

TileMazeGen-Doc-CN

中文文档:

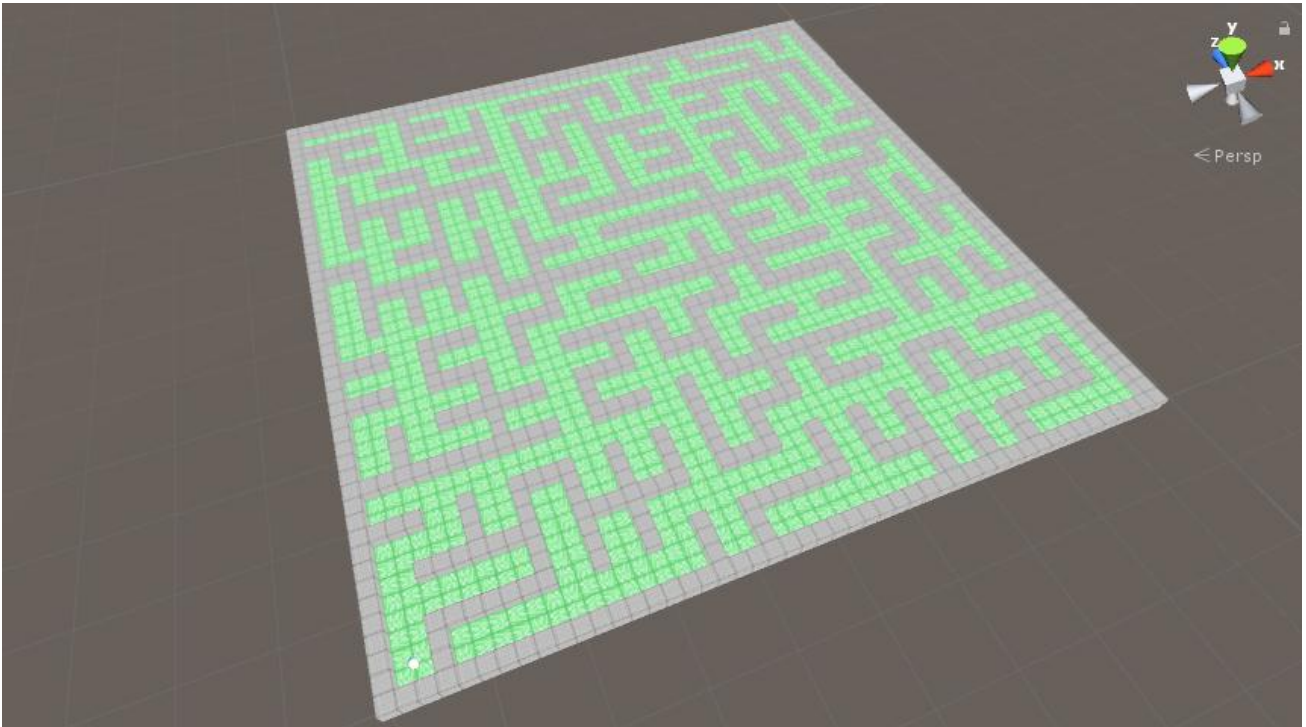
(1) 准备资源

(1.1) 简介

随机地图生成插件,是一个可以生成随机瓦块地图的Unity插件。他不光可以生成随机迷宫,还能生成基于Perlin噪音的随机地图。

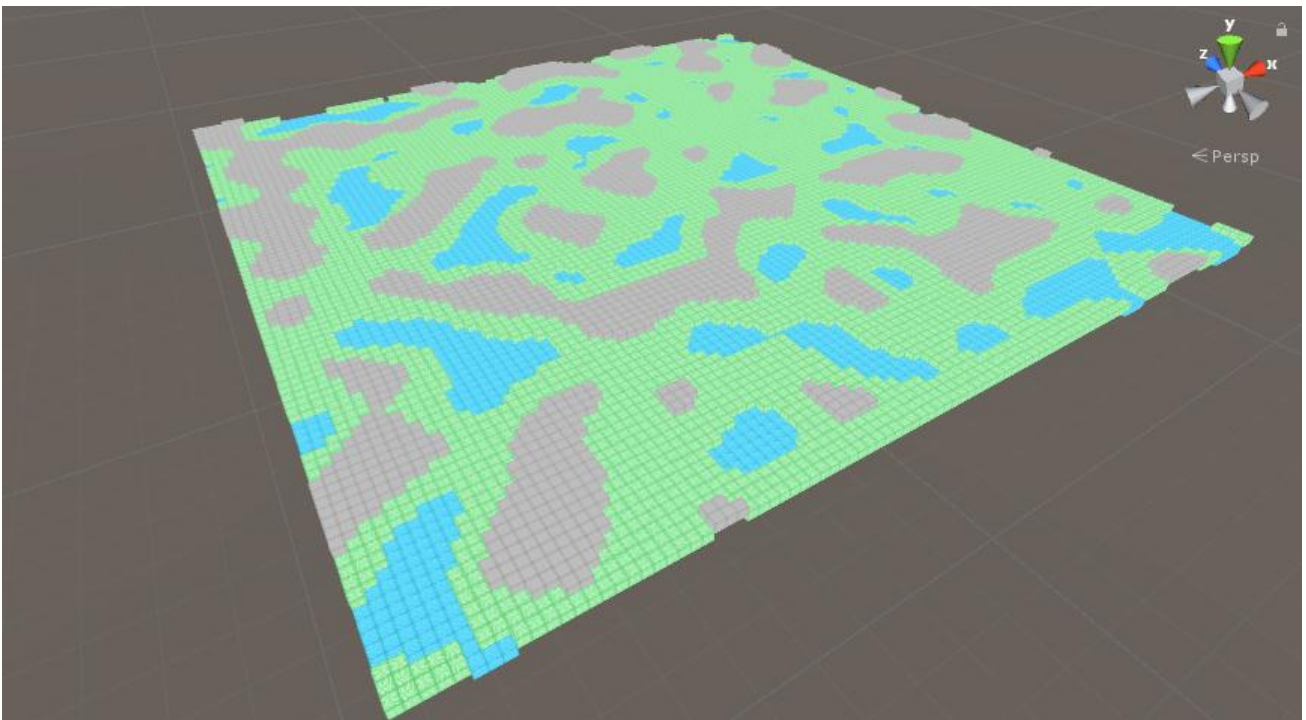
针对迷宫而言,他可以生成基于十二种算法的迷宫,每种迷宫都有自己的特色。你可以根据需求来做出选择。

这适合制作随机的地下城迷宫。



随机地图生成则主要依赖于Perlin噪音机制。给定相同的Size参数, OriginX和OriginY参数,总是能够得到相同的地图。

这适合制作迷宫的野外部分



如果你觉得我的地块不够漂亮, AssetStore有个不错的免费资源, 3D Cartoon Box Map, 和我的插件结合起来非常自然。当然你也可以轻松的建立自己的地块模型。

(1.2) 地块

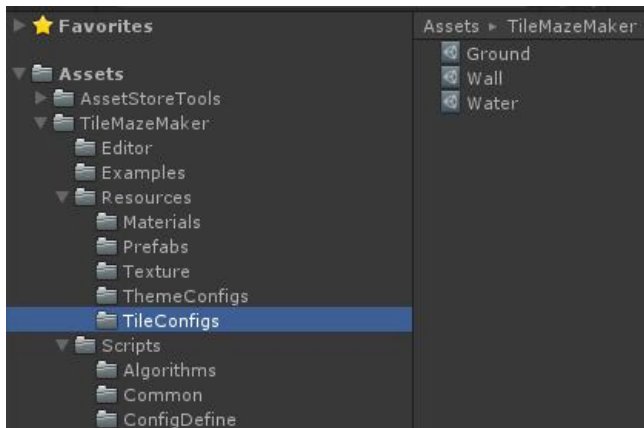
地块的规格,当前地块使用的是2*2*2的地块,当然这大小个是可配置的。你可以建立一个Cube,然后将它的Scale设置成2,2,2,最后为他分配一个合适的贴图。一个地块的3D模型资源就建好了。

如果你用3D模型软件建立立方体地块，并且将他们的贴图打包到一张贴图，则会额外多省下很多的Batch。

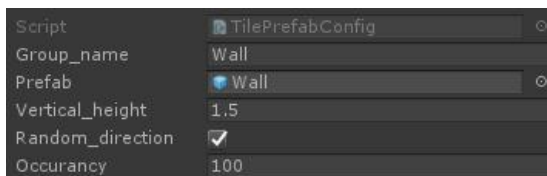
接下来你需要在菜单中选择

MazeGen-Tile Prefab Config 来创建一个地块配置文件。

他的默认位置在这里：



他看起来是这样的：



Group Name: 组名，可以把相同作用的模型，归为一组，比如墙壁组。

Prefab: 填入你刚刚创建的模型

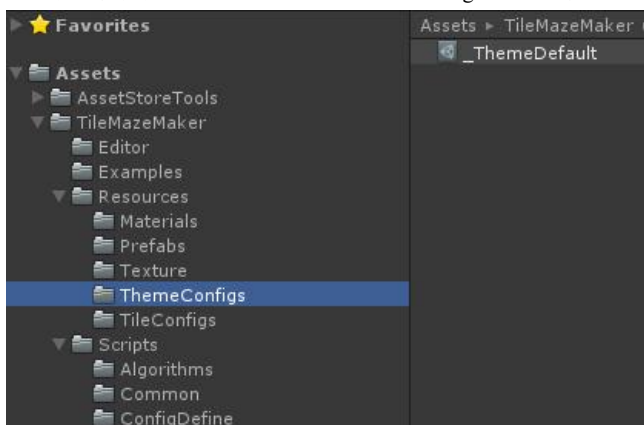
Vertical Height:相对高度便宜量

Random Direction: 地块是不是会出现四个方向上的偏移。

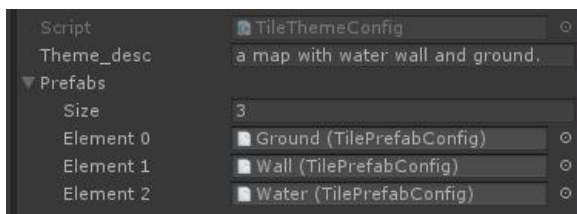
Occurance: 如果墙壁A的概率是200，墙壁B的概率是100，他们在同一个组中（拥有相同的组名）那么A出现的概率就是B的二倍。

（1.3）主题

请在菜单中选择：MazeGen-Tile Theme Config，生成一个主题配置文件，他的默认位置在这里



这个配置文件，是你配置一个主题的地方，比如沙漠主题，雪山主题，森林主题，地下城主题，每个主题可以保持为一个单独的文件，以备后用。



Theme_Desc: 写一个便于记忆的提示信息

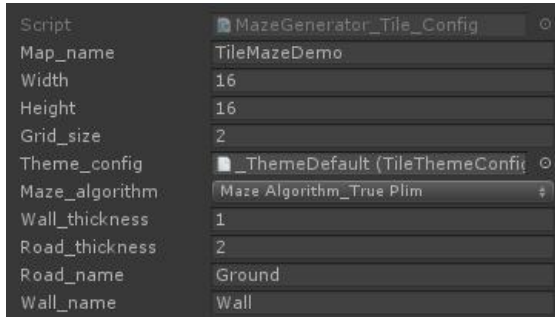
Prefabs: 设置Size，并且把这个主题所需的Tile配置文件拖进来，你不需要针对Group进行归类或排序。

这两个文件准备好之后，

（2）如何配置-随机迷宫

（2.1）配置

在菜单中选择MazeGen-MazeMapConfig，可以生成一个迷宫配置文件，他会出现在TileMazeMaker/Resources目录下面



MapName: 填写一个便于记忆的名字。

Width: 注意这个指的是迷宫的横向的逻辑格子数量，而不是Tile模型的数量

Height: 注意这个指的是迷宫的纵向的逻辑格子数量，而不是Tile模型的数量

GridSize: 如果你做的模型是3*3大小的，那么你需要修改这个值，以适配你的模型。

ThemeConfig: 将你之前配置好的主题配置文件拖进来。

MazeAlgorithm: 你可以在13种算法中选一个——其中一种算法是Aldour Border和Wilson算法的结合体，姑且把他也算作一种，但是需要注意的是

Aldour Border算法和Wilson算法的效率，在迷宫较大时，是很低的，所以建议使用的时候，使用AldourBorder-Wilson这个结合的算法。他能得到类似于前面两种算法的迷宫结构，并且效率会非常高。

WallThickness: 墙的厚度。

RoadThickness: 道路的厚度

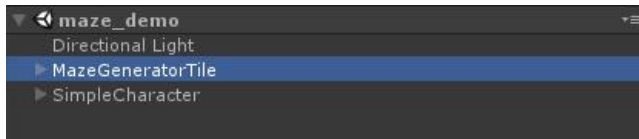
RoadName: 目前迷宫只支持两个组道路组，以及墙组，把他的名字，填写成Tile Prefab Config里面的组名字即可。

WallName: 墙壁组名，同上。

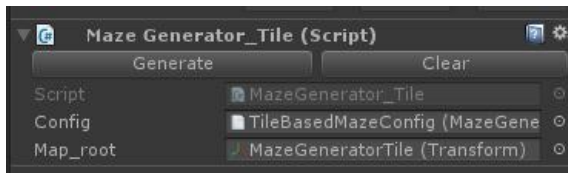
(2.2) 生成

在场景中建立一个空的GameObject，改个合适的名字，添加组件MazeGenerator_Tile。他看起来像是这样（你也可以参考例子：

maze_example）：



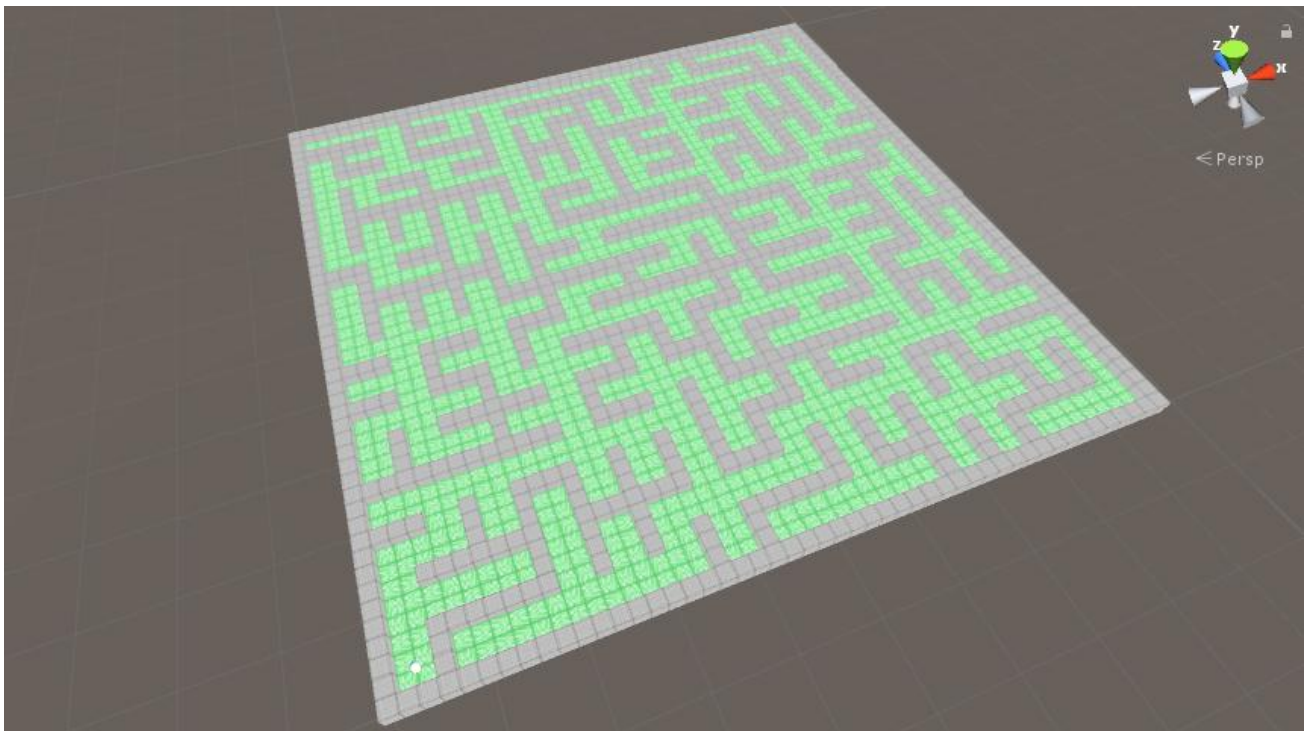
配置：



Config: 是(1.3)配置的配置文件

MapRoot: 是你希望的，迷宫的根节点，通常建议直接把管理器所在的GameObject拖进去。

点击Generate，可以生成一个，用指定算法生成的，随机的迷宫。

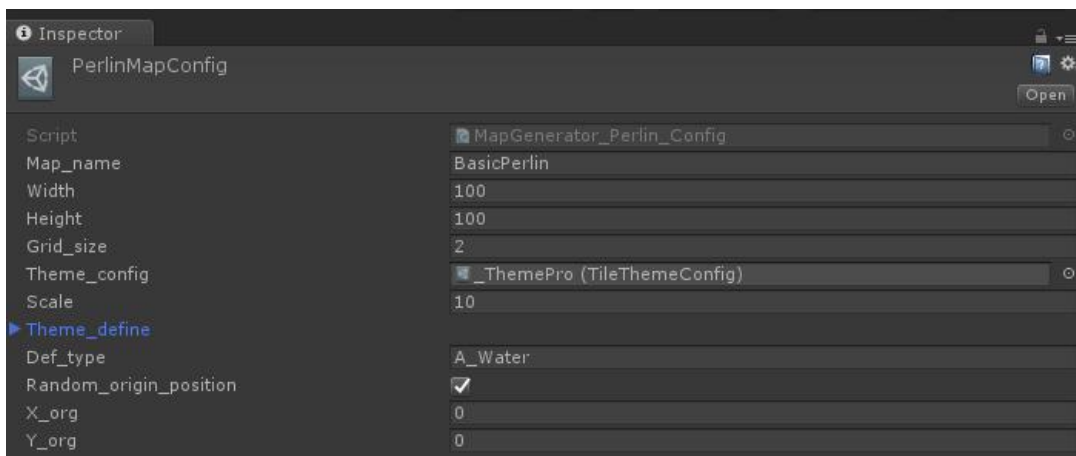


点击Clear，可以清除这个迷宫。

(3) 如何创建-随机地图

(3.1) 配置

在菜单中选择：MazeGen-Random Map Config可以创建一个基于PerlinNoise的随机地图的配置文件。他会出现在TileMazeMaker/Resources目录下面



MapName: 为这个地图添加一个可爱的名字

Width: 这次的数值，就是真真正正，Tile模型的数量了，100表示一行会有100个Tile

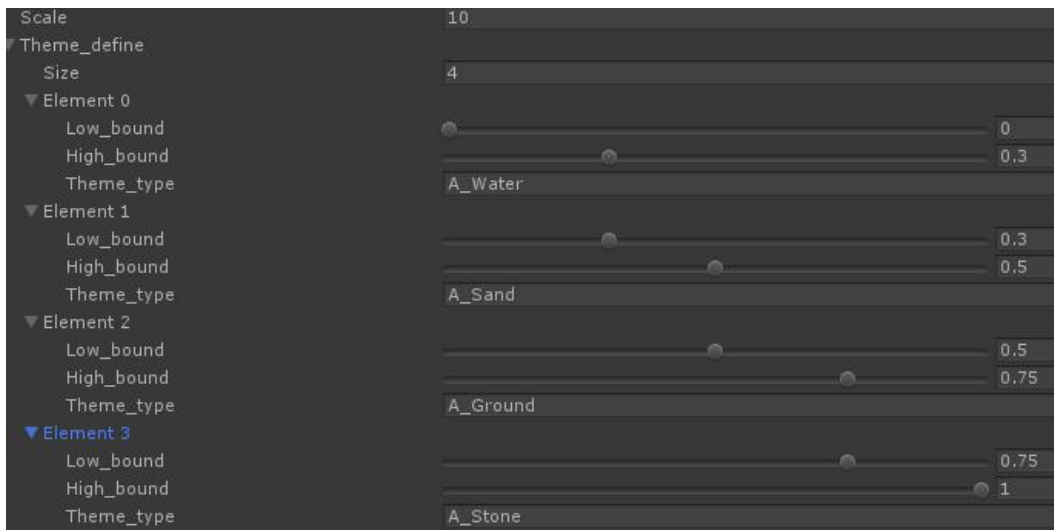
Height: 100表示一列会有100个Tile

Grid Size: 表示一个Tile的大小，如果你建立的是3*3大小的Tile，你需要修改这个值，来适配你的模型。

ThemeConfig: 将你之前创建的ThemeConfig文件拖进去。

Scale: 如果这个值很小，则地图看起来会很密集，10是一个不错的建议值，如果增大这个值，地图会变得稀疏，亲自试一下调节这个值，你就会很轻松理解他的原理。

ThemeDefine: 控制那个高度，生成何种Tile



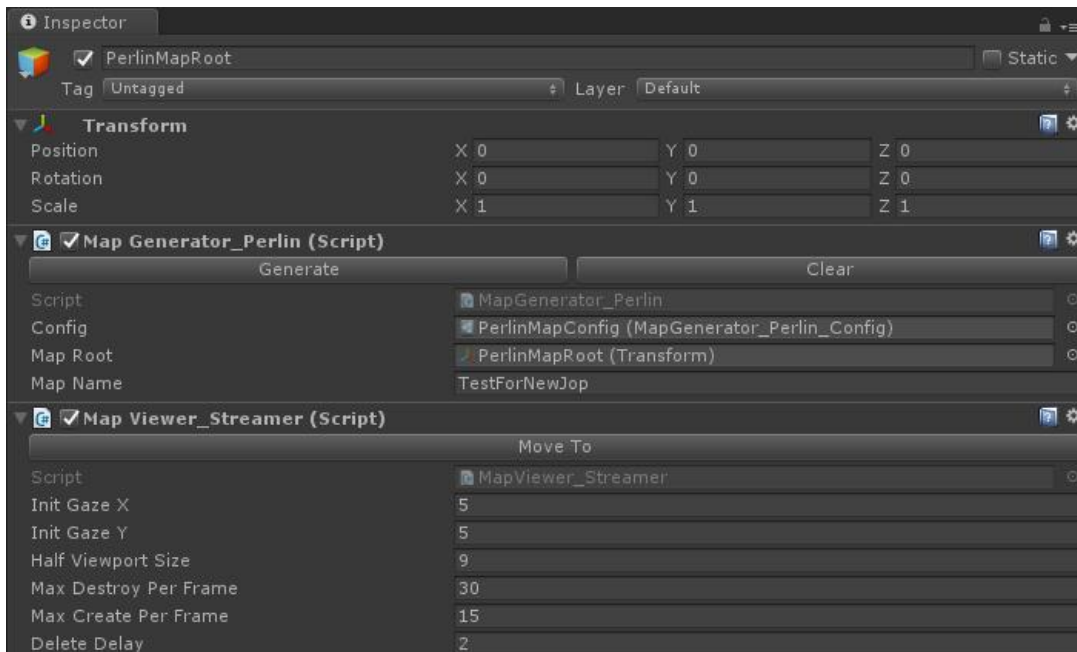
例如，如果你将随机范围设置为0-0.3，资源名字设为A_Water，PerlinNoise 随机出0-0.3的时候的时候，将会随机一个A_Water的格子，总体出现概率也约为30%。

Def_Type: 默认格子是什么，如果有漏掉的区间，则都是默认格子

Random_origin_positon: 选上这个选项，每次点击Generate你将获得不同的地图。

X_Org/_YOrg: 如果没有选择Random_origin_positon，你可以自定义一个地图样式，同样的X_Org和Y_Org值和Scale值，总能获得相同的地图。

(3.2) 生成



目前我已经对PerlinRandomMap做了扩展，让他能够支持流式读取，也就是支持地图分页了，这样你可以创建一个1000*1000，也就是由100万格子的地图，并且将他分页显示出来，而不必一次全部加载。

如果不需要这个功能，你可以使用MapView_Full这个脚本替代MapView_Streammer这个脚本

MapGenerator_Perlin

Config: 你之前配置的PerlinNoise Based Random Map的配置文件。

MapRoot: 你想要作为地图根节点的Transform。

MapName: 你想要保存的地图的文件的文件名。地图保存功能将会在下一个更新版本中提供。

MapViewStreammer

Init GazeX: 观察中心点格子的X索引

Init GazeY: 观察中心点格子的Y索引

HalfViewportSize: 分页读取时候的读取半径

MaxDestroyPerFrame: 每帧最多销毁多少个对象，如果使用了对象池，可以将这个值设置的高一些

MaxCreatePerFrame: 这个也是为了放卡顿设置的选项，如果使用了对象池，可以将这个值设置的高一些

DeleteDelay: 保留值。

点击**Generate**可以生成一个随机地图。点击**MoveTo**可以移动到当前观察点的位置，并生成相关区域的地图格子。

如图：分页显示地图的一个子集。

