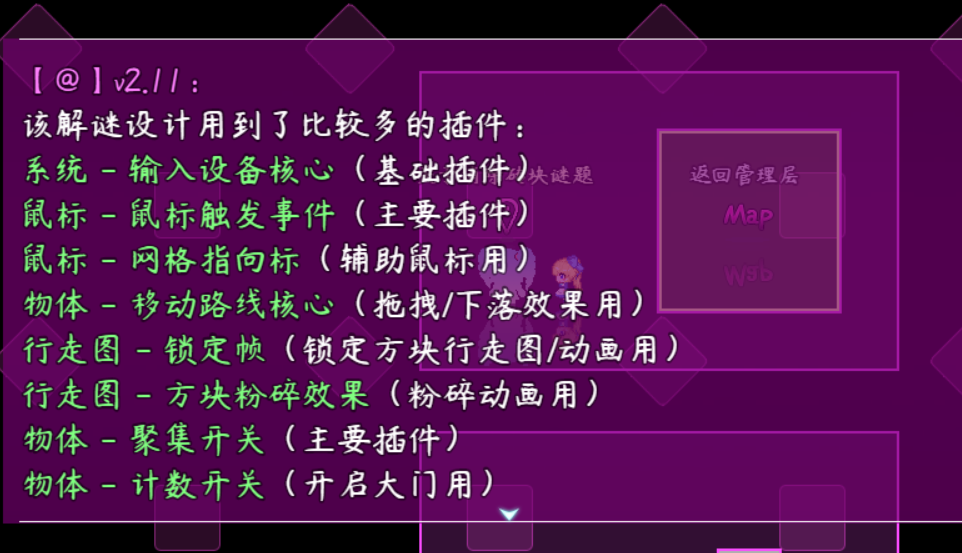
## 概述

### 相关插件

示例中有**谜题设计说明**，你可以根据那个参考制作小游戏。

下图为 设计-消除砖块 的相关插件：



## 脉冲开关关卡设计

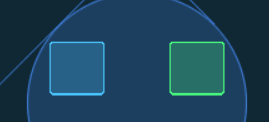
**脉冲开关看起来非常复杂，但是，无论关卡怎么变，都离不开数学与图论的规律。**

**该关卡很容易设计出无解谜题，把自己绕晕，必须基于一些理论来站稳脚根。**

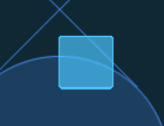
脉冲关卡中有下列定义：

（1）踩1次用开关。**这里称呼为 奇开关 。**

蓝色虽然可以踩1、3、5、7……无穷次，但是只要不是单线，就与踩1次的效果没有区别。

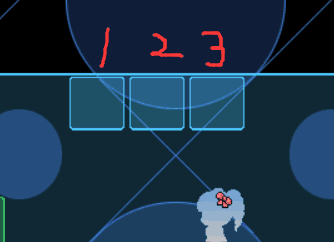


（2）踩0次或2次开关。**这里称呼为 偶开关。**



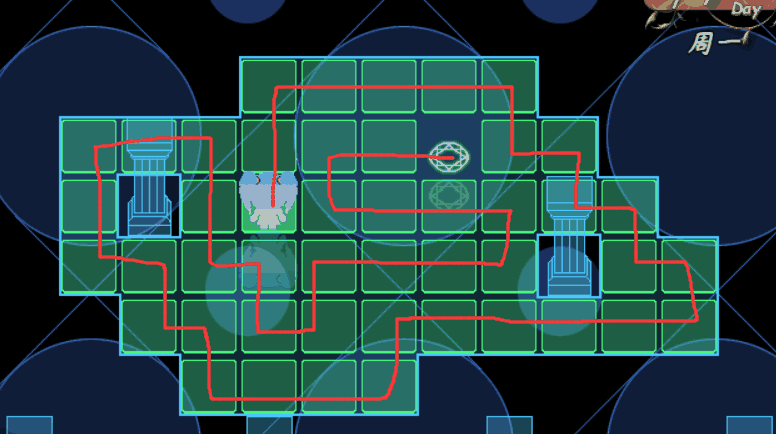
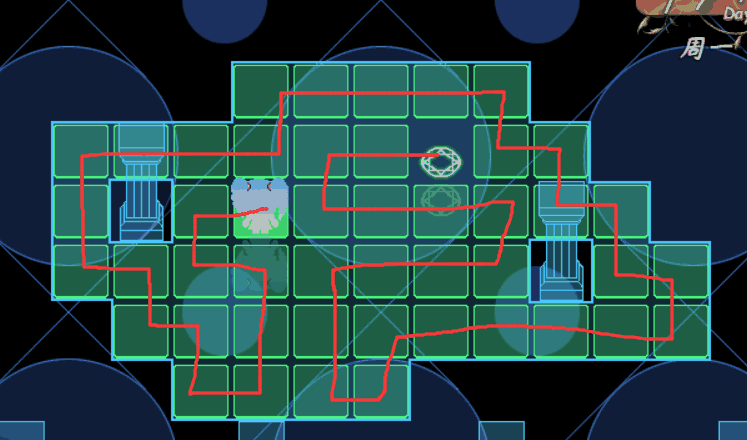
（3）下面需要从起点开始进行编号计数，这里的编号为数字分为 **奇数** 和 **偶数** 。

编号决定了开关的在复合条件下的点亮结果。



### 贴墙法

在**只有奇开关**里，只要你找到了最快捷的方法贴墙，那么无论开关如何复杂，都会迎刃而解。（即使变起点，也不能增加多少难度）

玩家经过摸索，迟早会渐渐掌握贴墙法，要想增加难度，应该在**偶开关**维度上增加。

增加难度的维度也有很多其他方法，我见过的有下面几个：

1.限制一个图块只能单向走。

2.一个图块是钥匙，踩了才可以解锁另一个图块路线。

3.有的开关踩了后，会传送到另一个开关。

(´⊙ω⊙`) 个人认为，这些维度只会把路线变得越来越复杂，并不能增加难度。

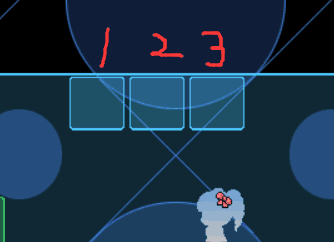
### 单路线点亮

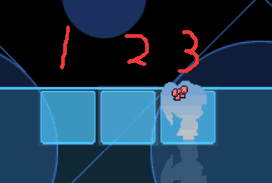
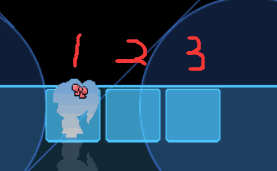
这里，开始分析**奇偶维度**上的设计，下面来看看组合的情况：

从起点1开始踩，图中为 **奇开关+奇开关+奇开关 = 奇数** 情况。

三个奇开关，因为点亮次数是奇数次，奇数+奇数+奇数 = 奇数，所以全部点亮结果是奇数位置。

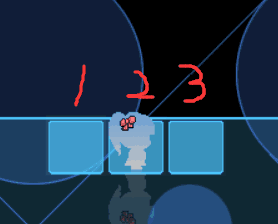
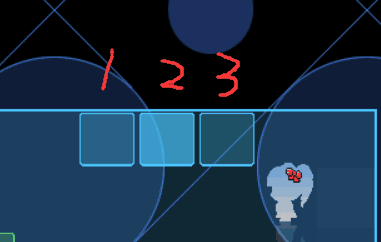
**根据计算得出，只踩在这3块开关上，想要全部点亮开关，你只能处在 奇数(1,3) 的位置。无论怎么变，在 偶数(2) 的位置，一定会有一个开关灭。**





接下来。从起点1开始踩，图中为 **奇开关+偶开关+奇开关 = 偶数** 情况。

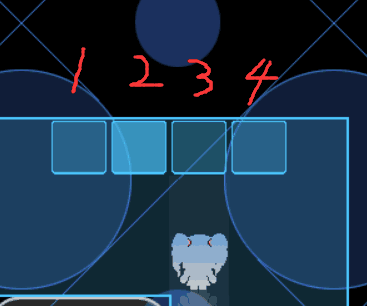
**根据计算得出，只踩在这3块开关上，想要全部点亮开关，你只能处在 偶数(2) 的位置。无论怎么变，在 奇数(1,3) 位置，一定会有一个开关灭。**

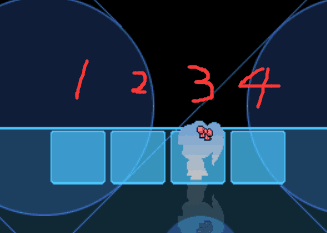


根据上面的知识，可以一直延伸到更多开关：

从起点1开始踩，图中为 **奇开关+偶开关+奇开关+奇开关 = 奇数** 情况，

根据计算得出，只踩在4块开关上，点亮全部开关，必须站在 奇数（1,3）上，偶数（2,4）点亮永远无解。

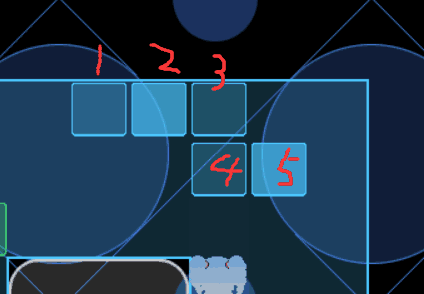


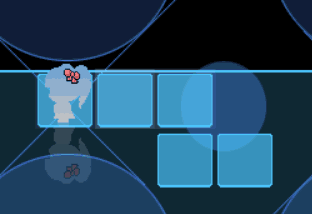
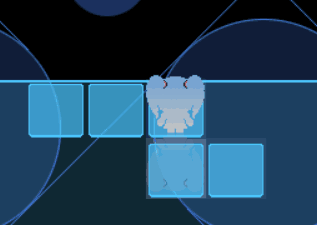


下面改变一下形状，

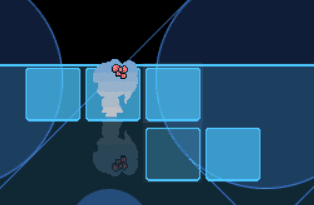
从起点1开始踩，图中为 **奇开关+偶开关+奇开关+奇开关+偶开关 = 奇数** 情况。

根据计算得出，只踩在5块开关上，点亮全部开关，必须站在 奇数（1,3,5）上，偶数（2,4）点亮永远无解。





只能在 奇数(1,3,5) 上全亮，偶开关一定会漏一个。



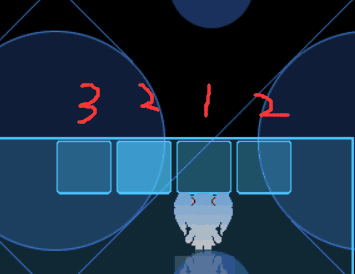
### 多路线点亮

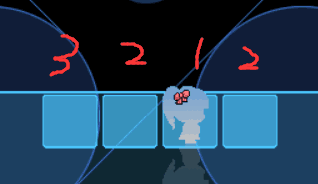
下面开始说明，起点不一样的情况：

**下图，起点从中间开始，这就意味着，左右分隔成了两条线。你需要分段开始判断数字。**

判断原理依然一样， **奇开关+偶开关+奇开关+奇开关 = 奇数** 情况，

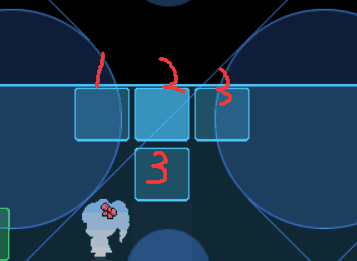
只能在 奇数(1,3) 处全亮，偶数(2) 一定会漏一个。





下图路线出现分支，也是多路线的情况，按照上述方法判断：

**奇开关+偶开关+奇开关+奇开关 = 奇数**， 奇数(1,3) 处可全亮。



**下面变复杂一点，**

从起点开始，出现分支，然后分支又合并：

奇开关+偶开关+奇开关 +奇开关+偶开关 +奇开关+奇开关+奇开关 = 偶数

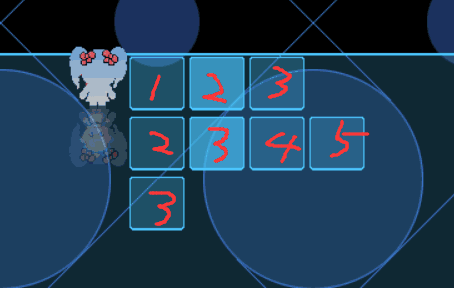
2,4,处全亮，其它的总会缺一个。



更多的路线可能性集合在一起，

奇开关+偶开关+奇开关 +奇开关+偶开关 +奇开关+奇开关+奇开关 = 偶数

2,4,处全亮，其它的总会缺一个。



**根据上述的情况，如果你的设计中布满了开关，要全部点亮，一定会有奇开关偶开关错开的情况，错开的就是永远无解情况，你需要注意你的终点位置。**

### 第六关的误导原理

**脉冲开关的误导比较难做，你必须基于理论，不然你自己会被绕进去。**╭(°A°`)╮

很多时候画出了无解的方法，以为是误导，结果……把自己绕晕了。

脉冲开关由于自己的数学特性，造就了两个极端，一个是永远无解，一个永远有解。

**误导，就是想办法把玩家绕进无解的深渊**。

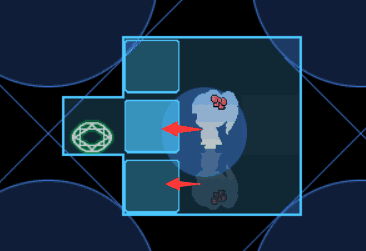
根据之前的奇开关偶开关理论，

从中间进，是2,1,2，奇开关+偶开关+奇开关=偶数。

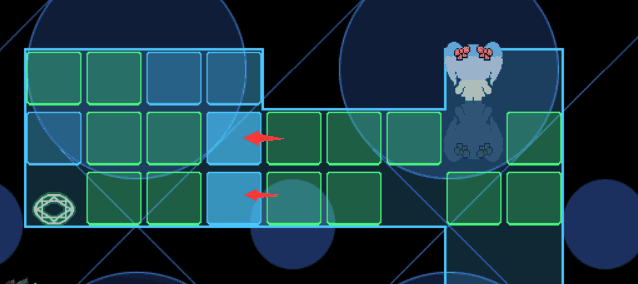
偶数永远有解，奇数永远无解，但是终点恰好是在 奇数(1) 位置。

从下面进，是1,2,3，奇开关+偶开关+奇开关=偶数。

偶数永远有解，奇数永远无解，这次的终点恰好是在 偶数(2) 位置。



第六关的误导图示如下图。（具体解谜方法就不赘述了，自己试试看吧！）



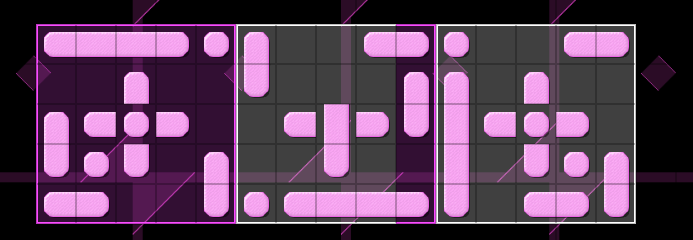
## 逻辑图形关卡设计

逻辑图形的关系实际类型并不多，有以下类型： 平移、旋转、对称、循环、缩放、并集、交集、差集、A+B=C等式推理、形状表示某数字。

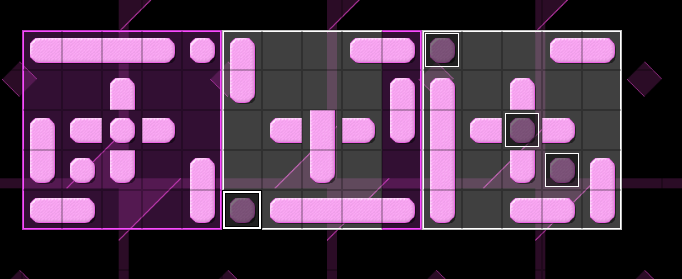
### 图形绘制方法

谜题如何设计这里不再赘述， 这里说明一下事件设置的方法：

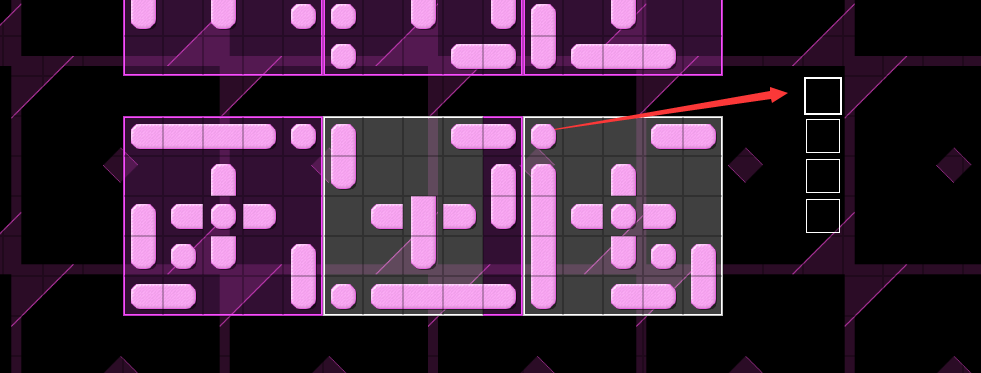
1.先把答案画好：



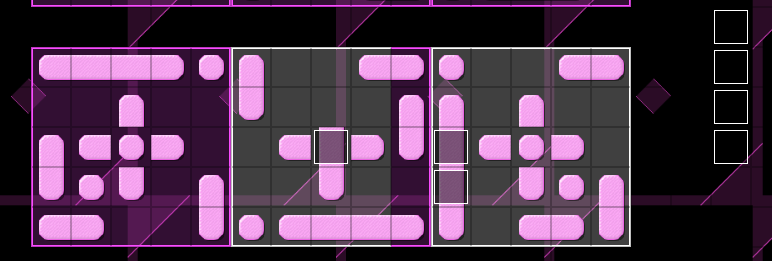
2.把形状1标出来：



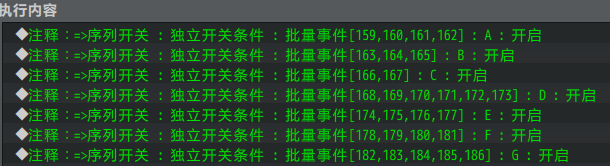
3.把形状拿出去，看看事件的id，分别为：159,160,161,162

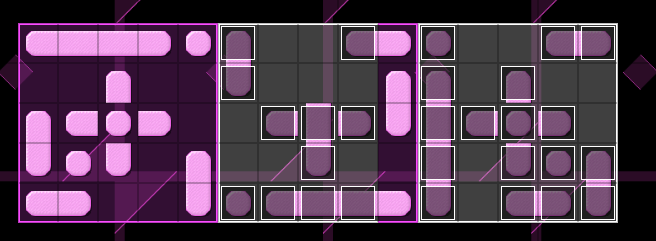


4.再放形状2的，以此类推。

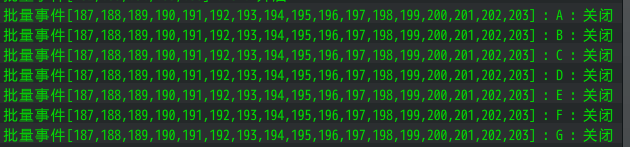


5.记录答案的所有形状：





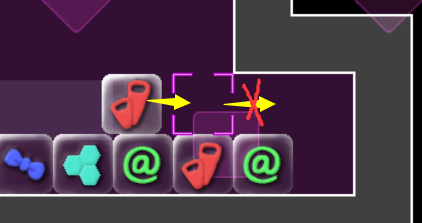
6.最后补足空的部分，并且设置 空 的独立开关全关闭状态。



## 消除砖块关卡设计

### 特殊规则

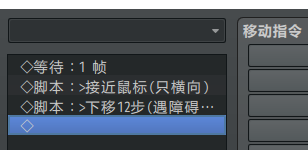
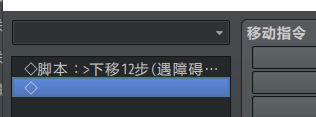
**玩家鼠标拖移时，如果接触到了方块，那么就消除**。



**如果是物体自身”下落”的情况，两个相邻方块不会被消除**。



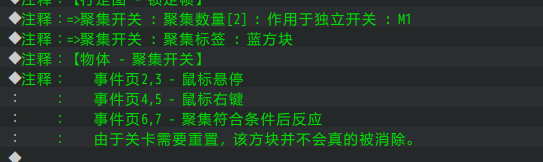
上述的特殊规则，是基于 聚集开关 插件的“**完全静止2帧之后，才会触发**”的功能而实现的。正常下落时，没有任何等待，而鼠标拖动时，会等待1帧，这样就可以实现下落不消除，而拖移会消除的效果了。



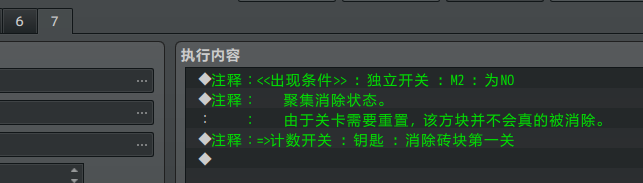
（如果不等待这一帧，鼠标一直拖移，就说明物体一直没有静止下来，那么聚集开关就不会被触发。）

### 消除设计

1．你可以直接复制示例中的方块到你的游戏中，修改下面的聚集数量和标签，控制不同的效果。



2．需要注意的是，由于关卡是要求所有砖块被消除，所以这里计数开关的 钥匙 ，是放在消除状态的事件页中的。



3．示例中的”重力”就是下移n步的结果，你当然可以修改成上移、左移、右移，使得方块的重力随时可变。

