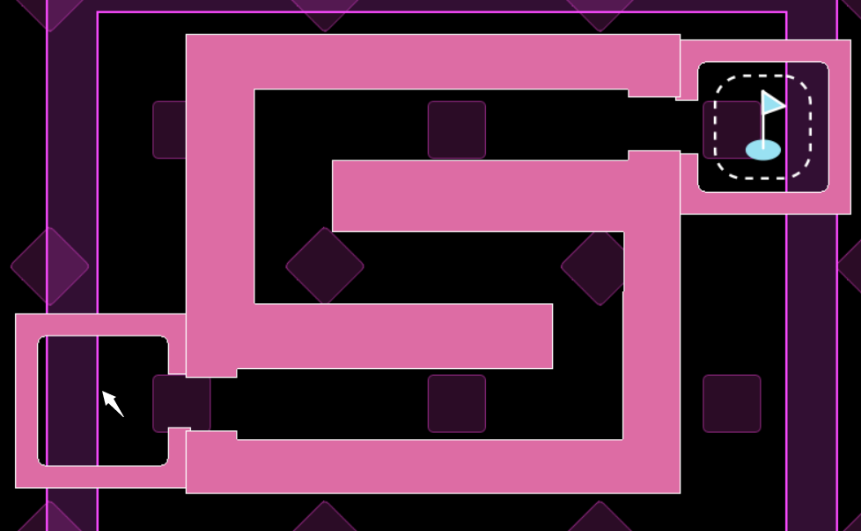
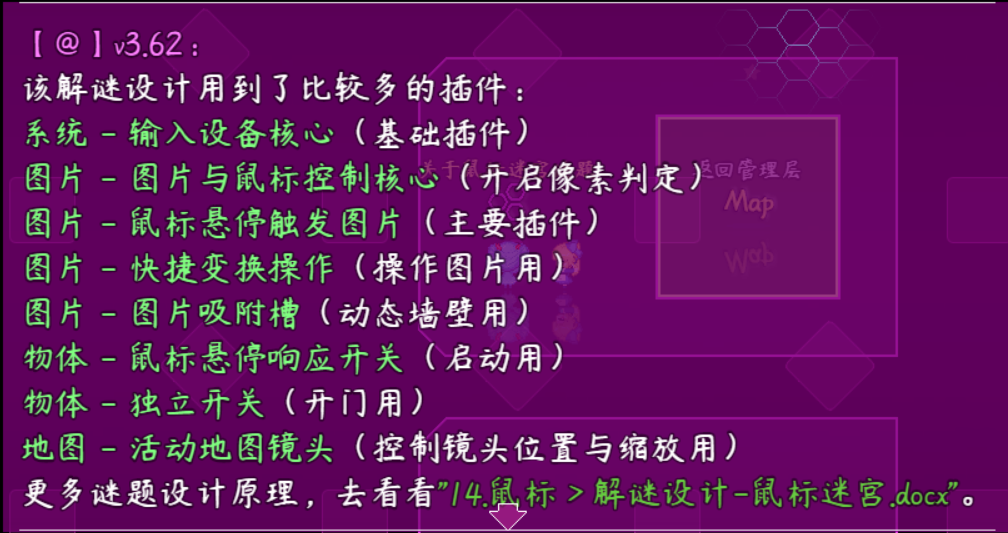
## 概述

### 基础插件

鼠标迷宫 游戏关卡如下图。



如果要从零开始设计，需要下图的插件：



**注意，该文档只详细说明 整体设计思路 和 解谜方法 。**

**对于具体 事件如何写 ，需要去示例中 图片管理层 了解一下 图片与鼠标 的使用。**

**想了解图片触发如何写，去看看 "16.图片 > 关于图片与鼠标控制核心.docx" 。**

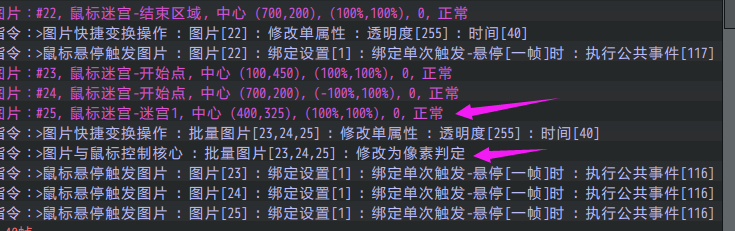
### 基础配置

具体注意示例下面的地方：

#### 1）图片

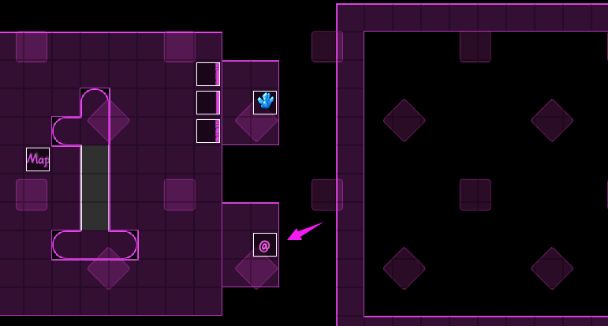
鼠标迷宫需要图片作为开始点、终点、迷宫墙壁、失败时的叉，

留意图片创建指令。



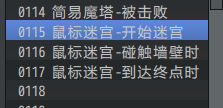
#### 2）地图事件

关卡通过一个事件进入到关卡。剩下的都通过公共事件来专门处理。

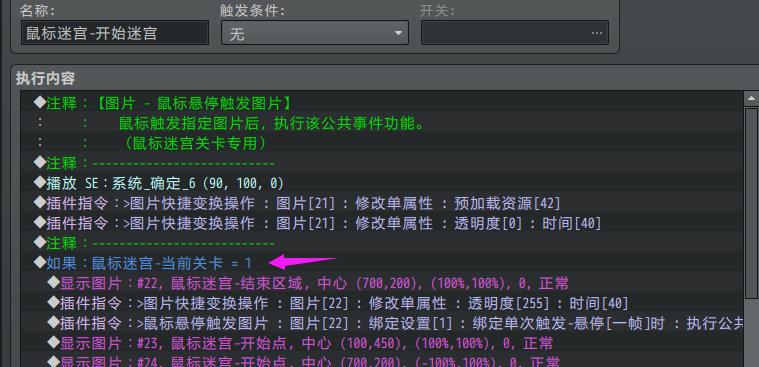


#### 3）公共事件

留意公共事件的控制设置：



注意这三个公共事件都区分6个关卡，虽然大部分指令相似，但注意小部分差异。



## 鼠标迷宫-谜题设计

### 设计简介

#### 1）游戏规则

纯鼠标控制的游戏，鼠标点击按钮后开始迷宫闯关。

如果鼠标碰到了迷宫边缘，则关卡失败。

鼠标需要到达终点才可通关。

（其实鼠标迷宫更像一个敏捷类的小游戏，而非解谜。）

|  |
| --- |
| 必须鼠标才可以玩，因为触屏可以瞬移，所以不适合。 |

#### 2）难度分配

迷宫路径的宽窄，能影响关卡难度。

迷宫墙壁是否为动态，能影响关卡难度。

迷宫路线的长度，能影响关卡难度。

#### 3）禁用能力

禁跳跃。

#### 4）卡关注意事项

该解谜可随时用鼠标重置，不会卡关。

### 设计建议

#### 1）设计者视角

从设计者的角度来说，这个小游戏单独看并不好玩。

该小游戏需要一个剧情契机，

比如：“解密码锁时”、“打开保险柜时”、“拆开某炸弹时”。

在这些剧情中穿插该小游戏，可以给玩家提供一部分代入感。

#### 2）关卡安排建议

如果在主线剧情中加此小游戏，注意不要设计的太难。

在支线剧情中，可以作为 宝箱开锁时 的开锁用小游戏。

#### 3）灵感来源

2024年5月2日，作者我整合了图片的所有插件。

也完成了 鼠标悬停触发图片 插件。

但是好几个群友用了最新插件后，总是跑来问我：

“为什么不能根据图片的透明部分来判断悬停？为什么没有“像素判定”？“

因为以前有个麻烦的家伙开发了能根据 图片的透明部分来判断悬停 的功能。

（插件PictureCallCommon.js）

本来我一点都不想加的。ヽ(#`Д´)ノ 最后还是勉强加一下吧。

加了之后，灵感涌现，我发现基于这个 像素判定 的特性，

可以制作那个 “胖子打穿电脑” 中的小游戏。



于是就开坑了。

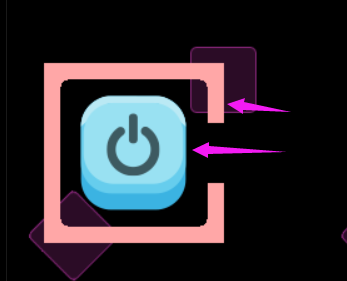
### 按钮设计细节

#### 1）按钮大小与迷宫范围

按钮的图片大小，需要在开始点墙壁内。

避免出现点击按钮就碰到墙壁的bug。



#### 2）点击按钮并开始游戏

开始迷宫前，玩家必须先把鼠标放在一个指定的范围内。

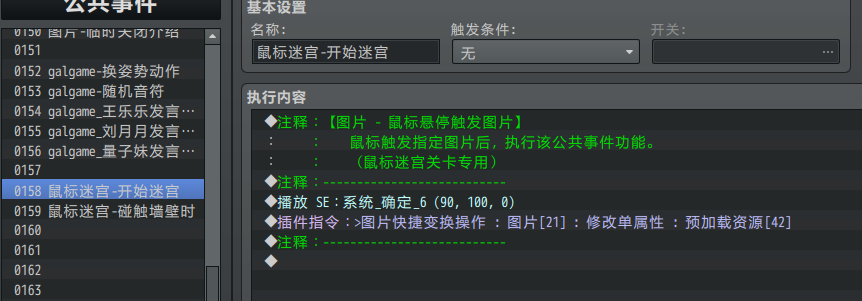
所以这里需要准备一个按钮。



玩家点击开始游戏后，显示按钮。

然后玩家点击按钮，按钮的范围 就是玩家开始游戏前 鼠标位置的范围。





由于这里作者我想节约公共事件数量，

所以这里只绑定了 左键按下[一帧] 的单次触发。

按钮点击后，直接消失，进入迷宫闯关流程。



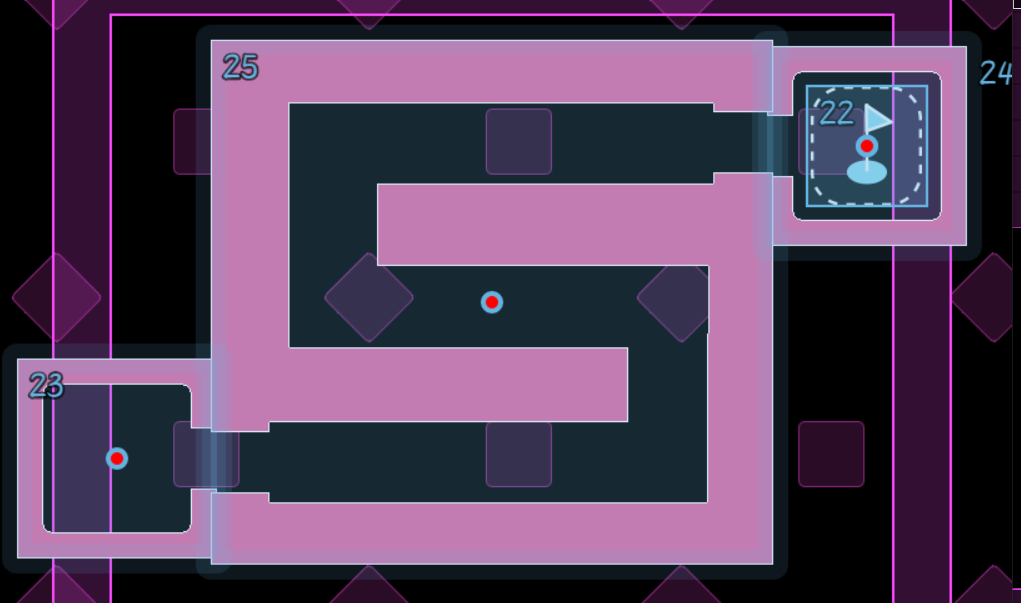
|  |
| --- |
| 一个按钮图片如果要实现完整的按钮点击功能，则需要设计：  悬停[一帧]时、离开悬停[一帧]时、  悬停左键按下[一帧]时、悬停左键释放[一帧]时  四个单次触发，并且每个触发都对应一个公共事件。  并且悬停时还需要制作按钮高亮的反馈效果。  详细介绍可以去看看：“16.图片 > 关于图片与鼠标控制核心.docx”。 |

### 迷宫设计细节

#### 1）像素判定

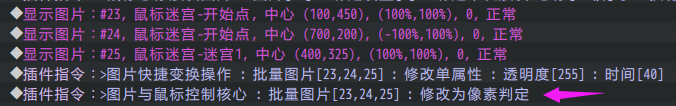
如果在游戏中打开debug查看碰撞体，你可以看到下面的情况：





终点范围就是终点图片的矩形范围。

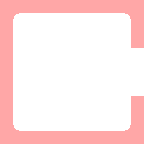
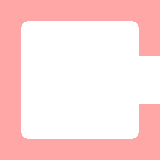
而迷宫的墙壁图片，都需要开启像素判定，确保鼠标根据墙壁来判断是否碰壁。



#### 2）墙壁厚度

如果玩家鼠标移动速度非常快，有几率实现鼠标穿墙效果。

所以这里需要将墙壁加厚一点。

|  |
| --- |
| 由于加厚要反复调整图片资源大小，比较麻烦，所以只要加厚一点点就差不多了。  也没必要完全禁止瞬移穿墙，毕竟有不少玩家愿意挑战这种玩法。 |

#### 3）起点与终点位置

在设计的时候，你可能会产生疑问：

“为什么不直接把整张图片画成迷宫？而要将起点终点拆成两个图片？”

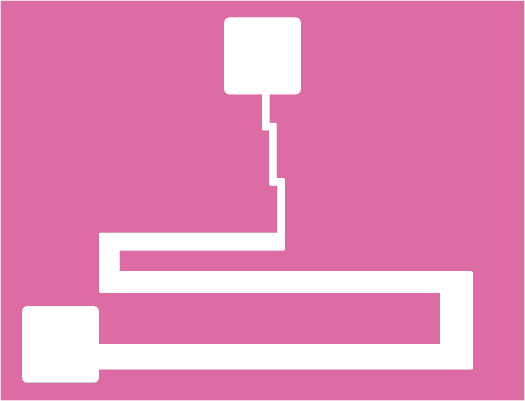
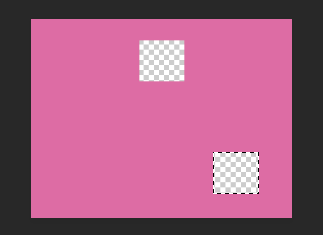
其实按照整张图片来设计也可以，只是设计思路上有些差异。

作者我最初的思路是：

确定按钮位置 》 确定终点位置 》 给起点终点画墙壁 》 画能连接起点终点的迷宫

后来第六关就是直接整张图片画成迷宫，思路为：

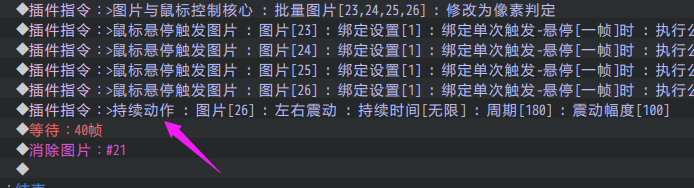
图片铺满画面 》 画迷宫路线 》 给起点终点抠两个洞 》 根据洞确定按钮位置



|  |
| --- |
| 不同的设计思路能影响设计关卡的指令。  所以如果某个设计方法行不通，可以尝试换其他的设计思路哦。 |

#### 4）动态墙壁设计

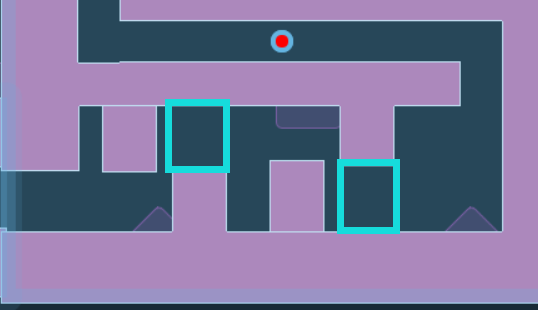
之前作者我没想那么多，直接用了持续动作，让图片左右移动。



在游戏测试的时候才想起来。

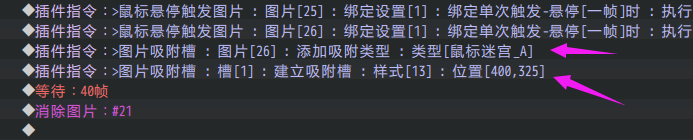
**持续动作只影响贴图，不影响碰撞体的真实位置。**

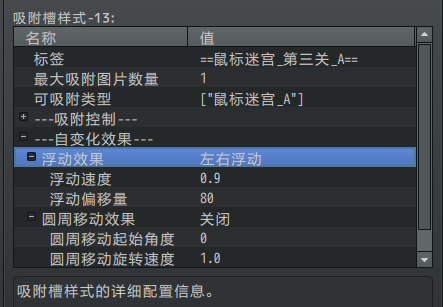
打开debug发现贴图在原位。



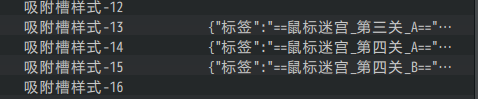
所以这里改用了吸附槽，

图片被吸附后，会被强制跟随移动，这样也就影响到了碰撞体的位置了。

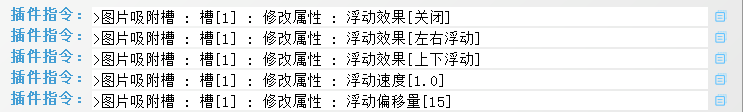




因此，每个动态墙壁都对应了一种吸附槽的样式：



你也可以节约样式，全部改成插件指令控制：



## 鼠标迷宫-谜题解法

### 放大窗口法

把游戏窗口放大，这样鼠标能活动的范围就变宽了，同时也降低了难度。

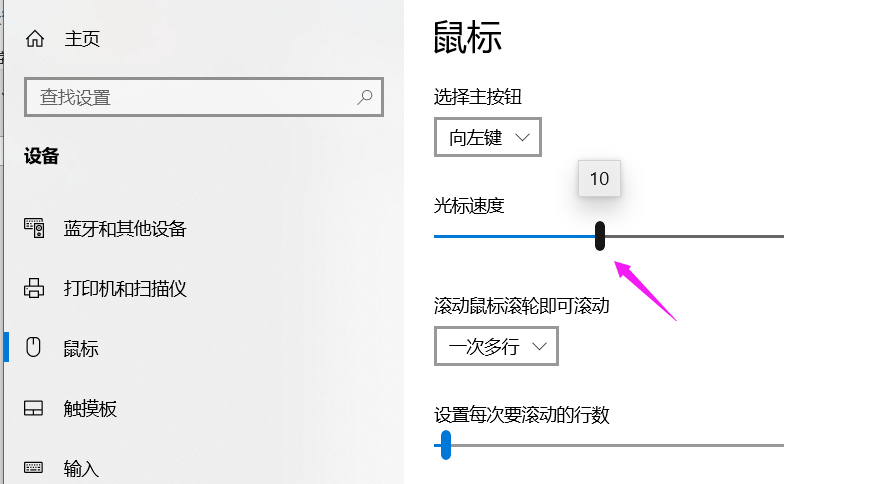


### 调整鼠标灵敏度法

这个游戏最大的问题就是容易手抖，然后就炸了。

面对这个问题，可以直接修改系统里的鼠标灵敏度，降低鼠标移动的难度。





|  |
| --- |
| 图中是win10的鼠标灵敏度设置。  如果是其他操作系统，可以去百度搜索修改鼠标灵敏度。 |