**镜头教程：offset-border篇**

注：本教程建议您观看冰淇淋的镜头教程https://www.bilibili.com/read/cv21284247作为前提。

什么是offset-border镜头系统？

其实这是我个人总结的一些镜头制作经验，由camera offset和camera offset border组成主要部分，配合flag与one way camera 的使用来减少晃动。

这一套镜头系统可以解决大部分情况下的镜头需求，但不太适用于下述使用情景：

（1）多路线设计，如设置草莓岔路，在岔路口处的镜头制作可能存在问题，使用上述系统可能会牺牲一部分的偏移量；

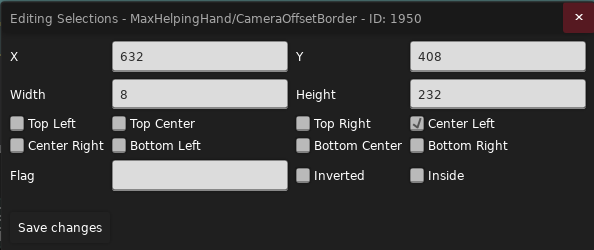
（2）运动路线过于复杂的情形，这种情况下需要使用camera offset (flag-toggled)等其他镜头trigger作为辅助，相对而言制作较为复杂；

（3）高速图。这类图需要的镜头偏移与border无关，我们需要使用到camera catch up speed。

相比冰淇淋教程中的offset-target系统，使用offset-border相对而言设置更为便捷，不会存在需要频繁改变偏移系数来实现效果的情况，因此在路线相对不复杂的情况下笔者建议使用offset-border系统来进行镜头制作。

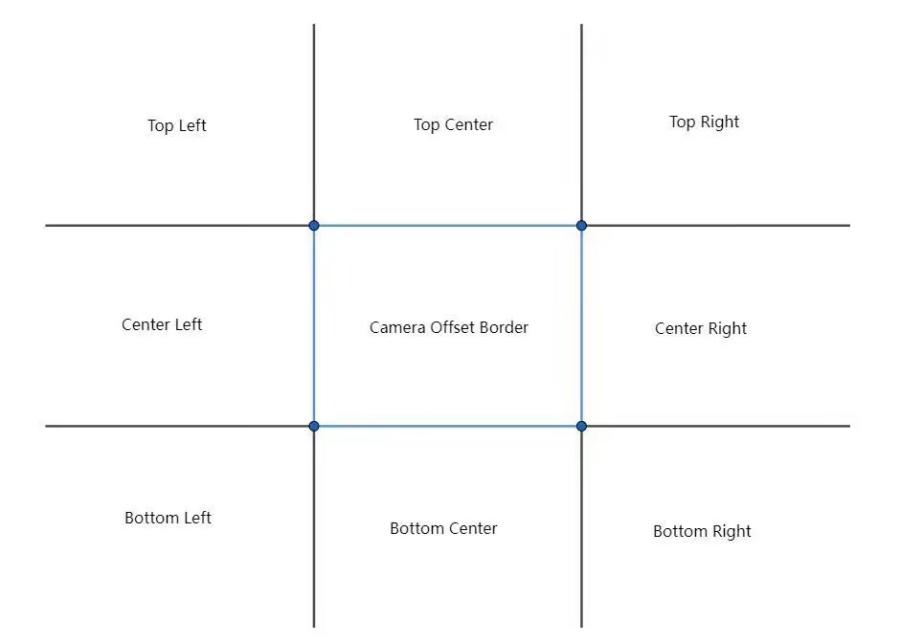
下面我将先介绍一下offset-border。

1.camera offset border的介绍



这是一个camera offset border（后称border）的页面。

我们观察top left这一行及其下方一行，这八个勾选框代表了这个trigger在玛德琳在什么区域是生效。下面是一个概览：



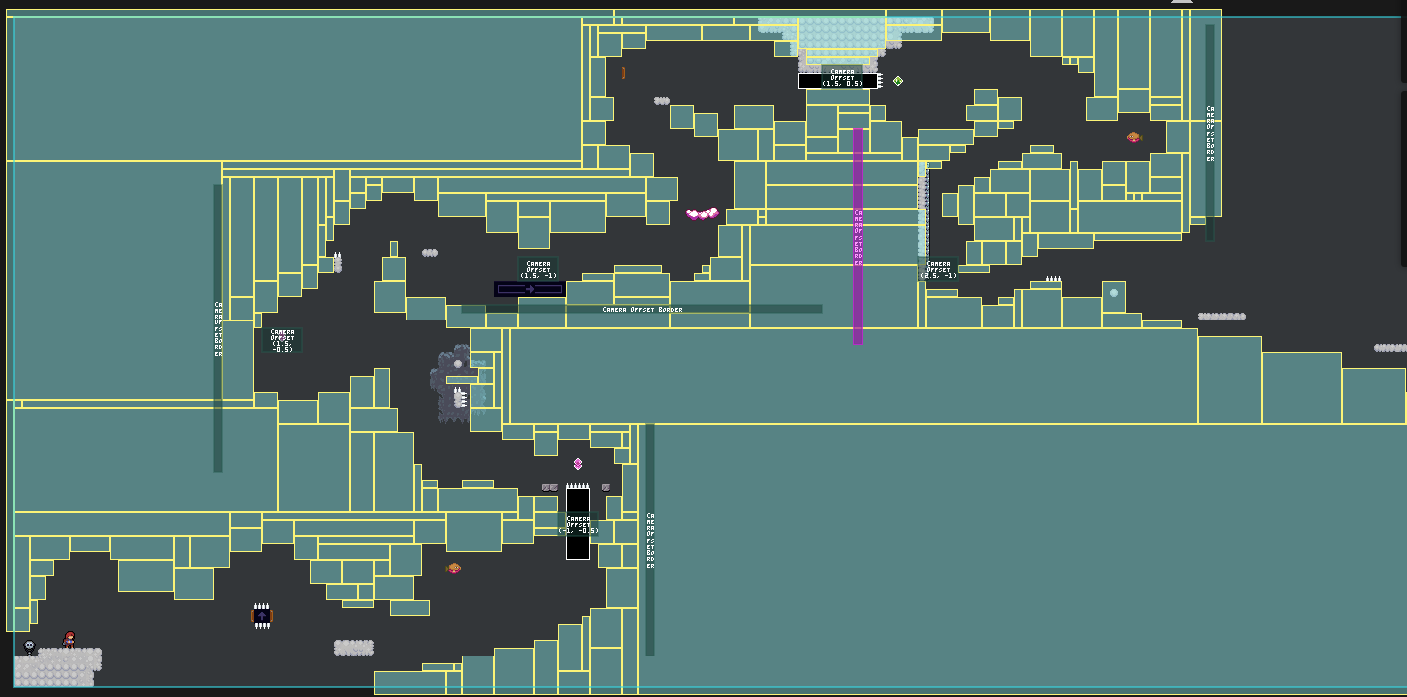
中间是camera offset border的位置，周围的区域被分为8个部分，每个部分对应一个勾选框，当该框被勾选时，玛德琳进入此位置将会受到border的作用。

flag为控制该border是否生效，例如flag内填入camera\_a01，该border就会被这个flag控制。右侧inverted，勾选时该border会在flag值为true时失效，否则在触发时生效。

border具体有什么作用呢？当玛德琳运动时，border会限制镜头的移动，让镜头的移动不会越过这个trigger所在的位置。

2.offset-border系统的使用思路

我们来看一个“简单”的例子。

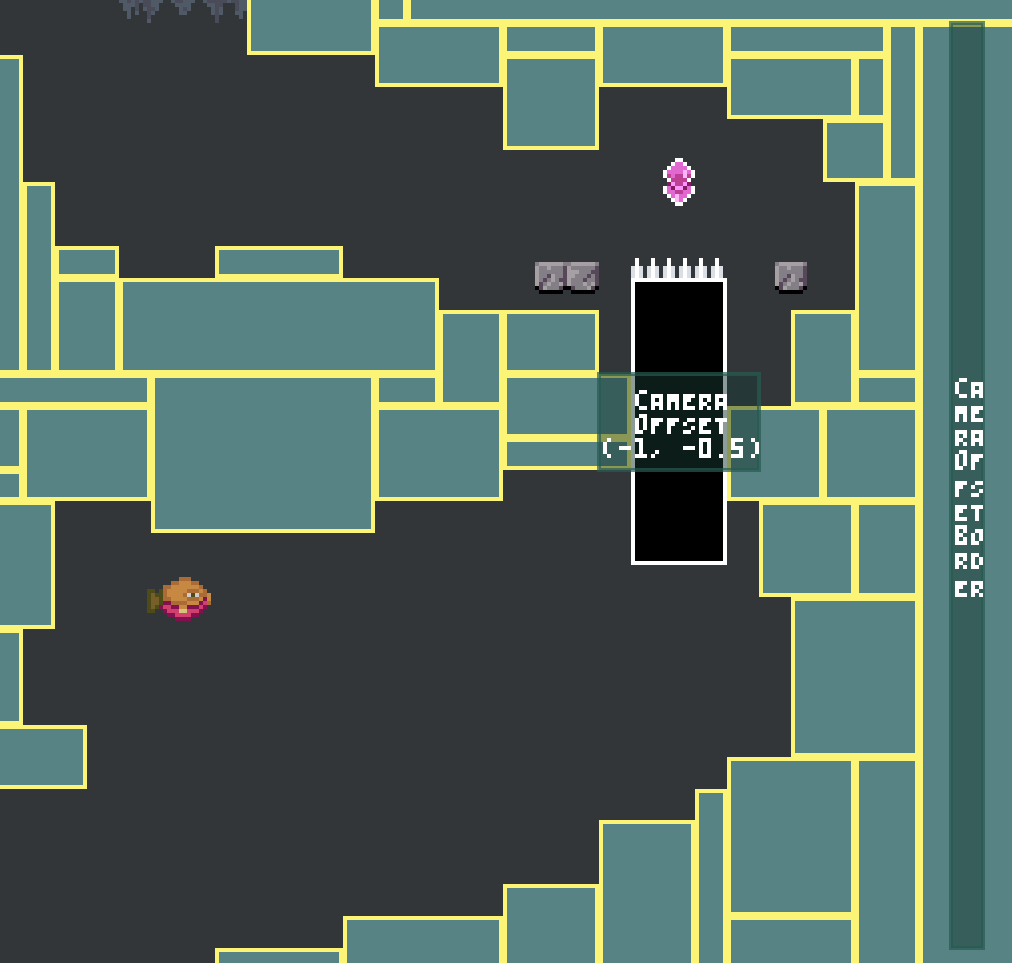


这是笔者某张图某面的部分镜头。这里的路线设计是十分明显的，由电网的限制可以明显看出来。

可以看出，offset-border系统没有很多复杂的trigger堆叠，通过在路线边界的位置设置border来限制镜头的移动，配合offset的调整让镜头的运动变的平滑。

具体的制作流程大致如下：

比如开头，玛德琳向右移动，因此我们可以在这里给一个x offset=2的镜头（图中未给出，笔者习惯第一个offset在房间属性中设置）。运动到河豚右侧时，此时需要上冲然后向左运动，因此右侧就没有必要给予视野了，因此笔者在右侧放置了一个border，并且只勾选了center left选项（如下图）。



此时，由于右侧存在border，镜头偏移虽然为x=2，但此时玛德琳仍处于屏幕右侧位置（如下图，图源吵吵金月视频）



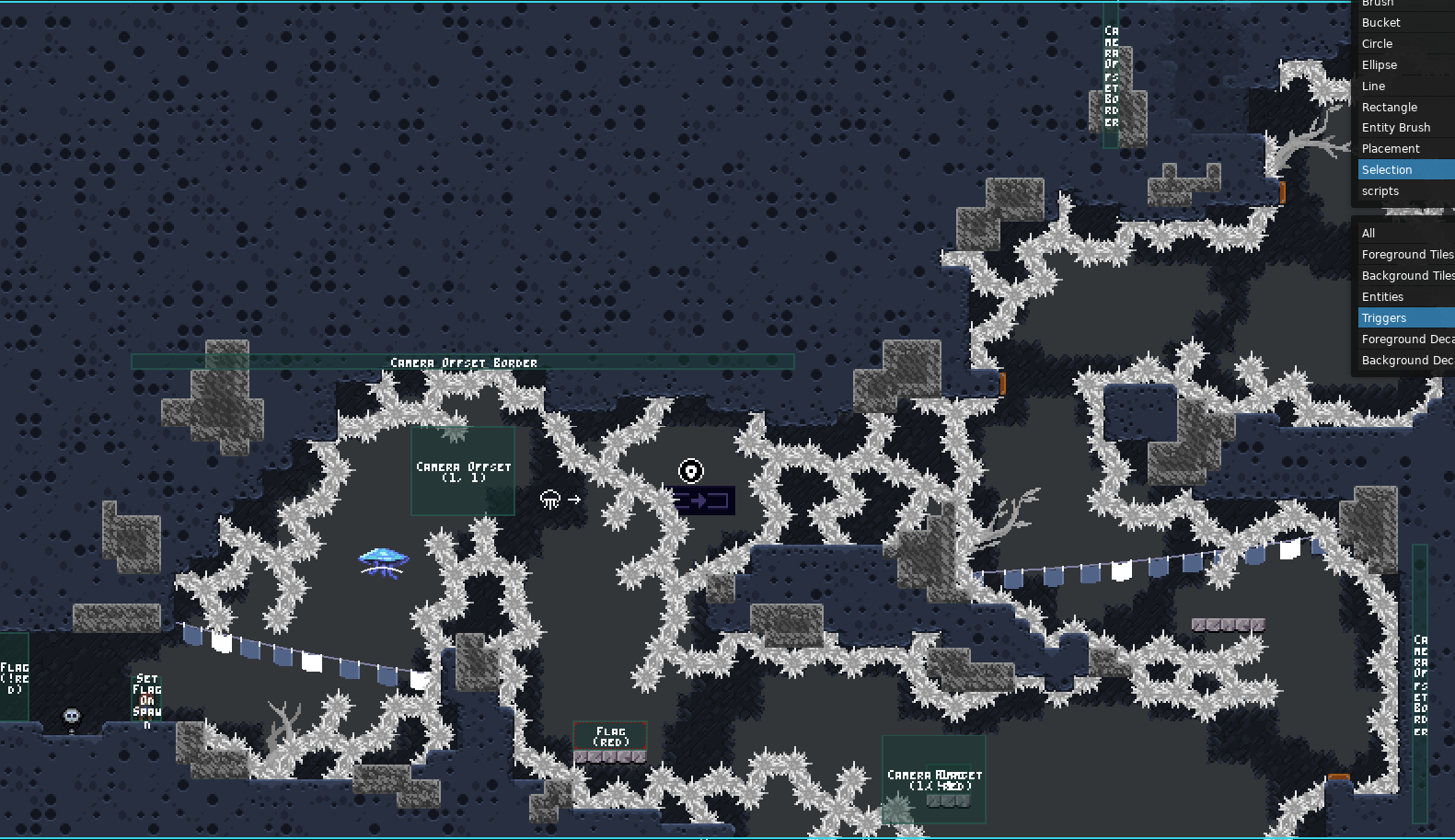
此时去改变x offset，只要偏移量不超过某个值，镜头不会发生变化。这个上限值与玛德琳与border的位置关系有关，具体情况下具体分析。笔者这里在上冲进果冻时将镜头x offset改成了-1，但镜头的x方向并没有变化（如下图），因为此时border仍在作用，x=-1的情况下border仍然在将镜头向左推。



后续的镜头设置同理，在转折处设置border，然后在镜头不发生变化的情况下更改offset，这种方式可以很大程度上解决镜头问题。

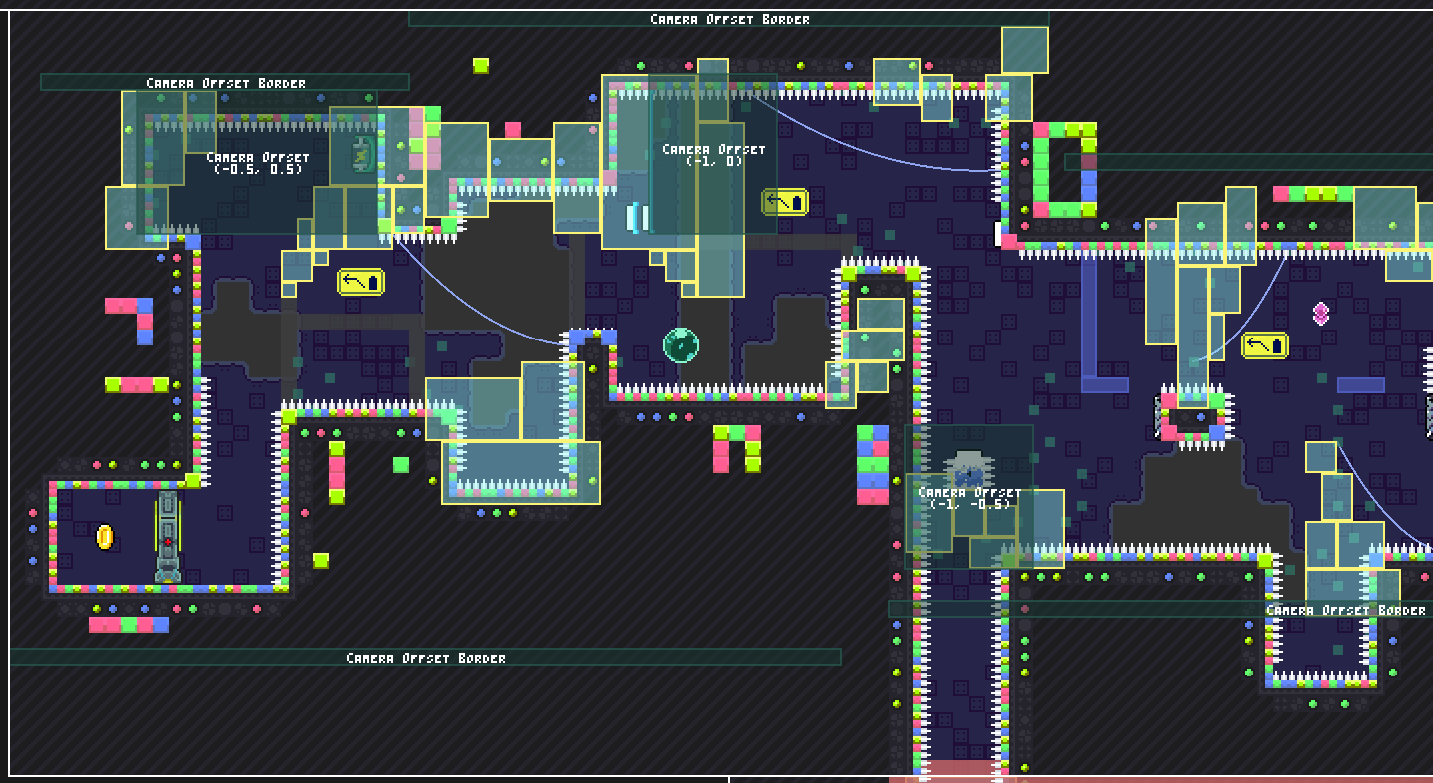
（注：很多读者可能发现了，上两张图在y方向的offset给的不是很够。实际上，在向上移动时笔者建议y offset给到-1左右，这样上方的视野才是足够的。）

以下是另一个例子：



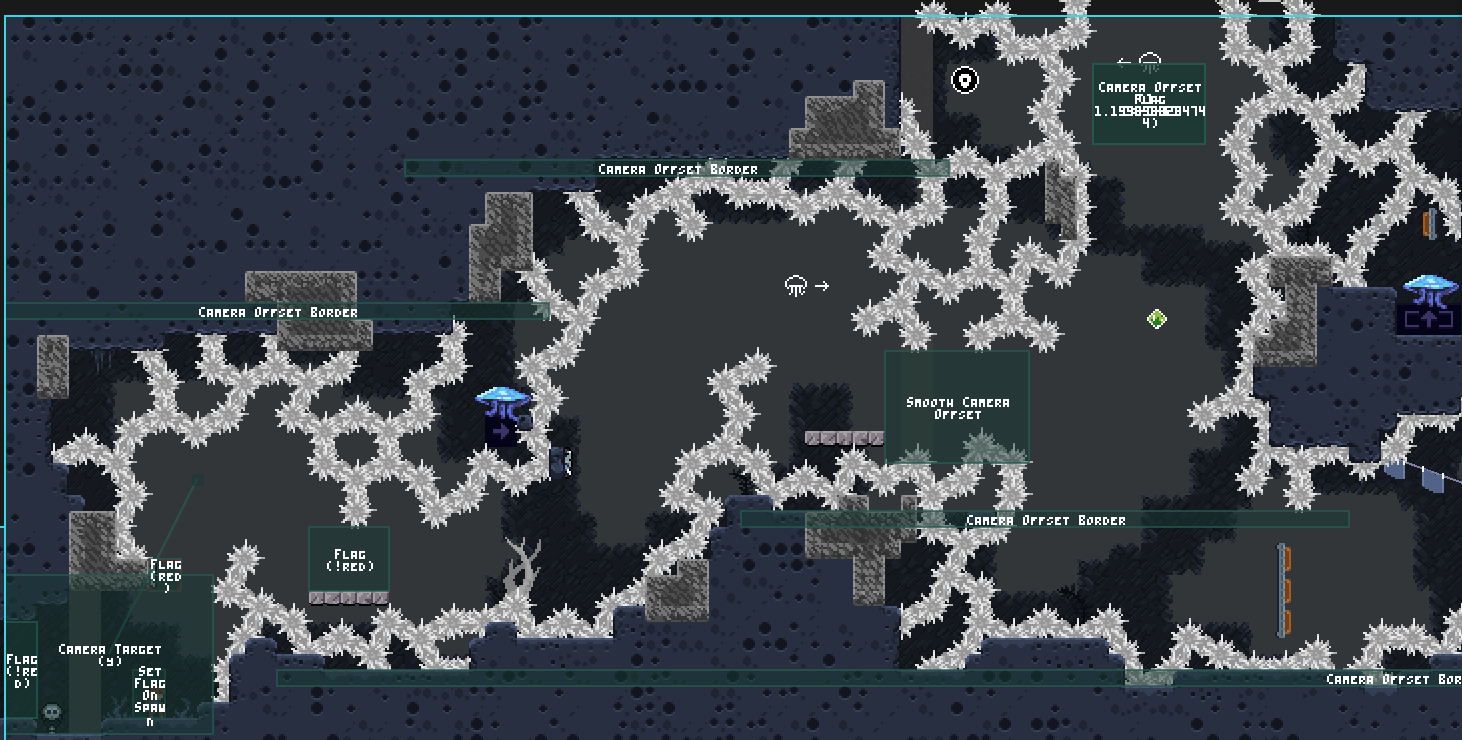
这图是画游的蓝莓果冻谷，这张图几乎所有的镜头都是使用offset-border系统完成的。如果对这一镜头系统比较感兴趣的话，可以去拆包看看这张图的bin，名字叫LittleV1.bin。

同样的举例：



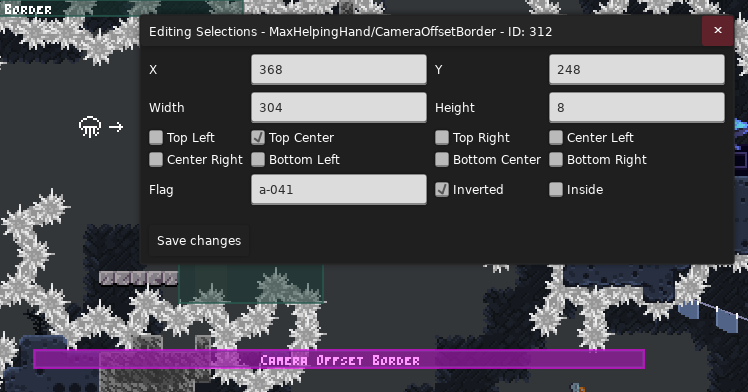
这图是画游的GDDN，斯塔在制作镜头时也使用了offset-border体系。不过他在部分房间的制作时还使用到了smooth camera offset，因为这图的运动相对较为复杂，仅仅使用offset-border体系有时无法保证镜头的稳定性。

3.引入flag



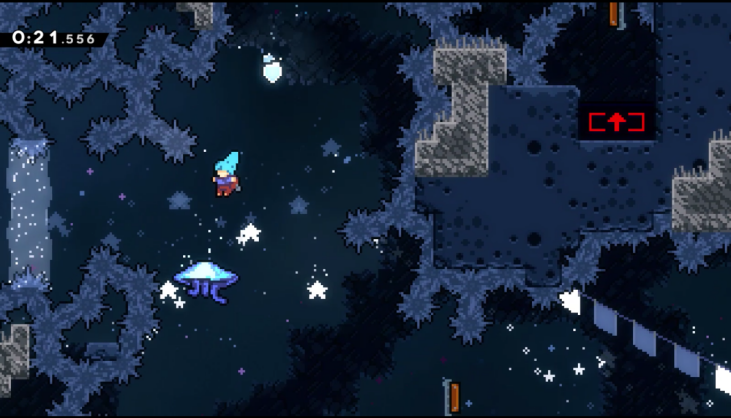
如上图，这里右侧需要先往上再向下，但中间往右时经历了一次下降，如何在这一过程中保证y方向的镜头可以看到要走的路呢？

这里就要使用到flag了。



如图，我们在这里放置一个border，并设置flag属性，勾选inverted。同时，在右侧最上方放置对应flag，勾选state（或使用temporary flag），因此玛德琳在运动到最上方抛出水母后，flag被触发，此时border失效，不再阻挡镜头下降。并在最上方设置一个y=1.5左右的offset，这样玛德琳下落时下方的镜头便可以看清。

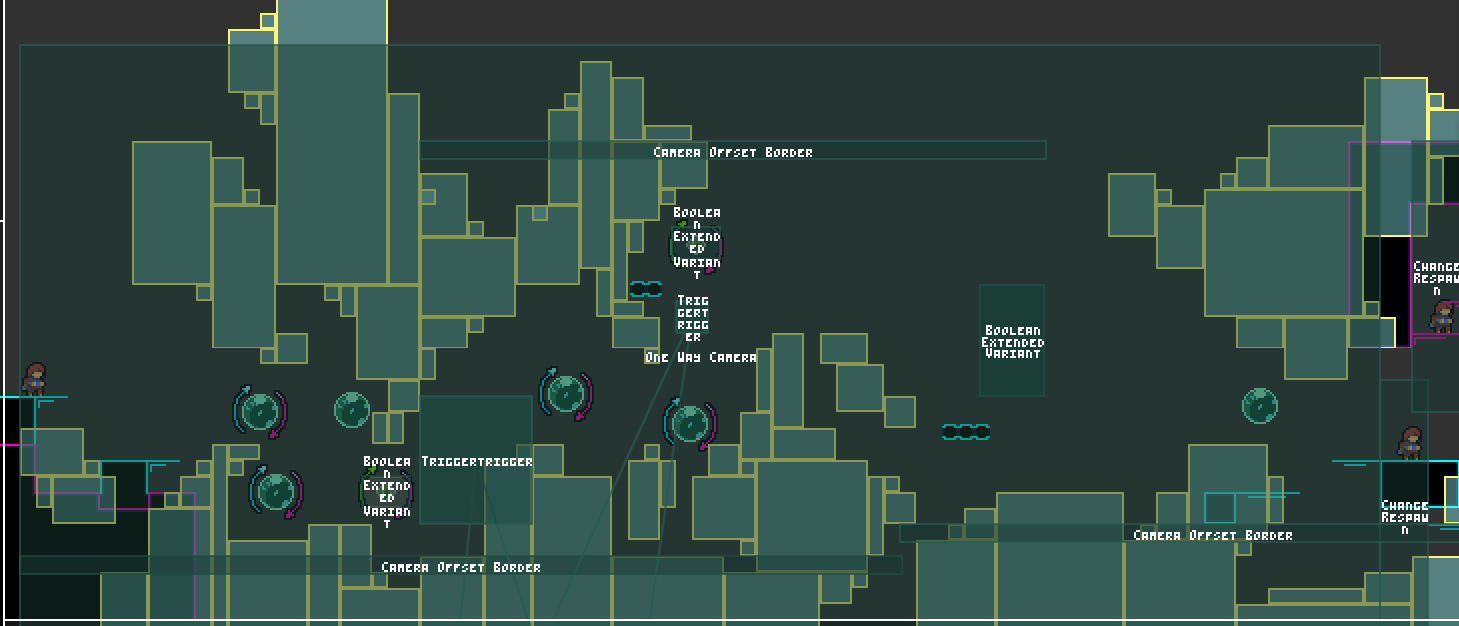
因此，这个border在第一次下落时会阻挡镜头下降，保留出上方的视野；第二次下落时border失效，不会影响镜头向下移动。（如图）



4.使用one way camera

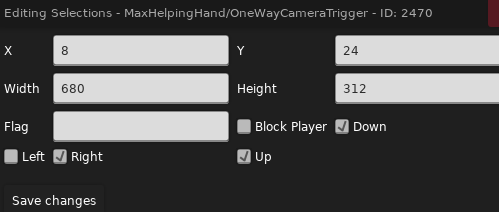
上述镜头使用基本涵盖了大部分可能的使用情况，但仍然存在一些情形：整体的移动趋势是向右的，但是存在部分向左的小幅度移动。但我不想镜头左右晃，有什么办法吗？

这里就要使用one way camera了。



上图是负负得正某面的镜头。排除一些奇奇怪怪的效果trigger，该面只使用了四个镜头trigger，其中三个为border，最后用一个one way camera覆盖整个房间。

One way camera的页面如图所示：



flag不做说明，主要看down,left,right,up四个方框。每一个方框勾选后，意味着在该trigger范围内允许对应方向的镜头移动。例如这里勾选了down,right,up三个框，意味着玛德琳在这个范围内移动时，镜头只会向右、上下移动，而不会向左移动。这种设置可以防止玛德琳在小幅度左移（如上图中的泡泡冲刺，反凌波等）时镜头左移发生晃动，一定程度上可以维持镜头的稳定。

另一种使用方式是：如果玛德琳的运动大致在某个高度范围内，可以使用one way camera，仅勾选left和right框，这样可以使得玛德琳在进行小幅的上下移动时，镜头不会发生上下晃动。由于示例地图在笔者撰稿时尚未公开，因此就不做举例了。

One way camera 还有一些妙用，但由于其使用与地图路线关系很大，这里不再赘述。

5.小结

综上，我们已经了解了camera-border系统的大致使用思路。这一系统在处理非高速图时可以起到镜头的基本作用，给予玩家足够的视野，但在维持镜头稳定性上，需要一些额外的镜头进行补偿。

镜头的制作不是一簇而就的，笔者也经历过镜头被其他人批判的情况，例如2023年春贺的新年塔。笔者想表达的是，其实镜头的制作是在不断的实践中优化的，愿意听取测图人员的意见，愿意不断优化调整自己的镜头，未来一定能做出优秀的镜头。镜头的制作也不会局限于某一种“范式”，针对某一具体路线，有自己的镜头设计方法。camera-border系统只是提供了一个相对简便且有效的镜头制作方法，具体使用还得视情况而定。

祝大家的镜头水平越来越好！