**烘培（Lightingmap）技术规范**



**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| 1.0 | 06/27/2016 | HuaCheng | Document created |



[烘培（Lightingmap）技术规范 1](#_Toc454891037)

[Version Control 1](#_Toc454891038)

[1 技术概述 2](#_Toc454891039)

[1.1 烘培概念 2](#_Toc454891040)

[1.2 烘培的意义 2](#_Toc454891041)

[1.3 烘培方式 2](#_Toc454891042)

[2 烘培适用规范 2](#_Toc454891043)

[2.1 适用原则 2](#_Toc454891044)

[2.2 技术替代方案 2](#_Toc454891045)

[3 光照贴图大小规范 3](#_Toc454891046)

[3.1 建议大小 3](#_Toc454891047)

[3.2 修改大小 3](#_Toc454891048)

[4 光照贴图加载规范 3](#_Toc454891049)

[4.1 动态加载资源的问题 3](#_Toc454891050)

[4.2 建议加载方式 3](#_Toc454891051)

[5 烘培流程 3](#_Toc454891052)

[5.1 烘培前设置 3](#_Toc454891053)

[5.1.1 UV设置 3](#_Toc454891054)

[5.1.2 模型设置 3](#_Toc454891055)

[5.2 烘培 4](#_Toc454891056)

[5.2.1 烘培参数设置 4](#_Toc454891057)

[5.2.2 建议烘焙参数 5](#_Toc454891058)

[5.2.3 开始烘培 5](#_Toc454891059)

# 



# 技术概述

## 烘培概念

烘焙简单地说就是一种将灯光信息烘焙到模型表面，以贴图的形式进行存储的技术。烘培生成的贴图称为LightingMap即光照贴图。

## 烘培的意义

在游戏中想要模拟一个逼真的光照环境需要很多不同的灯光相互配合，实时光影都是由显卡和CPU来计算的，需要消耗大量的系统资源。如果采用烘培技术，这些光影效果都记录到贴图上，运行的时候，显卡和CPU不需要再对光照效果进行运算，极大节约系统资源。

## 烘培方式

有以下两种方式：

 1．在3dmax等模型制作软件中对场景进行烘焙。将烘焙好的模型以及贴图导入到Unity中。

 2．用Unity自身的烘焙系统对场景进行烘焙。

本文档仅讲述使用Unity自身的烘培系统的技术规范。



# 烘培适用规范

## 适用原则

对于关卡类的游戏，如果关卡数量较多，每个关卡生成一张光照贴图会照成游戏包体的急剧增加，不建议采用烘培方式。

对于ARPG类游戏，关卡数量有限，可以采用烘培的方式

## 技术替代方案

修改美术资源，将光影信息绘制在模型贴图上。



# 光照贴图大小规范

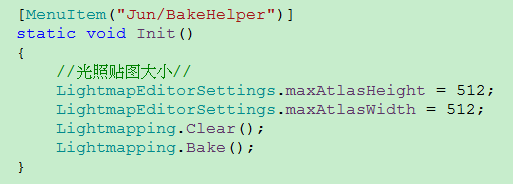
## 建议大小

建议每个场景使用一张512\*512像素的光照贴图。

## 修改大小

Unity默认生成的光照贴图是1024\*1024像素大小，对于手机游戏而言过大需要修改其大小。

提供一个C#脚本工具进行场景烘培，在脚本中设置光照贴图大小。





# 光照贴图加载规范

## 动态加载资源的问题

在Unity5.x中，prefab不保存模型的光照贴图，如果将模型分割出去进行动态加载，那么显示出来的模型并不带贴图效果。

## 建议加载方式

建议采用Application.LoadLevel接加载场景的方式。



# 烘培流程

## 烘培前设置

### UV设置

确保所要烘焙的模型有合适的lightmapping UV。如果想让Unity自动生成lightmapping UV，就在模型的导入设置中勾选 Generate Lightmap UVs。反之，如果需要保留美术设置的UV，则不要勾选 Generate Lightmap UVs。

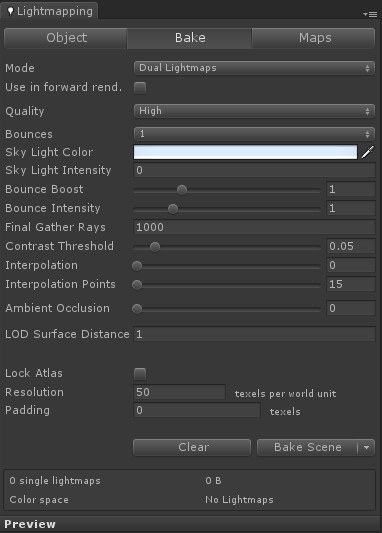
### 模型设置

任何Mesh Renderer, Skinned Mesh Renderer 或者 Terrain都要标注为static，或在LightingMapping中勾选lightmap static。

## 烘培

在Unity编辑器中点击Window>Lightingmapping可打开烘培设置菜单。

### 烘培参数设置



* Mode:

(1)Dual Lightmap mode:近景烘焙图(near lightmaps)和远景烘焙图(far lightmaps)都会被烘焙，只有deferred rendering path支持该模式。

(2)Single Lightmap mode:只有远景烘焙图(far lightmaps)会被烘焙，

(3)Directional Lightmap mode：

* Use in forward rendering:  
  一般可以忽略它。针对Dual lightmaps的设定,只在Mode选的是Dual时才会出现。在forward rendering模式下是否激活dual lightmaps,需要自己写对应的Shader。
* Quality：

High, low 2个选项。决定采集射线的数量，对比界限，以及反锯齿等设置。

* Bounces：

Global Illumination模拟中 light bounces的数量。0表示只有直接光照会被计算，如果要加入柔软的，真实的间接光照的话，则至少填1。

* Sky Light Color：

模拟天空从所有方向射来的光照。对室外场景很重要。

(注意：RenderSettings里的Ambient Light也参与烘焙，并且会覆盖Sky Light。所以使用Sky Light的话，则需要关闭Ambient Light,把Ambient调成（0,0,0,0）即可)

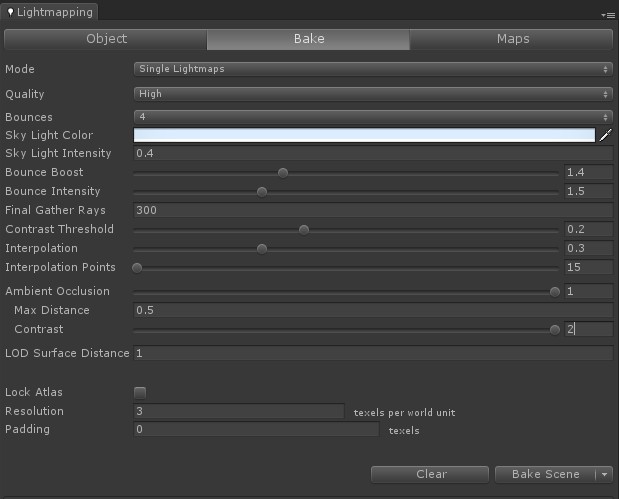
* Sky Light Intensity：Sky Light的强度。0表示关闭Sky Light.
* Bounce Boost：加强Bounced light也就是间接光照的效果.
* Bounce Intensity：间接光照的效果强度。
* Final Gather Rays:从所有收集的点发射的射线的数量，最高则效果最好。默认为1000
* Contrast Threshold: 越高表示光照越平滑同时细节越低
* Interpolation：颜色计算方式。0表示线性插值，1表示基于梯度的高级插值。有时1会产生人工光照的失真效果。
* Interpolation Points:越多表示光照越平滑同时细节越低。
* Ambient Occlusion：环境闭塞（全局闭塞）在烘焙中的量，和光照无关，仅仅基于距离的效果。
* Lock Atlas：锁上了，贴图就不会变化，没什么用。
* Resolution：

通常来说 Resolution设置平均每单位面积对应lightmap的多少色素。该值越高代表单位面积的色素越多即越精细。反之越低越粗糙。默认值为50。如果一个10x10的plane,resolution是50的话，在lightmap里该plane就占用500x500像素尺寸。  
当然也可以单调该数值。选中物体后，在lightmapping界面中会有Scale in lightmap的数值，表示的是该物体的烘焙信息在lightmap占的大小，1表示正常尺寸。0表示该物体没有lightmap效果，但会影响其他物体的lightmap运算。2表示正常尺寸的两倍大小。

* Padding:在lightmap里,不同物体的烘焙图的间距，单位为图素(texel)

### 建议烘焙参数

由于烘培参数较多，且涉及较多计算机图形学知识，现给出一个建议烘培参数。



### 开始烘培

上述参数设置完成后，点击Bake Scene 按钮，Unity会自动开始烘培。烘培后的贴图文件会生成在对应场景名命名的文件夹下。

