (Texture) — Select
Tiling X 1 Y 1
Offset X 0 Y 0

Right Texture (RGBA): None (Texture) — Select
Tiling X 1 Y 1
Offset X 0 Y 0

Top Texture (RGBA): None (Texture) — Select
Tiling X 1 Y 1
Offset X 0 Y 0

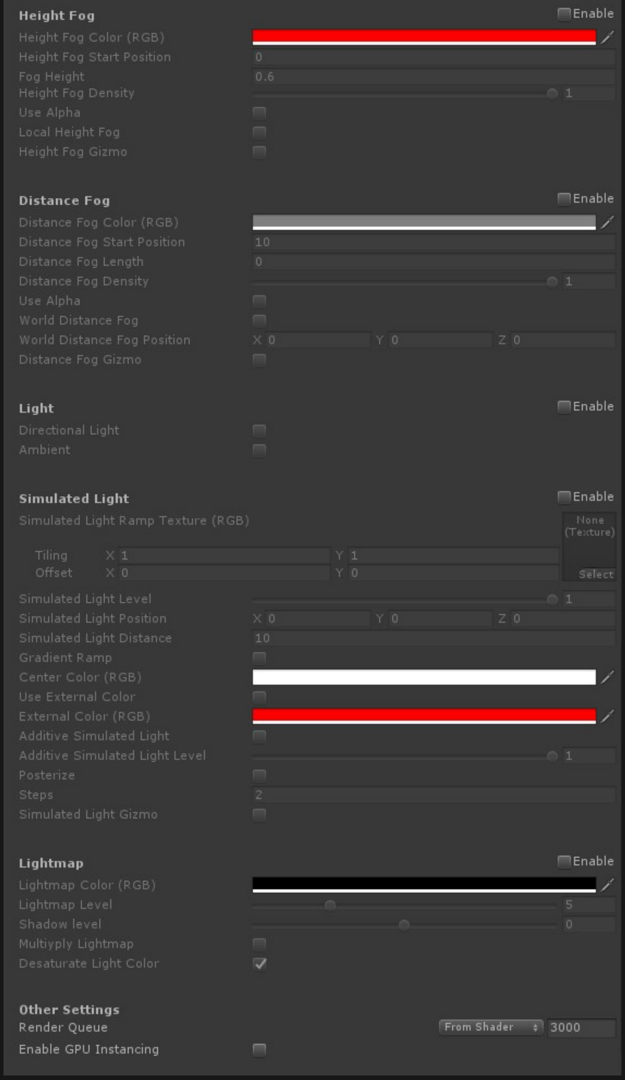
Rim Texture (RGBA): None (Texture) — Select
Tiling X 1 Y 1
Offset X 0 Y 0

Use Alpha Clip ☐
Alpha Cutoff: 0.5

Main Texture Level: 1
Front Texture Level: 1
Right Texture Level: 1
Top Texture Level: 1
Rim Texture Level: 1

Color Gradient ☐
Top Color (RGBA)
Gradient Center: 0.5
Gradient Width: 0.5
Gradient Revert ☐
Gradient Change Direction ☑
Gradient Rotation: 0

View direction ☐



Height Fog	□ Enable
Height Fog Color (RGB)	🟥
Height Fog Start Position	0
Fog Height	0.6
Height Fog Density	1
Use Alpha	□
Local Height Fog	□
Height Fog Gizmo	□
Distance Fog	□ Enable
Distance Fog Color (RGB)	▨
Distance Fog Start Position	10
Distance Fog Length	0
Distance Fog Density	1
Use Alpha	□
World Distance Fog	□
World Distance Fog Position	X 0 Y 0 Z 0
Distance Fog Gizmo	□
Light	□ Enable
Directional Light	□
Ambient	□
Simulated Light	□ Enable
Simulated Light Ramp Texture (RGB)	None (Texture) Select
Tiling	X 1 Y 1
Offset	X 0 Y 0
Simulated Light Level	1
Simulated Light Position	X 0 Y 0 Z 0
Simulated Light Distance	10
Gradient Ramp	□
Center Color (RGB)	⬜
Use External Color	□
External Color (RGB)	🟥
Additive Simulated Light	□
Additive Simulated Light Level	1
Posterize	□
Steps	2
Simulated Light Gizmo	□
Lightmap	□ Enable
Lightmap Color (RGB)	⬛
Lightmap Level	5
Shadow level	0
Multiyply Lightmap	□
Desaturate Light Color	☑
Other Settings	
Render Queue	From Shader 3000
Enable GPU Instancing	□

Blend Mode – Manages de blending mode.

- **Normal –** Without blending mode.
- **Darken –** Looks at the color information in each channel and selects the base or blend color—whichever is darker—as the result color. Pixels lighter than the blend color are replaced, and pixels darker than the blend color do not change.
- **Multiply –** Looks at the color information in each channel and multiplies the base color by the blend color. The result color is always a darker color. Multiplying any color with black produces black. Multiplying any color with white leaves the color unchanged. When you're painting with a color other than black or white, successive strokes with a painting tool produce progressively darker colors. The effect is similar to drawing on the image with multiple marking pens.
- **Color Burn –** Looks at the color information in each channel and darkens the base color to reflect the blend color by increasing the contrast between the two. Blending with white produces no change.
- **Linear Burn –** Looks at the color information in each channel and darkens the base color to reflect the blend color by decreasing the brightness. Blending with white produces no change.
- **Darker Color –** Compares the total of all channel values for the blend and base color and displays the lower value color. Darker Color does not produce a third color, which can result from the Darken blend, because it chooses the lowest channel values from both the base and the blend color to create the result color.
- **Lighten –** Looks at the color information in each channel and selects the base or blend color—whichever is lighter—as the result color. Pixels darker than the blend color are replaced, and pixels lighter than the blend color do not change.
- **Screen –** Looks at each channel's color information and multiplies the inverse of the blend and base colors. The result color is always a lighter color. Screening with black leaves the color unchanged. Screening with white produces white. The effect is similar to projecting multiple photographic slides on top of each other.
- **Color Dodge –** Looks at the color information in each channel and brightens the base color to reflect the blend color by decreasing contrast between the two. Blending with black produces no change.
- **Linear Dodge Or Additive –** Looks at the color information in each channel and brightens the base color to reflect the blend color by increasing the brightness. Blending with black produces no change.
- **Lighter Color –** Compares the total of all channel values for the blend and base color and displays the higher value color. Lighter Color does not produce a third color, which can result from the Lighten blend, because it chooses the highest channel values from both the base and blend color to create the result color.
- **Overlay –** Multiplies or screens the colors, depending on the base color. Patterns or colors overlay the existing pixels while preserving the highlights and shadows of the base color. The base color is not replaced, but mixed with the blend color to reflect the lightness or darkness of the original color.
- **Soft Light –** Darkens or lightens the colors, depending on the blend color. The effect is similar to shining a diffused spotlight on the image. If the blend color (light source) is lighter than 50% gray, the image is lightened as if it were dodged. If the blend color is darker than 50% gray, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.
- **Hard Light –** Multiplies or screens the colors, depending on the blend color. The effect is similar to shining a harsh spotlight on the image. If the blend color (light source) is lighter than 50% gray, the image is lightened, as if it were screened. This is useful for adding highlights to an image. If the blend color is darker than 50% gray, the image is darkened, as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.
- **Vivid Light –** Burns or dodges the colors by increasing or decreasing the contrast, depending on the blend color. If the blend color (light source) is lighter than 50% gray, the image is lightened by decreasing the contrast. If the blend color is darker than 50% gray, the image is darkened by increasing the contrast.
- **Linear Light –** Burns or dodges the colors by decreasing or increasing the brightness, depending on the blend color. If the blend color (light source) is lighter than 50% gray, the image is lightened by increasing the brightness. If the blend color is darker than 50% gray, the image is darkened by decreasing the brightness.
- **Pin Light –** Replaces the colors, depending on the blend color. If the blend color (light source) is lighter than 50% gray, pixels darker than the blend color are replaced, and pixels lighter than the blend color do not change. If the blend color is darker than 50% gray, pixels lighter than the blend color are replaced, and pixels darker than the blend color do not change. This is useful for adding special effects to an image.

- **Hard Mix-** Adds the red, green and blue channel values of the blend color to the RGB values of the base color. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if less than 255, a value of 0. Therefore, all blended pixels have red, green, and blue channel values of either 0 or 255. This changes all pixels to primary additive colors (red, green, or blue), white, or black.
- **Difference –** Looks at the color information in each channel and subtracts either the blend color from the base color or the base color from the blend color, depending on which has the greater brightness value. Blending with white inverts the base color values; blending with black produces no change.
- **Exclusion –** Creates an effect similar to but lower in contrast than the Difference mode. Blending with white inverts the base color values. Blending with black produces no change.
- **Subtract –** Looks at the color information in each channel and subtracts the blend color from the base color. In 8- and 16-bit images, any resulting negative values are clipped to zero.
- **Divide –** Looks at the color information in each channel and divides the blend color from the base color.
- **Hue –** Creates a result color with the luminance and saturation of the base color and the hue of the blend color.
- **Saturation –** Creates a result color with the luminance and hue of the base color and the saturation of the blend color. Painting with this mode in an area with no (0) saturation (gray) causes no change.
- **Color –** Creates a result color with the luminance of the base color and the hue and saturation of the blend color. This preserves the gray levels in the image and is useful for coloring monochrome images and for tinting color images.
- **Luminosity –** Creates a result color with the hue and saturation of the base color and the luminance of the blend color. This mode creates the inverse effect of Color mode.

Color – Color of the normals. Use the RGBA channels.

Front Light Level – Brightness of the light of the front normals.

Right Light Level – Brightness of the light of the right normals.

Top Light Level – Brightness of the light of the top normals.

Rim Light Level – Brightness of the light of the rim normals.

Front Opacity Level – Opacity of the color of the front normals.

Right Opacity Level – Opacity of the color of the right normals.

Top Opacity Level – Opacity of the color of the top normals.

Rim Opacity Level – Opacity of the color of the rim normals.

Main Texture – Texture of all normals. Use the RGBA channels.

Front Texture – Texture of the front normals. Use the RGBA channels.

Right Texture – Texture of the right normals. Use the RGBA channels.

Top Texture – Texture of the top normals. Use the RGBA channels.

Rim Texture – Texture of the rim normals. Use the RGBA channels.

Use Alpha Clip – Enables Alpha Clip.

Alpha cutoff – Alpha Cutoff value. Enabled if the selected rendering mode is Cutout.

Main Texture Level – Level of main texture in relation the source color.

Front Texture Level – Level of front texture in relation the source color.

Right Texture Level – Level of right texture in relation the source color.

Top Texture Level – Level of top texture in relation the source color.

Rim Texture Level – Level of rim texture in relation the source color.

Color Gradient – Color gradient.

Top Color – Color of the gradient. Use the RGBA channels.

Gradient Center – Gradient center.

Gradient Width – Gradient width.

Gradient Revert – Revert the ortientation of the gradient.

Gradient Change Direction – Change direction of the gradient.

Gradient Rotation – Gradient rotation.

View Mode – If enabled the color is applied based on the view direction.

Height Fog – Fog by vertex height.

Height Fog Color – Color of the fog.

Height Fog Start Position – Start point of the fog.

Fog Height – Height of the fog.

Height Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

Local Height Fog – If enabled the fog is created based on the center of the mesh.

Height Fog Gizmo – If enabled show height fog gizmo.

Distance Fog – Fog by camera distance.

Distance Fog Color – Color of the fog.

Distance Fog Start Position – Start point of the fog.

Distance Length – Length of the fog.

Distance Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn’t affect the transparent parts of the source color.

World Distance Fog – If enabled the fog is created based on a position of the world.

World Distance Fog Position – Posición en el mundo de la niebla.

Distance Fog Gizmo – World position of the distance fog.

Light – Light and Ambient light.

Directional Light – If enabled main directional light affect to the source mesh.

Ambient – If enabled ambient light affect to the source mesh.

Simulated Light – Simulated Light.

Simulated Light Ramp Texture – Color ramp of the simulated light based on a texture. The top part of the texture is the center of the simulated light and the bottom part is the external part of the simulated light.

Simulated Light Level – Level of simulated light in relation the source color.

Simulated Light Position – World position of the simulated light.

Simulated Light Distance – Simulated light circunference diameter.

Gradient Ramp – If enabled uses a gradient ramp between two colors instead a ramp texture.

Center Color – Color of the center of the simulated light if gradient ramp is enabled.

Use External Color – If enabled uses a color for the external part of the light instead de source color.

External Color – Color of the expernal part of the simulated light if gradient ramp is enabled.

Additive Simulated Light – If enabled adds the simulated light color to the source color.

Additive Simulated Light Level – Level of simulated light addition in relation the source color.

Posterize – If enabled converts the ramp texture or the gradient ramp to multiple regions of fewer tones.

Steps – Color steps of the posterization.

Simulated Light Gizmo – If enabled show simulated light gizmo.

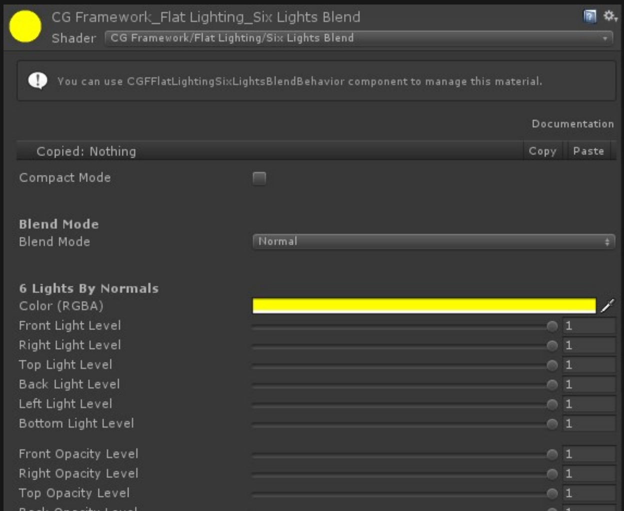
Use

Create a material with this shader and assign the material to the Material property of the MeshRenderer component.

SIX LIGHTS BLEND

Files: SixLightsBlend.shader
Path: “CGF/Shaders/FlatLighting”
Shader menu: “CG Framework/Flat Lighting/Six Lights Blend”

Shader that applies six lights based on the normals of the mesh with a blend mode.



CG Framework_Flat Lighting_Six Lights Blend
Shader: CG Framework/Flat Lighting/Six Lights Blend

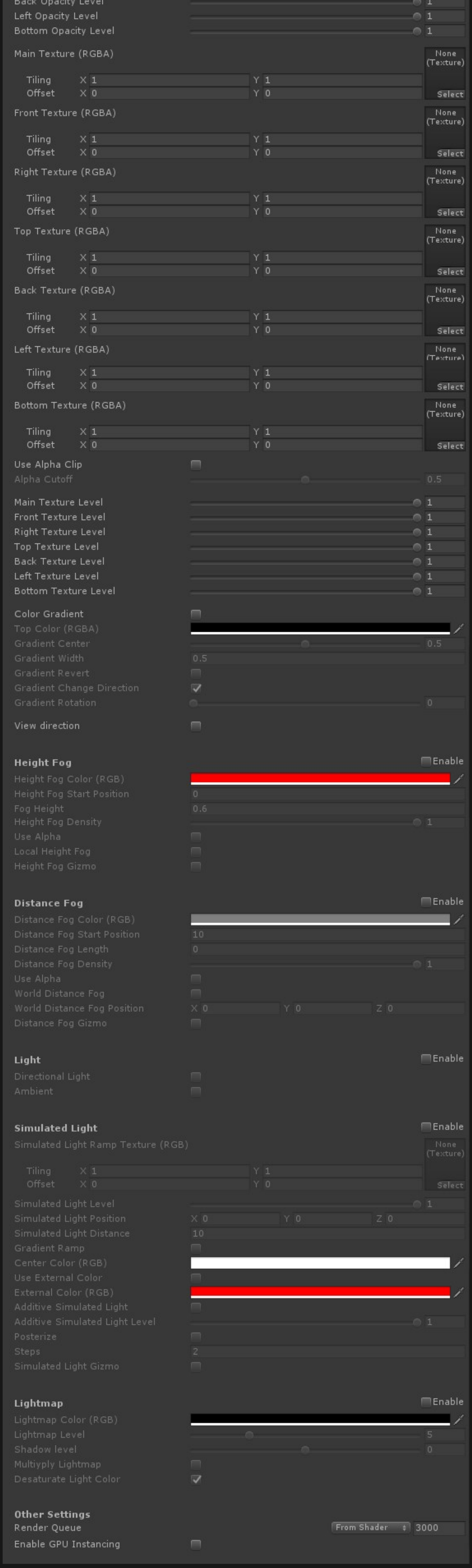
You can use CGFFlatLightingSixLightsBlendBehavior component to manage this material.

Compact Mode ☐

Blend Mode
Blend Mode: Normal

6 Lights By Normals
Color (RGBA)
Front Light Level 1
Right Light Level 1
Top Light Level 1
Back Light Level 1
Left Light Level 1
Bottom Light Level 1

Front Opacity Level 1
Right Opacity Level 1
Top Opacity Level 1



Back Opacity Level 1
Left Opacity Level 1
Bottom Opacity Level 1

Main Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Front Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Right Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Top Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Back Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Left Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Bottom Texture (RGBA) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0

Use Alpha Clip ☐
Alpha Cutoff 0.5

Main Texture Level 1
Front Texture Level 1
Right Texture Level 1
Top Texture Level 1
Back Texture Level 1
Left Texture Level 1
Bottom Texture Level 1

Color Gradient ☐
Top Color (RGBA)
Gradient Center 0.5
Gradient Width 0.5
Gradient Revert ☐
Gradient Change Direction ☑
Gradient Rotation 0

View direction ☐

Height Fog ☐ Enable
Height Fog Color (RGB)
Height Fog Start Position 0
Fog Height 0.6
Height Fog Density 1
Use Alpha ☐
Local Height Fog ☐
Height Fog Gizmo ☐

Distance Fog ☐ Enable
Distance Fog Color (RGB)
Distance Fog Start Position 10
Distance Fog Length 0
Distance Fog Density 1
Use Alpha ☐
World Distance Fog ☐
World Distance Fog Position X 0 Y 0 Z 0
Distance Fog Gizmo ☐

Light ☐ Enable
Directional Light ☐
Ambient ☐

Simulated Light ☐ Enable
Simulated Light Ramp Texture (RGB) None (Texture) Select
Tiling X 1 Y 1
Offset X 0 Y 0
Simulated Light Level 1
Simulated Light Position X 0 Y 0 Z 0
Simulated Light Distance 10
Gradient Ramp ☐
Center Color (RGB)
Use External Color ☐
External Color (RGB)
Additive Simulated Light ☐
Additive Simulated Light Level 1
Posterize ☐
Steps 2
Simulated Light Gizmo ☐

Lightmap ☐ Enable
Lightmap Color (RGB)
Lightmap Level 5
Shadow level 0
Multiyply Lightmap ☐
Desaturate Light Color ☑

Other Settings
Render Queue From Shader 3000
Enable GPU Instancing ☐

Blend Mode – Manages de blending mode.

- **Normal –** Without blending mode.
- **Darken –** Looks at the color information in each channel and selects the base or blend color—whichever is darker—as the result color. Pixels lighter than the blend color are replaced, and pixels darker than the blend color do not change.

- **Multiply –** Looks at the color information in each channel and multiplies the base color by the blend color. The result color is always a darker color. Multiplying any color with black produces black. Multiplying any color with white leaves the color unchanged. When you're painting with a color other than black or white, successive strokes with a painting tool produce progressively darker colors. The effect is similar to drawing on the image with multiple marking pens.
- **Color Burn –** Looks at the color information in each channel and darkens the base color to reflect the blend color by increasing the contrast between the two. Blending with white produces no change.
- **Linear Burn –** Looks at the color information in each channel and darkens the base color to reflect the blend color by decreasing the brightness. Blending with white produces no change.
- **Darker Color –** Compares the total of all channel values for the blend and base color and displays the lower value color. Darker Color does not produce a third color, which can result from the Darken blend, because it chooses the lowest channel values from both the base and the blend color to create the result color.
- **Lighten –** Looks at the color information in each channel and selects the base or blend color—whichever is lighter—as the result color. Pixels darker than the blend color are replaced, and pixels lighter than the blend color do not change.
- **Screen –** Looks at each channel's color information and multiplies the inverse of the blend and base colors. The result color is always a lighter color. Screening with black leaves the color unchanged. Screening with white produces white. The effect is similar to projecting multiple photographic slides on top of each other.
- **Color Dodge –** Looks at the color information in each channel and brightens the base color to reflect the blend color by decreasing contrast between the two. Blending with black produces no change.
- **Linear Dodge Or Additive –** Looks at the color information in each channel and brightens the base color to reflect the blend color by increasing the brightness. Blending with black produces no change.
- **Lighter Color –** Compares the total of all channel values for the blend and base color and displays the higher value color. Lighter Color does not produce a third color, which can result from the Lighten blend, because it chooses the highest channel values from both the base and blend color to create the result color.
- **Overlay –** Multiplies or screens the colors, depending on the base color. Patterns or colors overlay the existing pixels while preserving the highlights and shadows of the base color. The base color is not replaced, but mixed with the blend color to reflect the lightness or darkness of the original color.
- **Soft Light –** Darkens or lightens the colors, depending on the blend color. The effect is similar to shining a diffused spotlight on the image. If the blend color (light source) is lighter than 50% gray, the image is lightened as if it were dodged. If the blend color is darker than 50% gray, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.
- **Hard Light –** Multiplies or screens the colors, depending on the blend color. The effect is similar to shining a harsh spotlight on the image. If the blend color (light source) is lighter than 50% gray, the image is lightened, as if it were screened. This is useful for adding highlights to an image. If the blend color is darker than 50% gray, the image is darkened, as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.
- **Vivid Light –** Burns or dodges the colors by increasing or decreasing the contrast, depending on the blend color. If the blend color (light source) is lighter than 50% gray, the image is lightened by decreasing the contrast. If the blend color is darker than 50% gray, the image is darkened by increasing the contrast.
- **Linear Light –** Burns or dodges the colors by decreasing or increasing the brightness, depending on the blend color. If the blend color (light source) is lighter than 50% gray, the image is lightened by increasing the brightness. If the blend color is darker than 50% gray, the image is darkened by decreasing the brightness.
- **Pin Light –** Replaces the colors, depending on the blend color. If the blend color (light source) is lighter than 50% gray, pixels darker than the blend color are replaced, and pixels lighter than the blend color do not change. If the blend color is darker than 50% gray, pixels lighter than the blend color are replaced, and pixels darker than the blend color do not change. This is useful for adding special effects to an image.
- **Hard Mix-** Adds the red, green and blue channel values of the blend color to the RGB values of the base color. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if less than 255, a value of 0. Therefore, all blended pixels have red, green, and blue channel values of either 0 or 255. This changes all pixels to primary additive colors (red, green, or blue), white, or black.
- **Difference –** Looks at the color information in each channel and subtracts either the blend color from the base color or the base color from the blend color, depending on which has the greater brightness value. Blending with white inverts the base color values; blending with black produces no change.
- **Exclusion –** Creates an effect similar to but lower in contrast than the Difference mode. Blending with white inverts the base color values. Blending with black produces no change.
- **Subtract –** Looks at the color information in each channel and subtracts the blend color from the base color. In 8- and 16-bit images, any resulting negative values are clipped to zero.
- **Divide –** Looks at the color information in each channel and divides the blend color from the base color.
- **Hue –** Creates a result color with the luminance and saturation of the base color and the hue of the blend color.
- **Saturation –** Creates a result color with the luminance and hue of the base color and the saturation of the blend color. Painting with this mode in an area with no (0) saturation (gray) causes no change.
- **Color –** Creates a result color with the luminance of the base color and the hue and saturation of the blend color. This preserves the gray levels in the image and is useful for coloring monochrome images and for tinting color images.
- **Luminosity –** Creates a result color with the hue and saturation of the base color and the luminance of the blend color. This mode creates the inverse effect of Color mode.

Color – Color of the normals. Use the RGBA channels.

Front Light Level – Brightness of the light of the front normals.

Right Light Level – Brightness of the light of the right normals.

Top Light Level – Brightness of the light of the top normals.

Back Light Level – Brightness of the light of the back normals.

Left Light Level – Brightness of the light of the left normals.

Bottom Light Level – Brightness of the light of the bottom normals.

Front Opacity Level – Opacity of the color of the front normals.

Right Opacity Level – Opacity of the color of the right normals.

Top Opacity Level – Opacity of the color of the top normals.

Back Opacity Level – Opacity of the color of the back normals.

Left Opacity Level – Opacity of the color of the left normals.

Bottom Opacity Level – Opacity of the color of the bottom normals.

Main Texture – Texture of all normals. Use the RGBA channels.

Front Texture – Texture of the front normals. Use the RGBA channels.

Right Texture – Texture of the right normals. Use the RGBA channels.

Top Texture – Texture of the top normals. Use the RGBA channels.

Back Texture – Texture of the back normals. Use the RGBA channels.

Left Texture – Texture of the left normals. Use the RGBA channels.

Bottom Texture – Texture of the bottom normals. Use the RGBA channels.

Use Alpha Clip – Enables Alpha Clip.

Alpha cutoff – Alpha Cutoff value. Enabled if the selected rendering mode is Cutout.

Main Texture Level – Level of main texture in relation the source color.

Front Texture Level – Level of front texture in relation the source color.

Right Texture Level – Level of right texture in relation the source color.

Top Texture Level – Level of top texture in relation the source color.

Back Texture Level – Level of back texture in relation the source color.

Left Texture Level – Level of left texture in relation the source color.

Bottom Texture Level – Level of bottom texture in relation the source color.

Color Gradient – Color gradient.

Top Color – Color of the gradient. Use the RGBA channels.

Gradient Center – Gradient center.

Gradient Width – Gradient width.

Gradient Revert – Revert the ortientation of the gradient.

Gradient Change Direction – Change direction of the gradient.

Gradient Rotation – Gradient rotation.

View Mode – If enabled the color is applied based on the view direction.

Height Fog – Fog by vertex height.

Height Fog Color – Color of the fog.

Height Fog Start Position – Start point of the fog.

Fog Height – Height of the fog.

Height Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

Local Height Fog – If enabled the fog is created based on the center of the mesh.

Height Fog Gizmo – If enabled show height fog gizmo.

Distance Fog – Fog by camera distance.

Distance Fog Color – Color of the fog.

Distance Fog Start Position – Start point of the fog.

Distance Length – Length of the fog.

Distance Fog Density – Level of fog in relation the source color.

Use Alpha – If enabled fog doesn't affect the transparent parts of the source color.

World Distance Fog – If enabled the fog is created based on a position of the world.

World Distance Fog Position – Posición en el mundo de la niebla.

Distance Fog Gizmo – World position of the distance fog.

Light – Light and Ambient light.

Directional Light – If enabled main directional light affect to the source mesh.

Ambient – If enabled ambient light affect to the source mesh.

Simulated Light – Simulated Light.

Simulated Light Ramp Texture – Color ramp of the simulated light based on a texture. The top part of the texture is the center of the simulated light and the bottom part is the external part of the simulated light.

Simulated Light Level – Level of simulated light in relation the source color.

Simulated Light Position – World position of the simulated light.

Simulated Light Distance – Simulated light circunference diameter.

Gradient Ramp – If enabled uses a gradient ramp between two colors instead a ramp texture.

Center Color – Color of the center of the simulated light if gradient ramp is enabled.

Use External Color – If enabled uses a color for the external part of the light instead de source color.

External Color – Color of the expernal part of the simulated light if gradient ramp is enabled.

Additive Simulated Light – If enabled adds the simulated light color to the source color.

Additive Simulated Light Level – Level of simulated light addition in relation the source color.

Posterize – If enabled converts the ramp texture or the gradient ramp to multiple regions of fewer tones.

Steps – Color steps of the posterization.

Simulated Light Gizmo – If enabled show simulated light gizmo.

Use

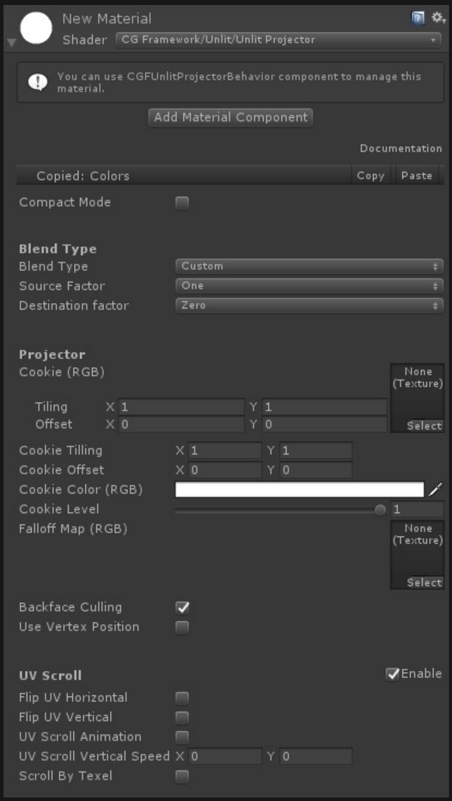
Create a material with this shader and assign the material to the Material property of the MeshRenderer component.

UNLIT

PROJECTOR

Files: Projector.shader
Path: "CGF/Shaders/Unlit"
Shader menu: "CG Framework/Unlit/Projector"

Projector shader.



New Material
Shader: CG Framework/Unlit/Unlit Projector

You can use CGFUnlitProjectorBehavior component to manage this material.

Add Material Component

Documentation

Copied: Colors — Copy — Paste

Compact Mode □

Blend Type
Blend Type: Custom
Source Factor: One
Destination factor: Zero

Projector
Cookie (RGB): None (Texture) — Select
Tiling X 1 Y 1
Offset X 0 Y 0
Cookie Tilling X 1 Y 1
Cookie Offset X 0 Y 0
Cookie Color (RGB)
Cookie Level: 1
Falloff Map (RGB): None (Texture) — Select

Backface Culling ☑
Use Vertex Position □

UV Scroll ☑ Enable
Flip UV Horizontal □
Flip UV Vertical □
UV Scroll Animation □
UV Scroll Vertical Speed X 0 Y 0
Scroll By Texel □

Blend Type – Manages the blending mode.

- **Custom**
- **Alpha Blend**
- **Premultiplied**
- **Additive**
- **Soft Additive**
- **Multiplicative**
- **Double Multiplicative**
- **Particle Additive**
- **Particle Blend**

Source Factor – Manages the blending factor from source. Source refers to the calculated color, Destination is the color already on the screen.

- **Zero** – Blend factor is (0, 0, 0, 0).

- **One** – Blend factor is (1, 1, 1, 1).
- **Dst Color** – Blend factor is (Red destination, Green destination, Blue destination, Alpha destination).
- **Src Color** – Blend factor is (Red source, Green source, Blue source, Alpha source).
- **One Minus Dst Color** – Blend factor is (1 – Red destination, 1 – Green destination, 1 – Blue destination, 1 – Alpha destination).
- **Src Alpha** – Blend factor is (Alpha source, Alpha source, Alpha source, Alpha source).
- **One Minus Src Color** – Blend factor is (1 – Red source, 1 – Green source, 1 – Blue source, 1 – Alpha source).
- **Dst Alpha** – Blend factor is (Alpha destination, Alpha destination, Alpha destination, Alpha destination).
- **One Minus Dst Alpha** – Blend factor is (1 – Alpha destination, 1 – Alpha destination, 1 – Alpha destination, 1 – Alpha destination).
- **Src Alpha saturate** – Blend factor is (f, f, f, 1); where f = min(Alpha source, 1 – Alpha destination).
- **One Minus Src Alpha** – Blend factor is (1 – Alpha source, 1 – Alpha source, 1 – Alpha source, 1 – Alpha source).

Destination Factor – Manages the blending factor from destination. Source refers to the calculated color, Destination is the color already on the screen.

- **Zero** – Blend factor is (0, 0, 0, 0).
- **One** – Blend factor is (1, 1, 1, 1).
- **Dst Color** – Blend factor is (Red destination, Green destination, Blue destination, Alpha destination).
- **Src Color** – Blend factor is (Red source, Green source, Blue source, Alpha source).
- **One Minus Dst Color** – Blend factor is (1 – Red destination, 1 – Green destination, 1 – Blue destination, 1 – Alpha destination).
- **Src Alpha** – Blend factor is (Alpha source, Alpha source, Alpha source, Alpha source).
- **One Minus Src Color** – Blend factor is (1 – Red source, 1 – Green source, 1 – Blue source, 1 – Alpha source).
- **Dst Alpha** – Blend factor is (Alpha destination, Alpha destination, Alpha destination, Alpha destination).
- **One Minus Dst Alpha** – Blend factor is (1 – Alpha destination, 1 – Alpha destination, 1 – Alpha destination, 1 – Alpha destination).
- **Src Alpha saturate** – Blend factor is (f, f, f, 1); where f = min(Alpha source, 1 – Alpha destination).
- **One Minus Src Alpha** – Blend factor is (1 – Alpha source, 1 – Alpha source, 1 – Alpha source, 1 – Alpha source).

Cookie – Projection texture. Only uses the RGB channels.

Cookie Tilling – Scale of the UV of the texture of the projection.

Cookie Offset – Offset of the UV of the texture of the projection.

Cookie Color – Color of the projection. Only uses the RGB channels.

Cookie Level – Projection level intensity.

Falloff Map – Texture that determines the fading of the projection along its trajectory, it is a linear gradient texture. Only uses the RGB channels.

Backface Culling – If enabled cull the projection on the backfaces of the mesh.

Use vertex position – “If enabled use the vertex position instead the vertex normal to cull the backfaces of the mesh.

UV Scroll – Scroll and Flip the UVs from a texture.

Flip UV Horizontal – Flip UV Horizontal.

Flip UV Vertical – Flip UV Vertical.

UV Scroll Animation – If enabled the UV they animated.

UV Scroll Vertical Speed – UV Scroll Speed.

Scroll By Texel – Scroll animation texel by texel.

Use

Create a material with this shader and assign the material to the Material property of the Projector component.

+34 93 125 29 73
info@chloroplastgames.com

Pelai, 1 Local 5
Castelldefels, Barcelona,
08860

PRESSKIT