## 概述

### 相关插件

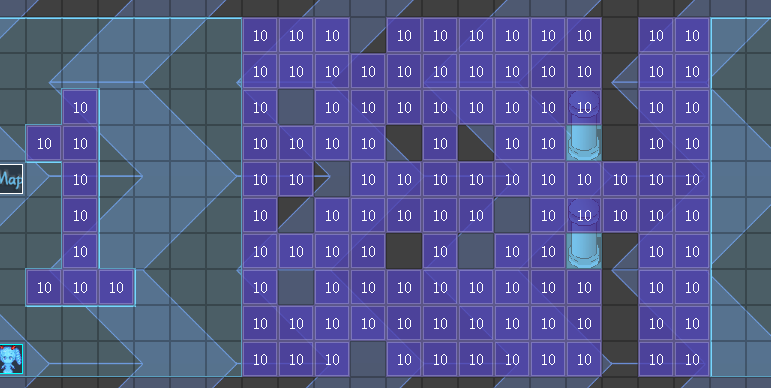
插件如下：

◆Drill\_LayerSlipperyTile 图块 - 物体滑行

地图中可以放置光滑的地面，使得玩家或事件可以在上面滑行。

光滑地面占用了R区域10，

R图块占用可以去看看“26.图块 > 关于插件与图块R占用说明.xlsx”。



### 名词索引

以下你可以按住ctrl键点击下面的词，可以直接定位到想了解的名词：

|  |  |
| --- | --- |
| 物体滑行 | [光滑地面](#光滑地面) [滑行状态](#滑行状态)  [单向滑行](#单向滑行) [斜向滑行](#斜向滑行) |
| 图块细节 | [区域判定](#_区域判定) [滑行行走图设计](#_滑行行走图设计)  [转向毯设计](#_转向毯设计) [冰块事件设计](#_冰块事件设计) |

## 物体滑行

### 定义

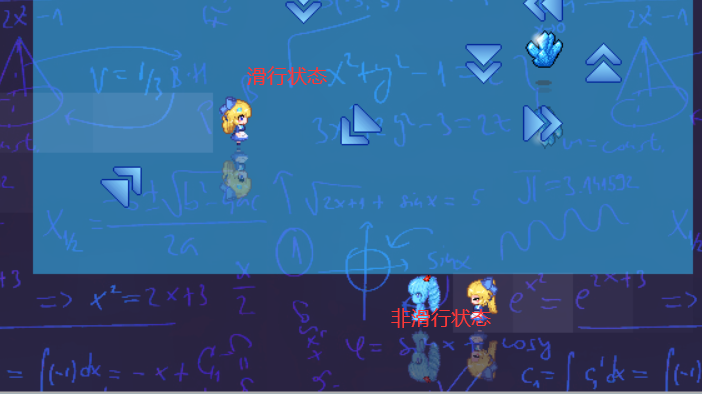
**光滑地面：**光滑地面是指能够滑行的地面，非光滑的地面上不能滑行。

在地图中涂R图块就能将指定图块变成光滑地面。



**滑行状态：**滑行状态是指物体在光滑地面上所处于的状态。

玩家/事件处于滑行状态时，无法控制自身的转向、移动，只能沿着固定方向滑行。



### 游戏操作与玩法

#### 1）单向滑行

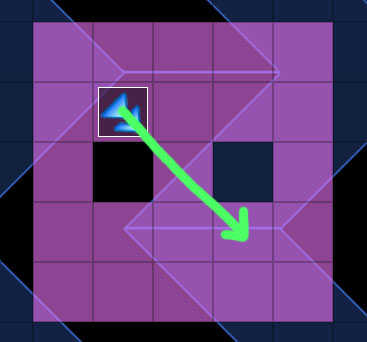
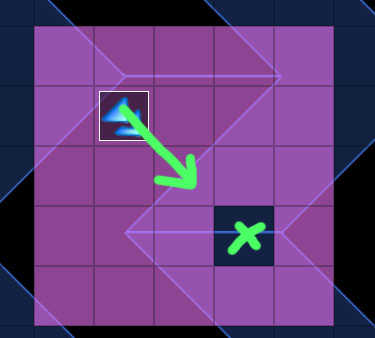
单向滑行指物体以 上、下、左、右 的方向进行滑行。



#### 2）斜向滑行

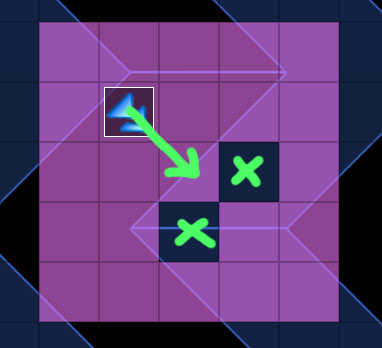
斜向滑行指物体以 左上、左下、右上、右下 的方向进行滑行。

斜向每次会识别**下一个图块的通行性**，下图如果只下方或者右方有阻碍时，则不会阻挡滑行。

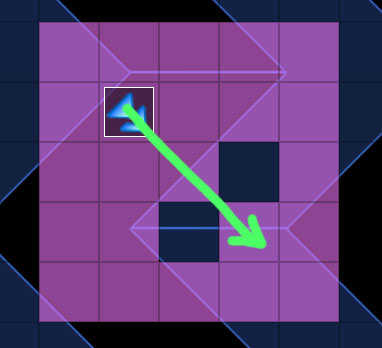
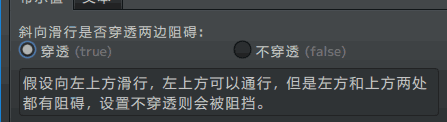


默认情况下，如果指定方向的两侧都有阻碍，则会造成阻挡。

八方向行走图也基于这个原理。可变激光区域的穿透也基于该原理。



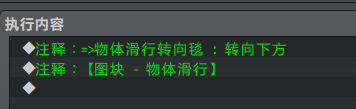
但插件中，为了确保游戏性，你设置关闭这种机制，直接穿透。



#### 3）转向毯/手动转向

转向毯是一个特殊的事件，玩家/事件踩到转向毯后，会转为固定朝向。

（转向毯建议做成单独的、在事件下方的固定事件，不要出现 转向毯还能滑行 的情况。）



你还可以通过插件指令手动强制转向。

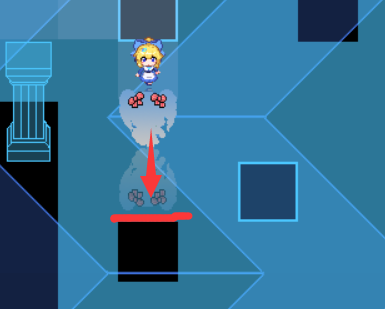


### 游戏硬性机制

#### 1）受到阻碍后，停止滑行

受到的阻碍通常为墙壁、其他事件。

如果没有设置禁止，甚至还可以放置炸弹，与其他滑行的事件碰撞。



#### 2）离开光滑地面后，停止滑行

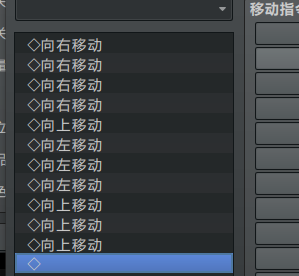
玩家走到非光滑地面时，会停止滑行。



#### 3）滑行时，移动路线会暂停执行

玩家/事件进入滑行后，移动路线会处于暂停状态，

只有结束滑行后，才可以继续执行移动路线。

#### 4）滑行时播放不同行走图

滑行时，玩家/事件会播放特定的行走图。



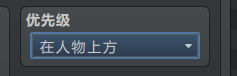
默认行走图会播放 移动动画的第3帧（踏出一只脚）。

但如果你使用了动画序列，则可以播放自定义"滑行"和"滑行静止"的动画。

详细可以看后面章节：[滑行行走图设计](#_滑行行走图设计) 。

#### 5）飞行物体不会滑行

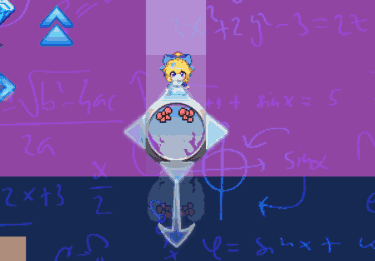
设置为“在人物上方”时，该事件不会滑行。



### 与其他能力组合

#### 1）光滑地面上无法原地转向

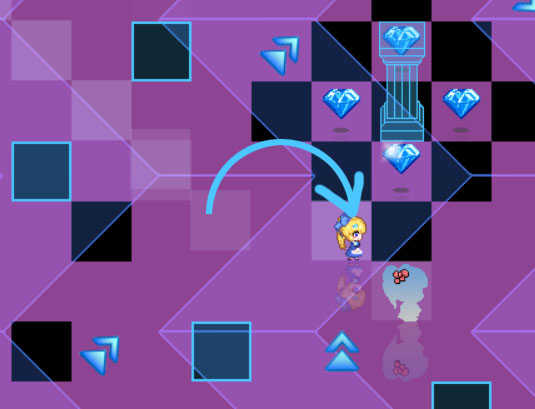
如果角色站在光滑图块上，鼠标点击转向按键或者键盘转向，都没有效果，并且发出错误提示音。



#### 2）滑行时可以跳跃

在光滑图块滑行时，玩家可以跳跃跳过转向毯，或者斜向滑行时跳跃。





## 图块细节

### 区域判定

|  |
| --- |
| 网格区域移动的机制是固定的。  举个例子，小爱丽丝从图块A移动到图块B。    **只要小爱丽丝开始移动，那么小爱丽丝就已经处于B图块位置。**  **跳跃同理，只要起跳，那么小爱丽丝就已经处于B图块位置。**  移动/跳跃过程，仅仅是处在B图块的位置，播放行走过程的动画效果而已。  这时候A图块是可通行的，B图块是阻塞的。 |

**区域判定修正：**修正后，身体的一半以上到达B区域，才判定为处于B区域。

此修正可根据你的实际游戏情况考虑开关。



物体滑行的功能在判断 物体是否在光滑地面上 用到了区域判定。

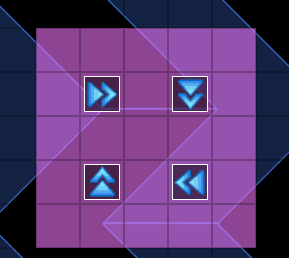
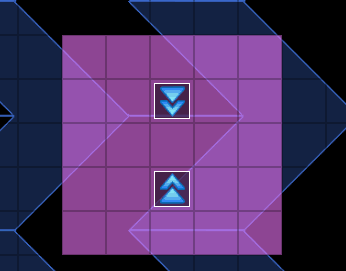
修正与不修正差别不大。

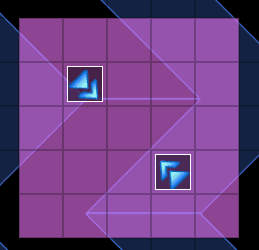
### 转向毯设计

转向毯的定义可以见前面章节：[3）转向毯/手动转向](#_3）转向毯/手动转向) 。

设计光滑地面 放置转向毯 时，

要避免下面的情况，这样会造成死循环，卡关。



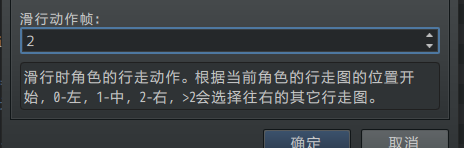


## 滑行设计

### 滑行行走图设计

默认行走图会播放 移动动画的第3帧（踏出一只脚）。

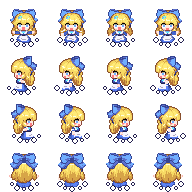




但如果你使用了GIF动画序列，

你可以自定义"<行走图-滑行>"和"<行走图-滑行静止>"的状态元。

这两个动画可以是动态的，比如小爱丽丝站不稳/打滑的动作。

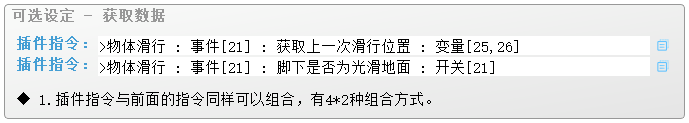


详细内容去看看："7.行走图 > 关于行走图GIF动画序列.docx" 的 动画序列标签表 。

文档中也有介绍 自定义动画序列行走图 的方法。

### 滑行状态数据获取

你可以通过插件指令，获取到部分滑行状态数据。



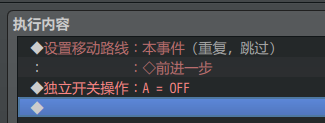
### 冰块事件设计

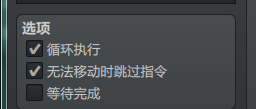
除了光滑的冰面，当然还有本身就是大冰块的事件。

#### 1）错误的制作方法

你可以考虑用纯事件来制作冰块，被推后，一直向前移动，如下图。







被玩家推过后，持续向前移动，直到无法移动即可。

这里有个问题，表面上推动后，的确没有继续移动了。

但是，当你推开阻塞的冰块，你会发现这个冰块会继续前进。

这是因为冰块被推动后，一直都在执行“前进一步”，并没有执行到独立开关A关掉。

只要你勾上了“循环执行”那么冰块的事件指令就永久循环在移动路线里面了，出不来。

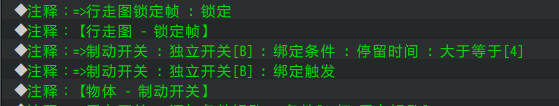
#### 2）制作方法

你必须通过别的情况来确保 冰块 被阻塞时，真的能停下来。

这里就需要主要需要用到 行走图锁定帧和制动开关 插件。

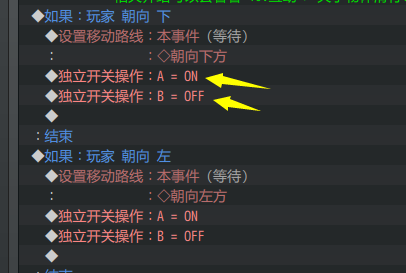
◆Drill\_EventFrameLock 行走图 - 锁定帧

◆Drill\_EventStoppingSwitch 物体 - 制动开关



当物体被推动时，将B关了，开始监听。

如果事件停下，则独立开关B会被立即开启。



前进状态的事件页中，只要并行判定独立开关B是否开启，

开启则切回事件页（关掉独立开关A）停止即可。

