# 地图

## 关卡

因为变成了rougelike模式，所以关卡不再是传统通关模式，请按照“向前行走一定距离即进入下一关”的方式来理解关卡配置。

### 行走距离

* distance，int类型。
* 增加的一个关键字段用来解释在不同距离使用不同的配置数据。
* 原则上，关卡配置会按照距离升序排序。
* 比如，假设现在有3条数据，距离分别是20,40,60。玩家行走距离是10，没有超出第一条数据，那么就应该使用第一条数据的内容；玩家行走到50米，查询后，就应该使用第三条数据。玩家行走到100米，经查询，最大距离数据是60，那么此时依然应该使用60这条数据（因为找不到比60更大的了）。

事实上，距离判定，应该是以屏幕最上方一行的距离作为判定依据，而不是玩家的行走距离，这里为了说明方便，所以是以玩家行走距离作为判定。

注意，每一次进行距离判定的时候，应该是一个模块用光的时候。比如玩家虽然走到21米，但是因为当前使用的模块还没有使用完毕，所以此时应该继续使用该模块，直到本模块使用完毕（所谓“使用”是指使用该模块来生成行），才会取下一条配置数据再使用新的模块。

### 空格资源

* emptyList，string数组，把以前那个字段（res\_emptys）废掉了。
* 空格子应该使用哪一个prefab。
* 考虑到程帆应该会画多种地图演示，所以这里提供多个美术资源进行随机挑选（暂时先这样，我也不知道程帆之后是否会有其他需求，这个主要依程帆）。

### 阻挡资源

* obstructList，string数组。
* 阻挡应该使用哪一个prefab。

### 地图模块

* map\_models，ulong数组。
* 虽然是之前就有的数据，但是这里用法要改变一下。
* 决定使用哪一个模块的时候，会在这些模块里面随机调一个，但是为了避免重复出现相同的模块，设计了以下权重算法：

假设，现在有模块2000001|2000002|2000003，它们3个初始权重都是1，所以选中它们的概率都是1/3。假设选中了2000001，此时，其他所有模块权重翻倍，而2000001不变，所以他们的权重就变成了1,2,2，此时再次选中的2000001的概率为1/5。依次类推。

注意，本算法导致权重指数增加，所以在处理的时候需要注意避免超过边界值。可以当最大权重超过1000时，所有权重除以10，但最小权重为1。

关卡数据切换时候，取新的模块数组，权重也会恢复到1。

### 敌人列表

* enemys，ulong数组。
* 本距离下，应该使用哪些敌人，随机。

### 敌人等级

* enemy\_level,int数组。
* 出现敌人的等级，这是一个范围，在该范围内随机。

### 物品列表

* items，ulong数组。
* 先放这里，以后再决定要怎么用。

## 模块

### 地图块内容

* contents，string二维数组(注意之前的content弃用)。
* 第一维表示该模块有多少行，第二维长度一定是6，表示每一行的内容。
* 一行包含6格，每一格，如果没有任何内容，表示为空格。
* o,表示这格是一个阻碍。
* r,表示这格是一个敌人，它带两个参数（使用\_分开，注意这没有直接在数组里做切分），第一个代表敌人强度，第二个表示出现概率（1为100%）。
* 目前处理这两种格子就好，之后再做扩展（设计到机关与补给的时候，补完）。

### 如何使用？

* 先确定使用哪个地图模块。
* 每个地图模块，从第0号元素开始（可以使用contents.ToArray(i)这种方式），取到每一行的内容，依此生成地图每一行。
* 直到用掉了该模块所有数据，换下一个模块。