

# 梦境节点编辑器 V1.0

## 网格物体配置文件说明文档

## 目录

0. 目标读者
1. 说明
2. 网格物体导入说明(.obj)
3. 网格物体配置文件语法
4. 网格物体属性和渲染器类型
5. 小结
6. 常见问题

## 0. 目标读者

请在阅读本文当前有限阅读用户手册。

目标读者应当具有基本的三维物体创建经验以及贴图创建经验。该文档不会描述如何从三维软体中导出.obj文件的具体过程。如有需要请参考具体三维工具和绘图软件的说明文档。

若读者具有编程基础以及 GLSL 基础那么更好。

## 1. 介绍

网格物体配置文件（以下简称网格配置文件）的介绍在用户手册中已有说明，此处为重要概念：网格配置文件需要放在任一“Data”文件夹内，或者“Data”文件夹的子文件夹；从文件管理器中拖拽网格配置文件到应用程序界面的画布区域以载入。

网格配置文件用来指定需要载入的网格物体和贴图，以及定义网格物体的表面材质属性。在应用程序中，网格配置文件中的属性直接映射到具体的 GLSL 渲染器中。

要想定义一个完整的网格物体，共有三种数据需要提供：三维物体、贴图、以及对于三维网格物体其他属性的文字描述例如占据的空间大小和表面的反射强度。

## 2. 网格物体导入说明(.obj)

当从外部三维编辑软体中导入三维物体到梦境编辑器中时，以下几点需要设置：

- 模型需要被三角化
  - 模型需要具有 UV 坐标
  - 梦境编辑器使用右手坐标系，以 Y 轴为上，-Z 为前
  - 模型必须使用一个材质，并且模型场景文件中只应该含有一个模型
- 导入具体模型的案例教程可以向作者索要。

## 3. 网格物体配置文件语法

网格物体配置文件使用类似于 C/C++ 的语法来定义网格物体的属性。每一行代表一个属性，行与行之间可以使用“；”符号进行区分，但是并不必要。例如 ShapeMesh =

"Temp\Plane.obj"

“//”和“/\*”/“”可以用来提供注释。

有四种基本数据类型可以使用：boolean、integer、float、string。这四种数据类型的数据也可以使用“{}”进行创建。例如：PointLightColor = {1.0, 1.0, 1.0}

Boolean 值必须使用 true 或者 false 来制定，不可以使用数字 0 和 1，也不需要使用双引号。例如 PointShadow = true;

Floating 值必须含有小数点，否则算作 integer，例如 PointLightStrength = 1.0 或者 PointLightStrength = 1. 都是合法的 1.0 的表示。

对于每一个特定类型的渲染器，有一系列预制的属性可以使用，例如 PointLightColor 和 ShapeMesh。每个网格物体配置文件使用这些属性来完全描述一个物体节点。关于这些属性的细节请参见后文。

Strings 用于指定贴图文件的路径，这些路径都是相对于“Data”目录的，而不是相对于网格物体配置文件本身，例如"SimpleShapes\ParallaxTest.png"指向名为 ParallaxTest.png 的贴图，该贴图实际位置可以在 C:\ApplicationPath\Data\SimpleShapes 或者 C:\DocumentPath\Data\SimpleShapes

#### 4. 网格物体属性和渲染器类型

总共有三种渲染器内容可以与物体节点一起使用，每个具有不同的可用属性：Texture(2)、Lambert(3)和 SEM(4)。

网格物体配置文件格式：所有属性中除了基本信息外都可选；属性指定不必按照顺序  
基本信息

属性	数据类型	举例	使用方式	没有指定时的默认行为
Shader	integer	Shader = 2	Normally Meshes Use Simple Texture Shader(2) or Phong Shader (3) or SEM Shader(4); Certain Shaders Doesn't Need a Texture Coordinate; Once This Has Been Specified, Only Related Properties to That Shader Need to Be Specified, Unrelated Properties Won't Be Loaded.	Shape(1) with default shape parameters
ShapeMesh	string	ShapeMesh="File name"	The Name of The Actual Mesh File	MeshNode won't load properly
CollisionMesh	string	CollisionMesh="Filename"	The name of The Collision File, Can Be The Same As ShapeMesh; Assign This Property Only If Needed(If you don't know when you need this, then always assign this property)	A spherical collision will be generated for the mesh
Radius	float	Radius = 5.0	Unit in Object space(i.e. meters); Notice floats must have a floating point otherwise it won't get recognized, I know this sucks but bear with it	Default value 1 unit

### Shape Shader Properties

属性	数据类型	举例	使用方式	没有指定时的默认行为
Color	float*	Color = {0.1, 0.1, 0.1}		Default Red

### SimpleTexture Shader Properties

属性	数据类型	举例	使用方式	没有指定时的默认行为
ImageTexture	string	ImageTex = "Filename"		Default will use system Texture

### SEM Shader Properties

属性	数据类型	举例	使用方式	没有指定时的默认行为
EnvBaseTexture	string	EnvBaseTexture = "Filename"		Default Glossy Black
EnvTopTexture	string	EnvTopTexture = "Filename"		Default White
GlossTexture	string	GlossTexture = "Filename"		Default White

### Lambert Shader Properties

属性	数据类型	举例	使用方式	没有指定时的默认行为
GroundColor	float[3]	GroundColor={0.277, 0.624, 0.227}	Ground reflection onto the object	Default Black
PointLightColor	float*	PointLightColor = {0.1, 0.1, 0.1}	A light that facilitates lighting the mesh	Default White
PointLightLocation	float*	PointLightLocation = {1,2,3}	The location of the light relative to the object	Default {1,1,1}
DiffuseColor	float*	DiffuseColor = {0.1, 0.1, 0.1}	Only use if DiffuseTex is not specified; Describes the amount of diffuse light accepted by the object, the actual color depends on the lighting condition	Default Red
DiffuseTexture	string	DiffuseTex = "Filename"	If not specified, use diffuse color;	No Default

			If specified, diffuse color is ignored	
SpecularColor	float*	SpecularColor = {0.1, 0.1, 0.1}	Specifies the faked highlight reflection.	White is physically correct
SpecularPower	float	SpecularPower = 1	Specifies how specular the surface is. A bigger value will cause light reflections on the surface to concentrate on a smaller area; Range in [0, 1]	Default 0
SpecularTexture <Should be GreyScale> <Complex Usage>	string	SpecularTex = "Filename"	If not specified, use SpecularPower; If Specified, SpecularPower is ignored. When used it specifies the SpecularPower at each surface point. Only the Red channel of the texture is used to specify SpecularPower; Green channel is used for mesh transparency; Blue channel for height(Remember to invert black and white) map in Parallax mapping	No Default
EmissiveColor	float*	EmissiveColor = {0.1, 0.1, 0.1}	Only use if EmissiveTexture is not specified	Default Red
EmissiveTexture	string	EmissiveTexture = "Filename"	If not specified, use EmissiveColor; If specified, EmissiveColor is ignored	No Default
NormalTexture	string	NormalTexture = "Filename"	If not specified, not used; Used in conjunction with	Default Flat

			Height map in SpecularTexture for parallax mapping effects	
--	--	--	---	--

## **5. 小结**

讲一个三维模型导入到应用程序的流程为：首先创建一个三维模型并且创建相关的UV和三角化，导出为.Obj格式；然后创建一个相关的贴图文件（可选）；最后创建一个网格物体配置文件来描述物体节点的属性。把这些文件（一个.obj文件、多个.png/.jpg/.bmp文件、一个.spec文件）拷贝到任一“Data”目录中，然后拖拽.spec文件到应用程序界面画布区域以载入。

## **6. 常见问题**

- 如果载入一个物体之后物体只有阴影、或者表面颜色为黑色，请检查网格物体配置文件中贴图的路径是否正确：通常这是由于贴图文件名拼写错误因此应用程序无法正确载入贴图。