涉及obj

objDayNight - 主obj，控制global.dayNight的自增，并绘制日夜化后的表面

objTorchLight - 火炬，夜晚点亮白天熄灭

objWGhost - 幽灵鬼，见详述

objClock - 根据global.dayNight绘制钟楼显示的时间

objGhostBorder - 鬼碰到该obj会掉头

objGhostTrigger - 鬼碰到该开关会控制对应的门（就是开门或关门）

objBlockHallow - 给零梦用的砖，当objBlock看就好，没什么特别的

objDoor - 见详述

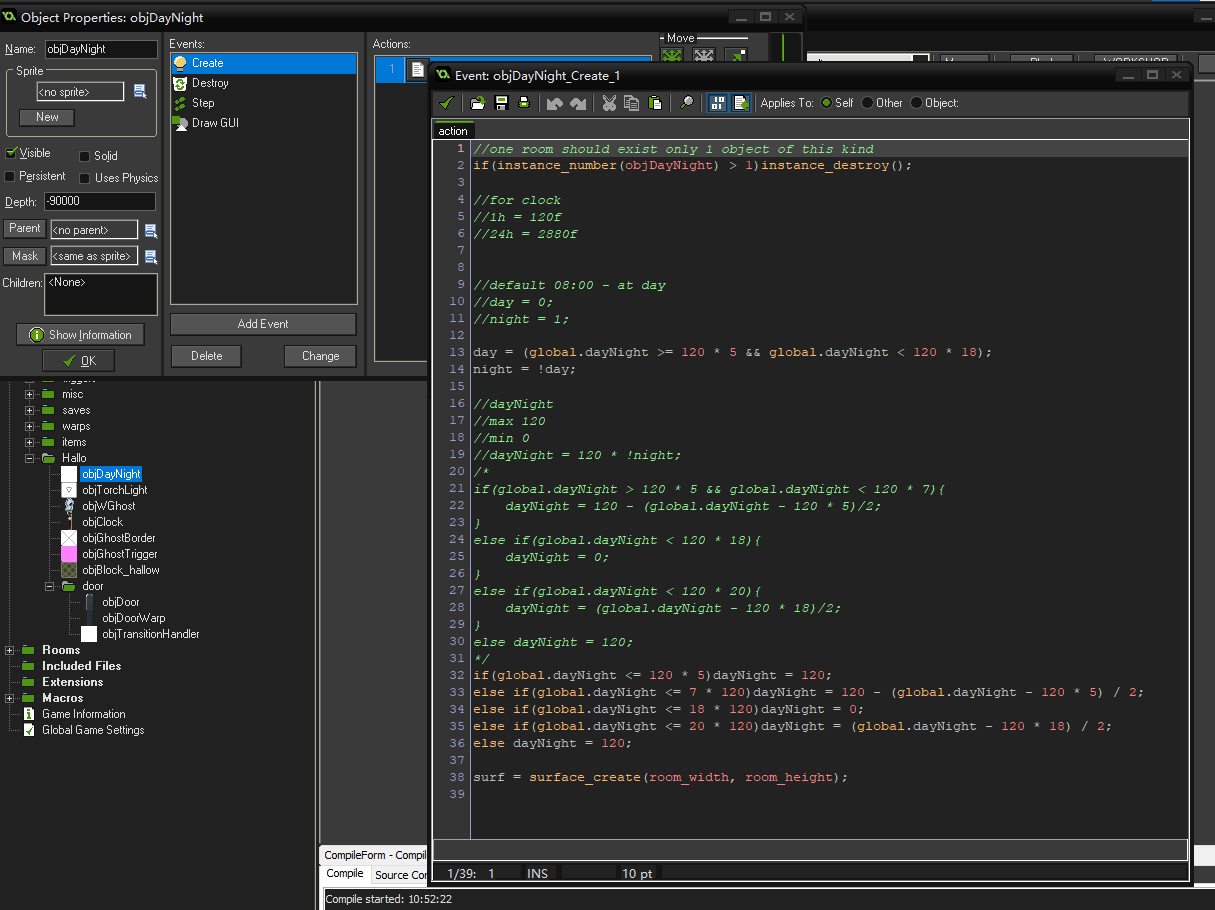
objDoorWarp - kid碰到objDoorWarp之后播放特效，以作出“进门”的效果

objTransitionHandler - 处理kid进入objDoorWarp之后的房间切换特效

详述

objDayNight

Create



它是单例：整个游戏中只能同时存在一个（如果不是单独游戏的话需要新增房间判定来清除该obj）

day, night: boolean flag，用来指示当前日夜状态

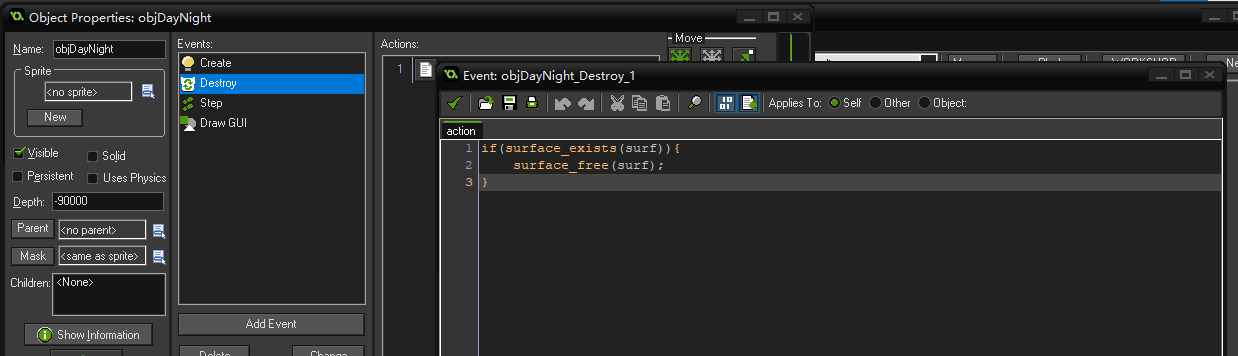
（例如彻底切换到夜晚之后day = 0, night = 1）

dayNight: 日夜切换进度（dayNight值最大120，此时滤镜效果拉满，dayNight值为0时滤镜失效，obj正常绘制）

surf: 用来绘制日夜化后的房间的表面

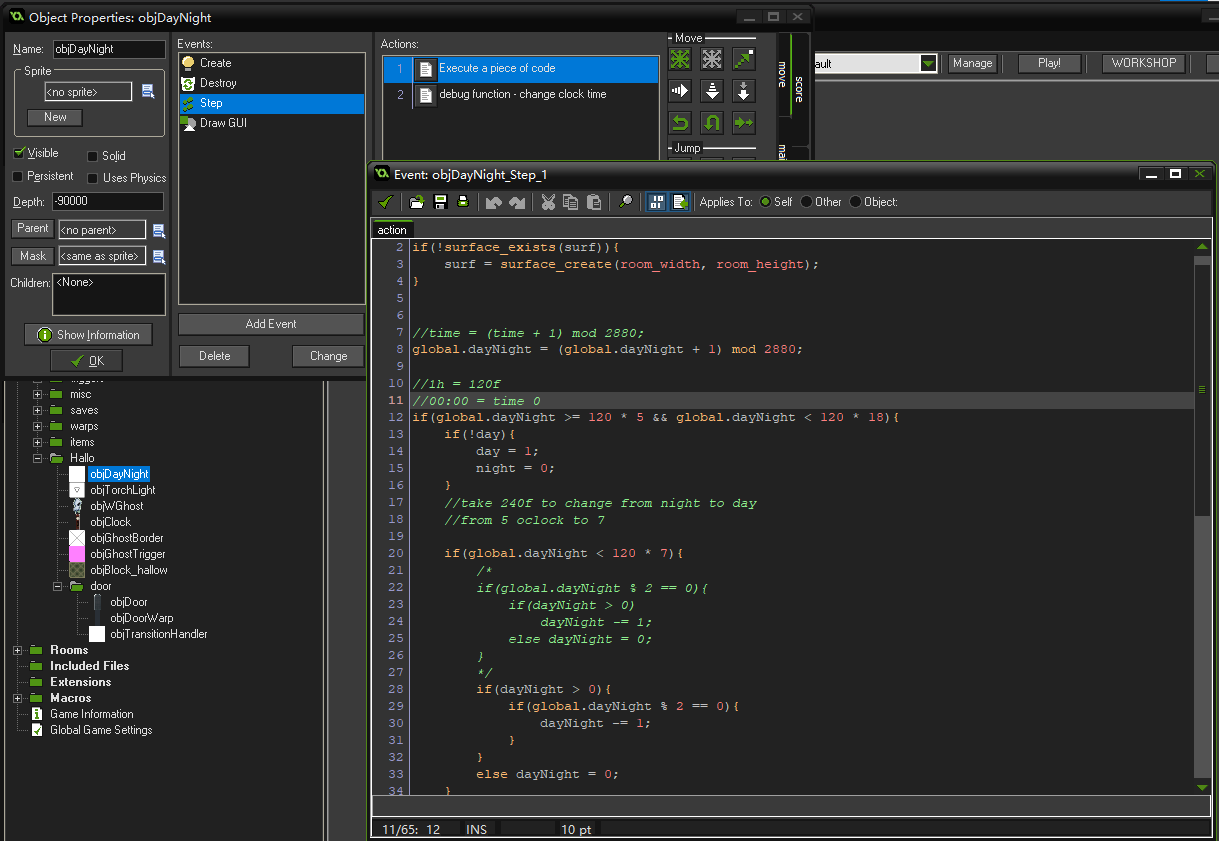
//表面的深度取决于绘制它的obj的深度，所以这个surf在整个stage都应当是置顶的

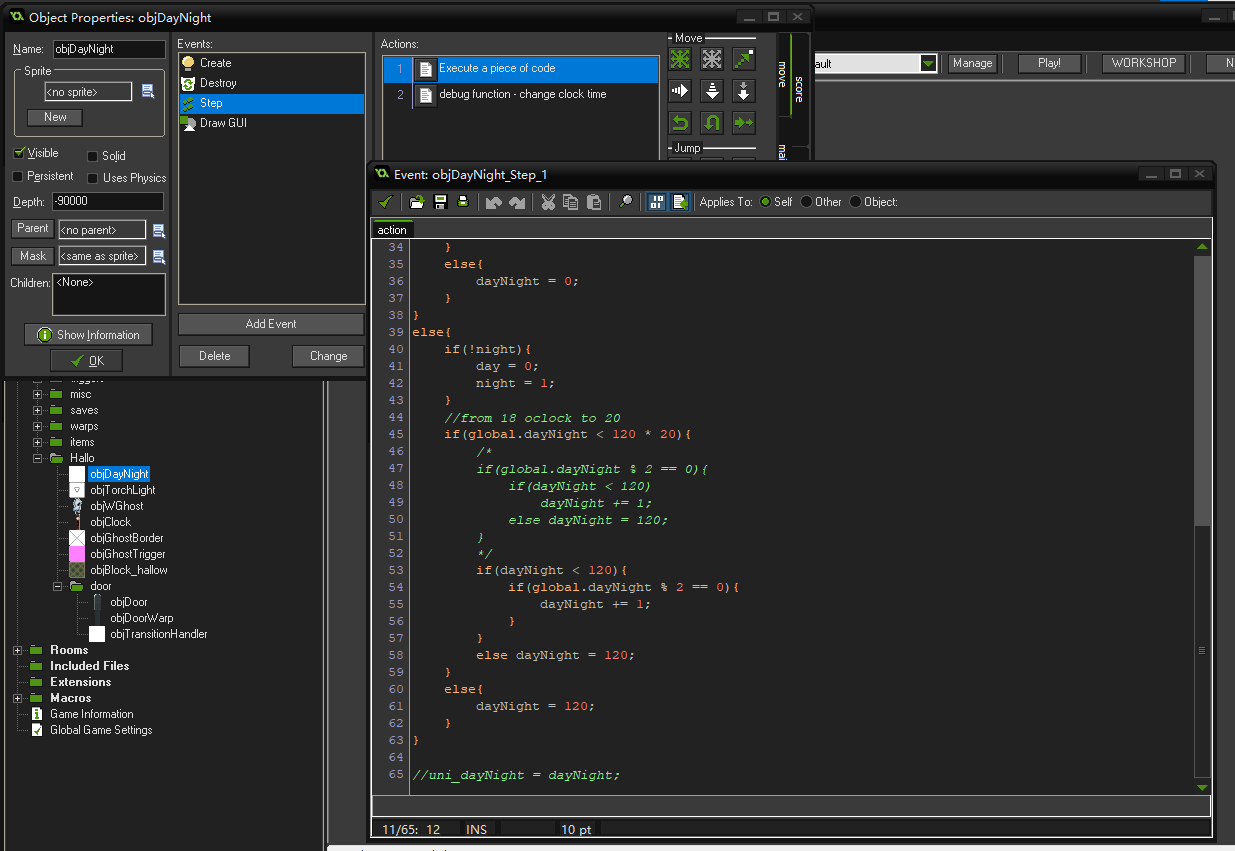
Destroy



没啥好说的，去除surf 表面

Step





首先确认surf未损坏（似乎不检测也可以，但不知道会不会有什么bug）

然后每“小时”为120帧 = 2.4秒

“一天”的时间为120 \* 24 = 2880帧

5-7点： dayNight 从120每2帧-1，最低为0

dayNight值会控制Draw GUI中的滤镜强度，最高120，最低0

7-18点： dayNight稳定为0

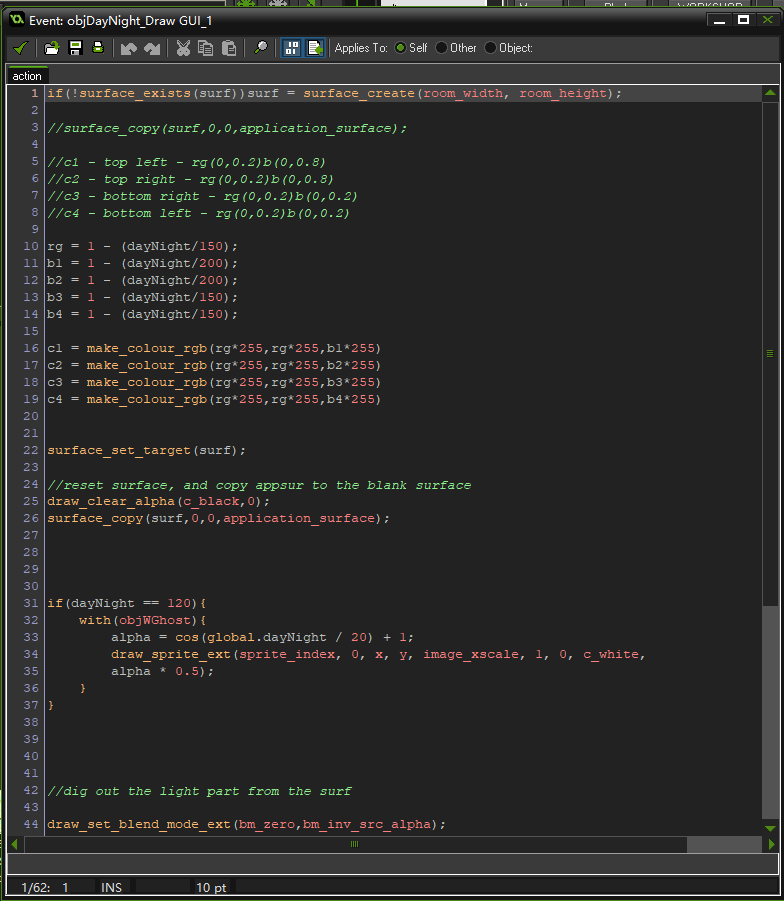
18-20点： dayNight从0每2帧+1，最高120

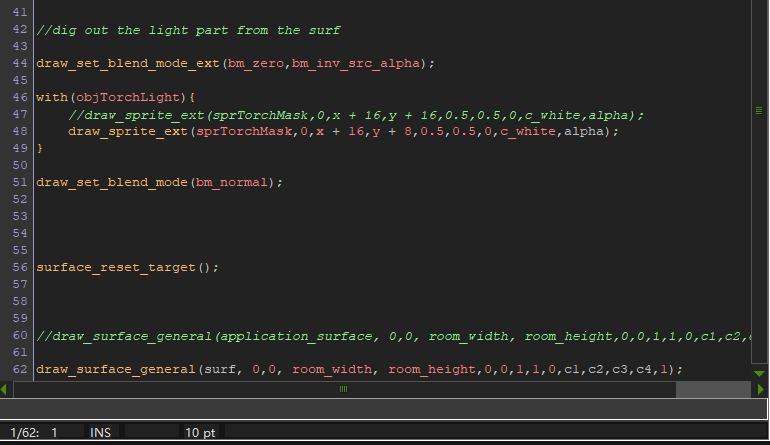
20-5点： dayNight值稳定为120

debug function中可以通过翻页键来控制global.dayNight值前进或拖后1“小时”

实际游玩的时候可以清除，或者可以加个时光机什么的？？

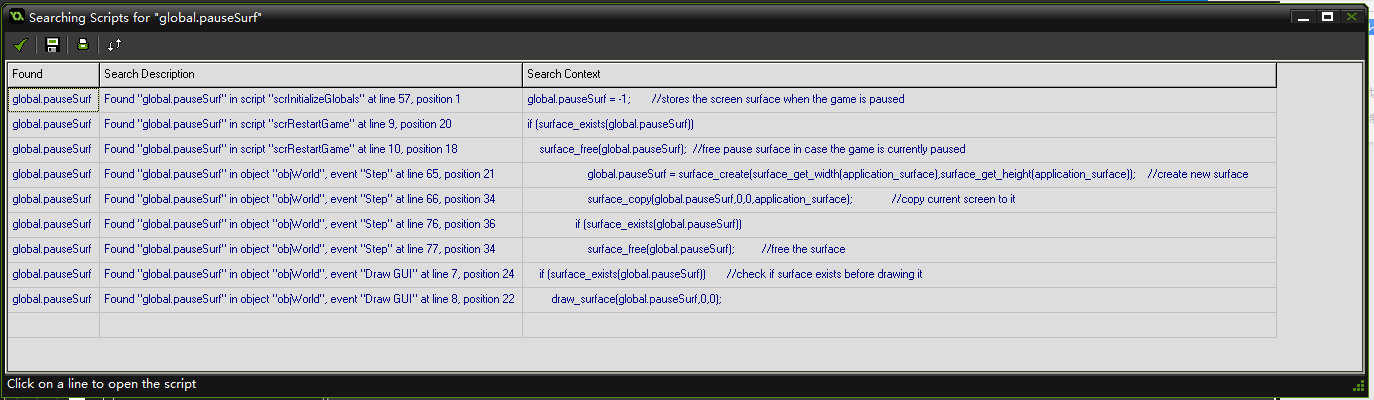
Draw GUI





确保surf未损坏

**注意：gms下如果更改窗口大小的话会损坏表面！当时未发现该问题，建议参考objWorld中暂停表面的机制**



rg 为对应的红、绿值的比例

b1,b2,b3,b4分别对应的是左上、右上、右下、左下四个位置的蓝值

c1,c2,c3,c4分别对应的是左上、右上、右下、左下四个位置的实际颜色

开始更新表面

先清除表面为透明

然后把当前的application\_surface给复制到surf上（application\_surface 就是游戏每帧默认呈现的那个表面）

如果已经彻底转换为夜晚的话在幽灵的位置直接绘制随global.dayNight值变动绘制的image\_alpha的幽灵贴图

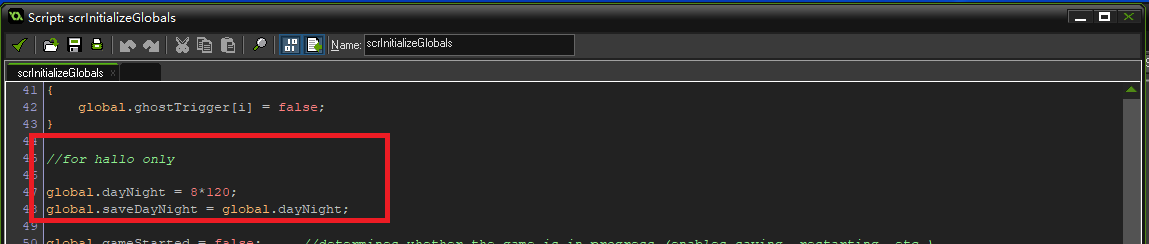
然后让每个torchLight 以它的火焰中心为中心从app\_surf挖出sprTorchMask形状的图片（即“照明”效果）

更新表面完毕，把surf绘制到房间内

备注：

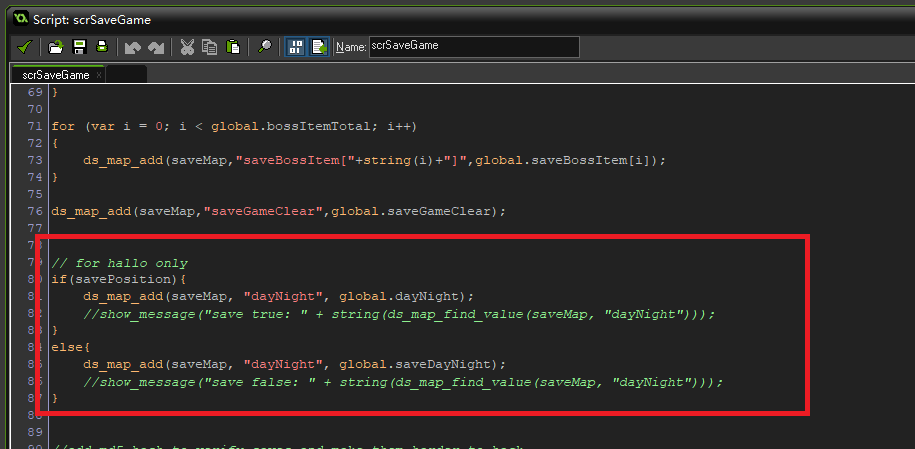
在如下原生脚本中有修改：

scrInitializeGlobals



如果初次进入该stage默认时间是早上8点

scrSaveGame



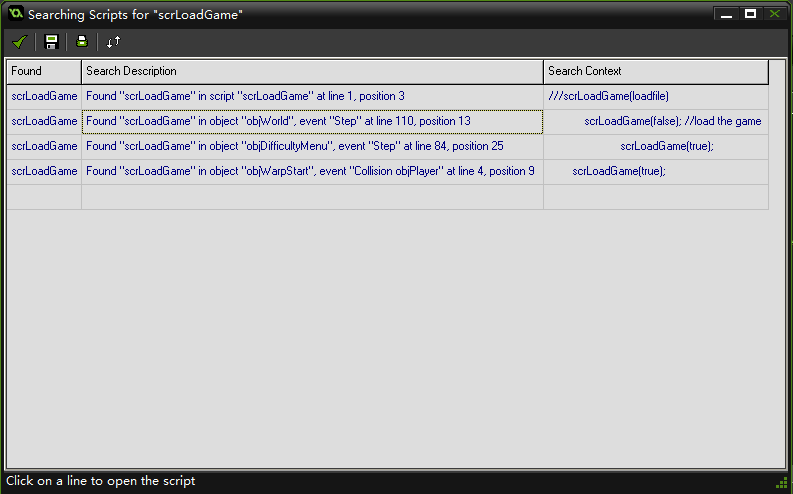
如果是save触发的保存游戏的话则保存当前的global.dayNight

否则保存上一次进行保存时或初始的global.dayNight值

该机制旨在确保

1. 按r不会影响日夜切换和“时间”自增
2. 重新进入游戏的时候“时间”不会被重置

scrLoadGame



如图，仅当重新进入游戏的时候才会呼叫scrLoadGame(true)，按r只会呼叫scrLoadGame(false)

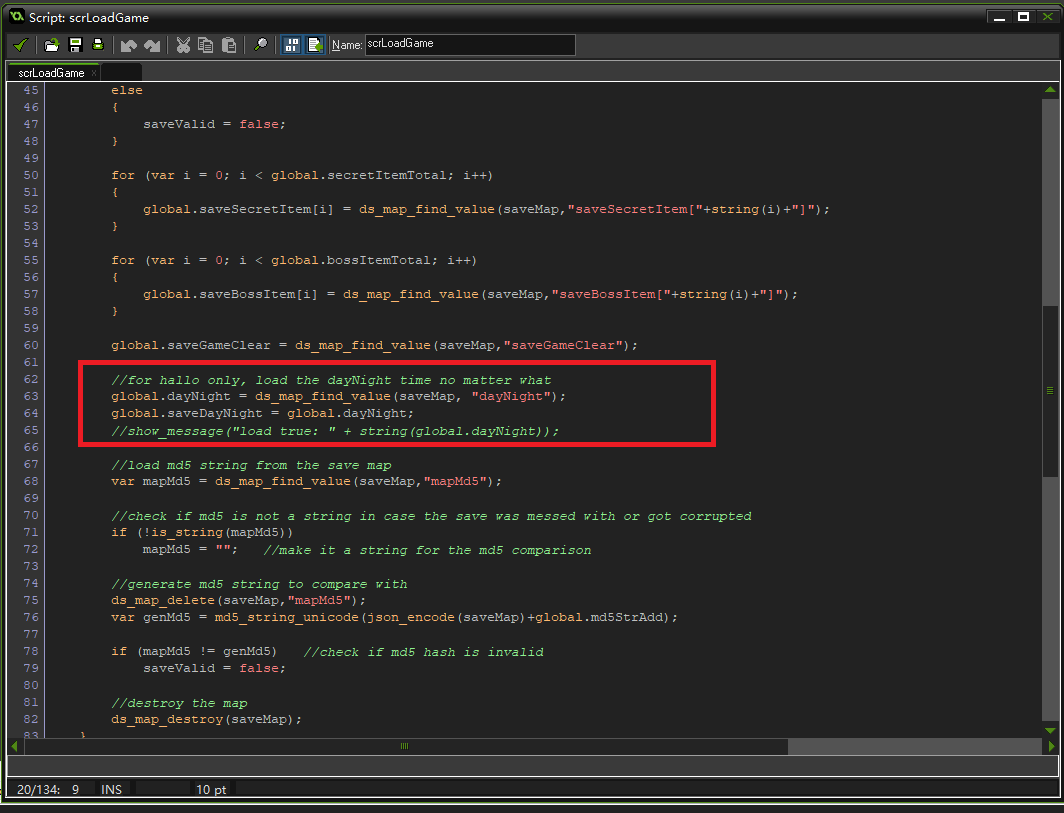
scrLoadGame的传入t/f值会赋予里面的loadFile变量

如果loadFile为true则会读取save文件的信息，否则只会读取上一次保存时更新的信息

上一次保存时更新的信息为global.savexxxxxxxx

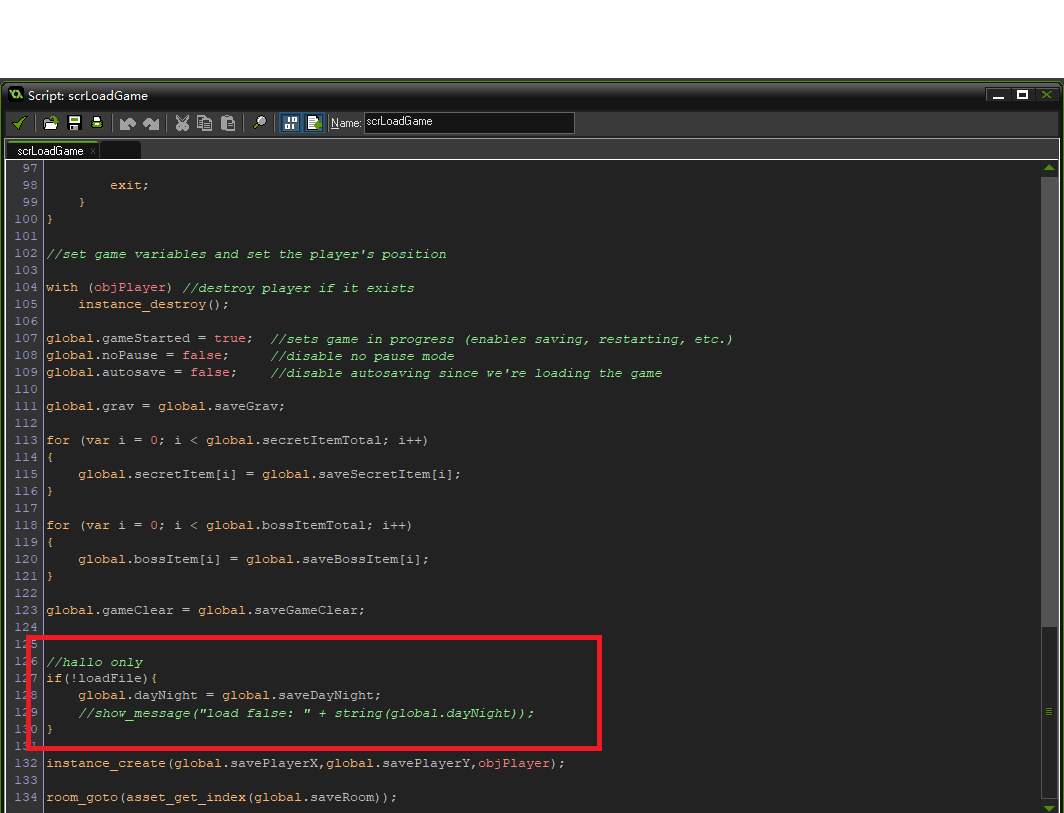
save文件的信息为global.xxxxxxx

在yoyoyo的结构中变量的储存和读取一般是由一对变量实现的



这里的代码是放在if(loadFile) 中

也就是说每次进入房间读档的时候都会读取save文档中的dayNight值并更新对应的global.saveDayNight值

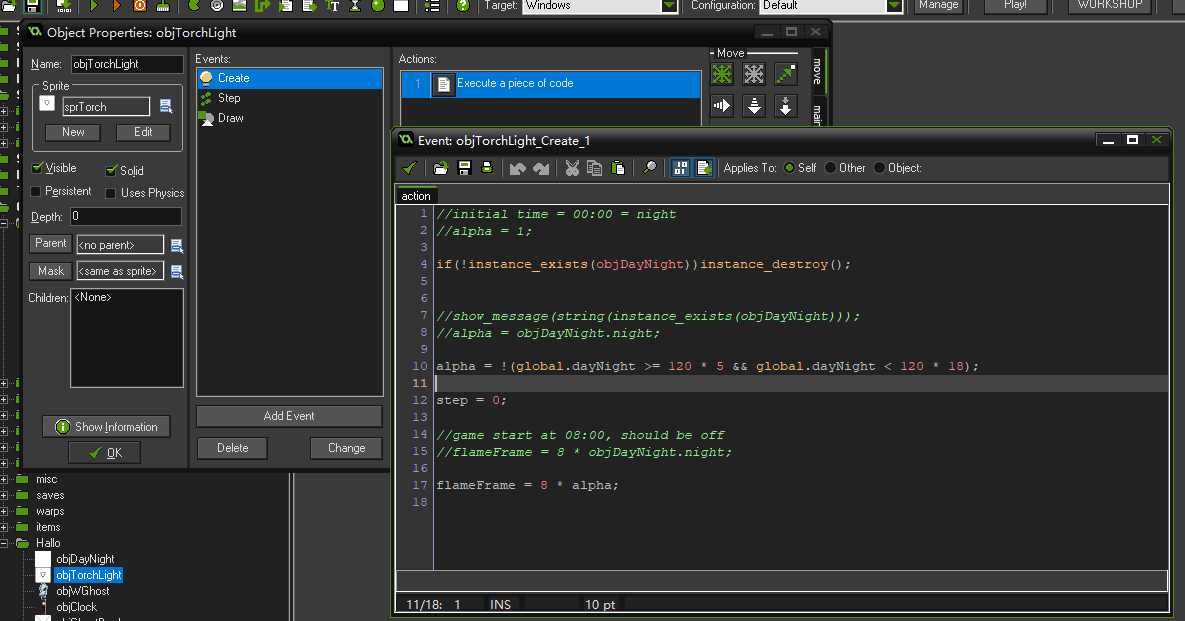


这里的代码是放在if(loadFile)外

用意是重新打开游戏读档后更新global.dayNight为上一次存储的global.dayNight

objTorchLight

Create:



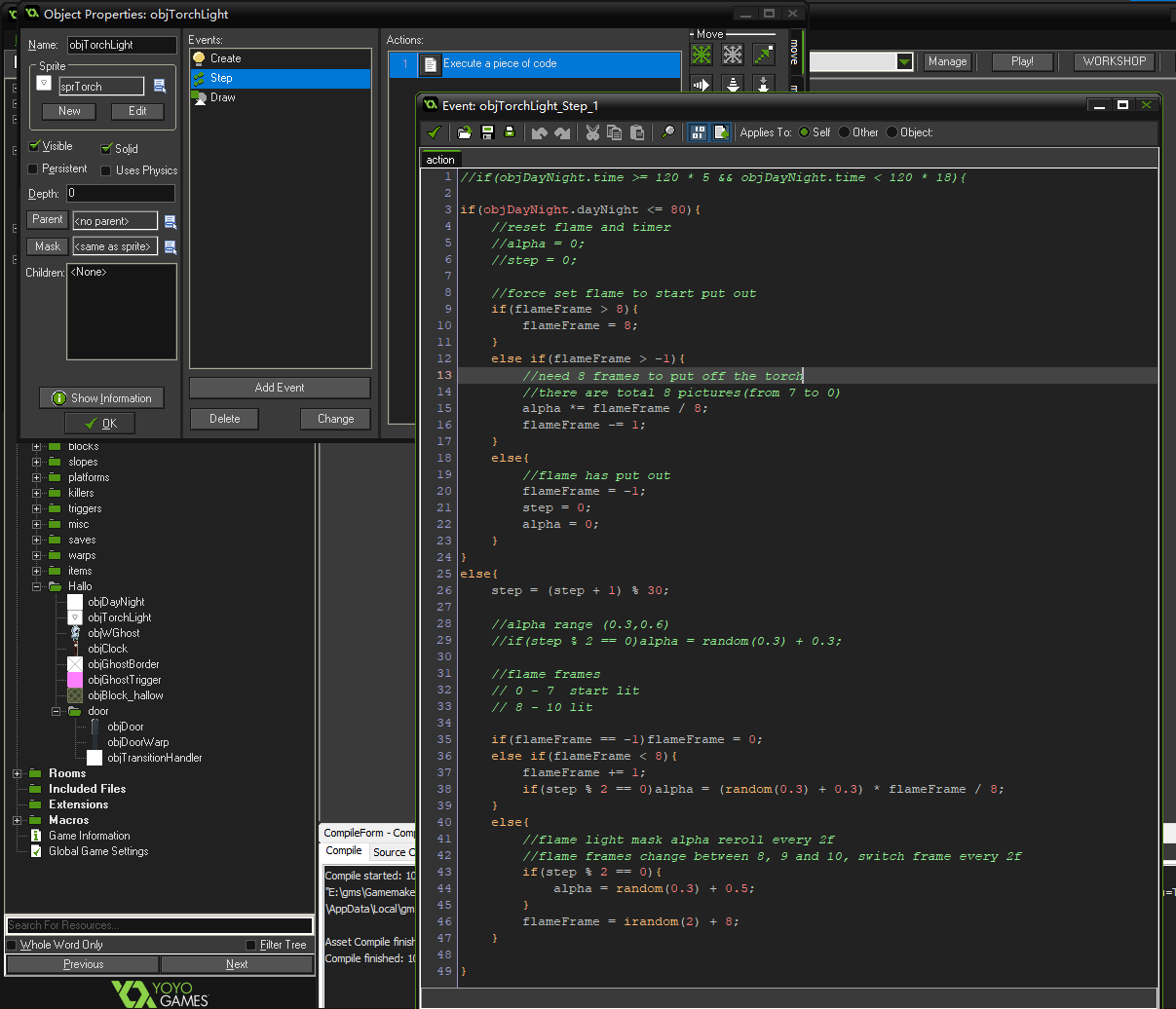
如果objDayNight不存在的话objTorchLight将自毁

alpha为一个判定是否为夜晚的flag

step 控制objDayNight的Draw GUI中绘制sprTorchMask的透明度以及火焰显示哪一帧贴图

fireFlame为当前火焰贴图的帧序号

Step:



objDayNight的dayNight值不大于80（即大概“早上5点40分”）时

开始“扑灭”火焰

sprFlame的绘制帧fireFrame从当前帧往0号帧进行每帧帧号-1的回退

alpha->objDayNight中绘制sprTorchMask的image\_alpha根据fireFrame更新

直到fireFrame 降为 -1 时，设置step = 0, alpha = 0

否则

如果fireFrame = -1 则它未“点燃”

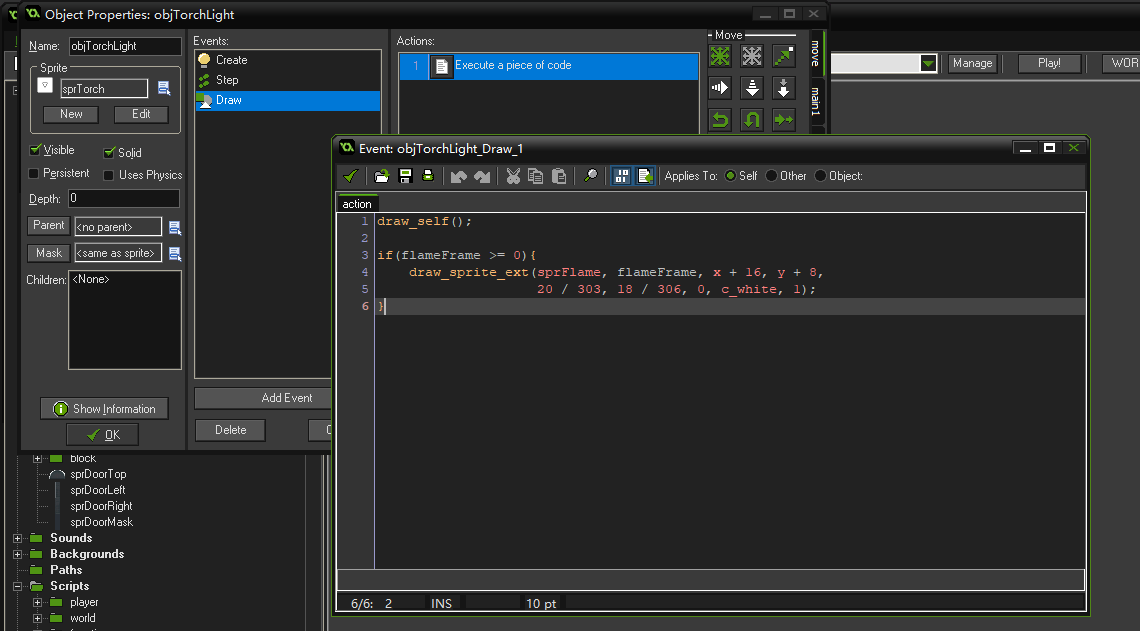
所谓“点燃”就是设置fireFrame = 0

否则

fireFrame每帧+1（“点燃”效果）

当fireFlame达到8以后它的值每帧都随机在[8,10]∈Z之间变换

Draw



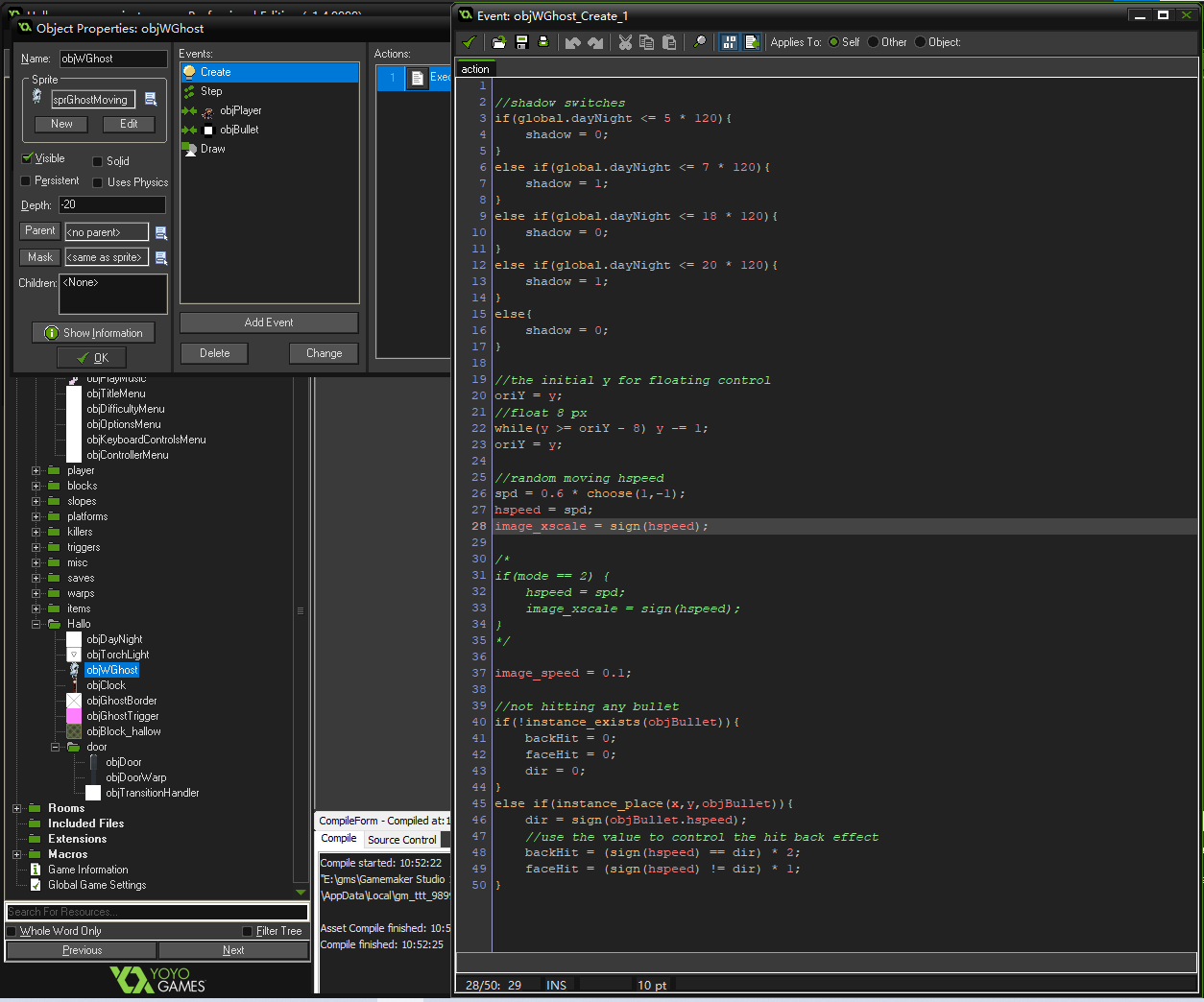
绘制火炬的底

然后如果fireFrame >= 0，即已经“点燃”

在适当的位置绘制火焰贴图

objWGhost

Create



shadow是指示幽灵是否开始在阴影和幽灵之间切换的flag

在objDayNight开始日夜切换时shadow = 1

其它时候为0

幽灵设定中是在不断上下浮动的

而oriY就是这个上下浮动的中间高度

create事件中其实可以直接指定oriY = y，不过不确定会不会有奇怪的特性

spd 决定幽灵的初始前进速度和方向

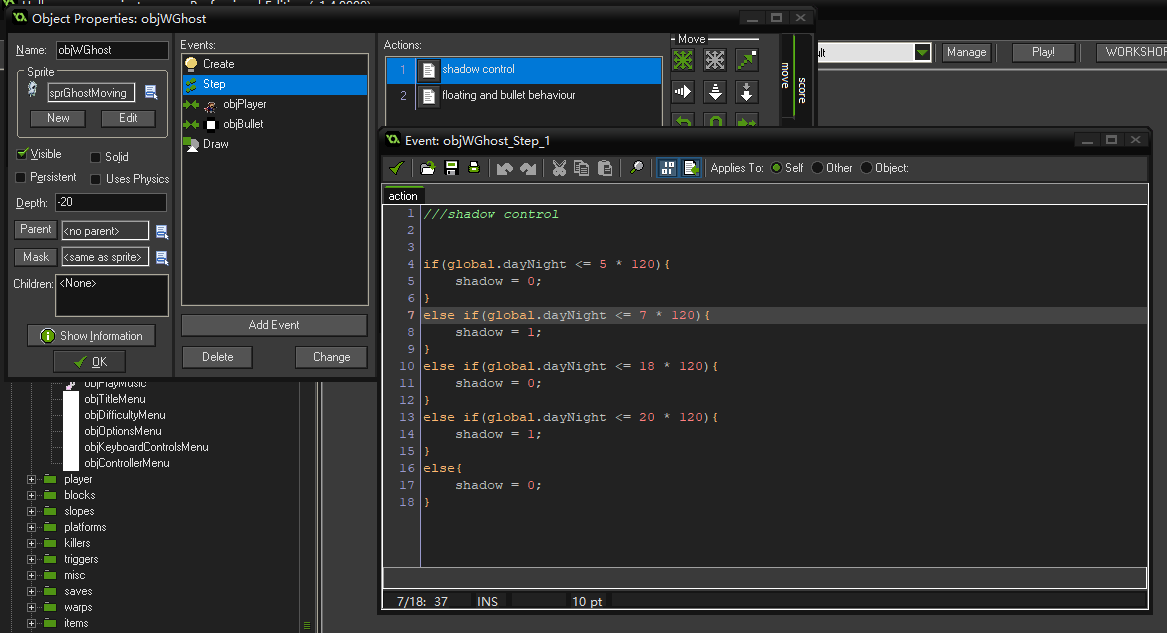
幽灵的朝向根据hspeed决定

backHit和faceHit存储当前幽灵是否正面中弹以及背面中弹

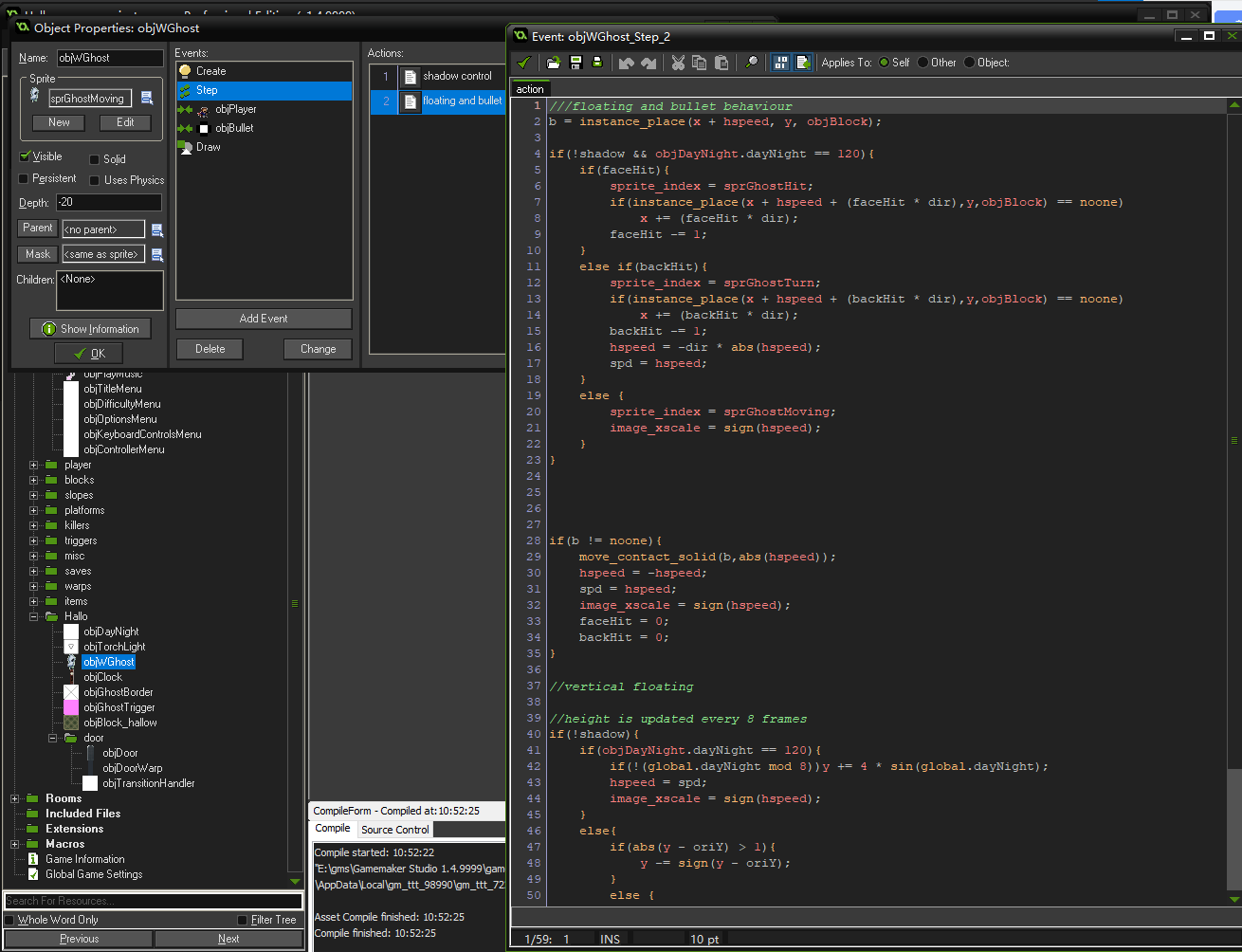
dir是当前子弹的朝向（这里假设所有子弹都是同向的，但如果po的话可能有问题）

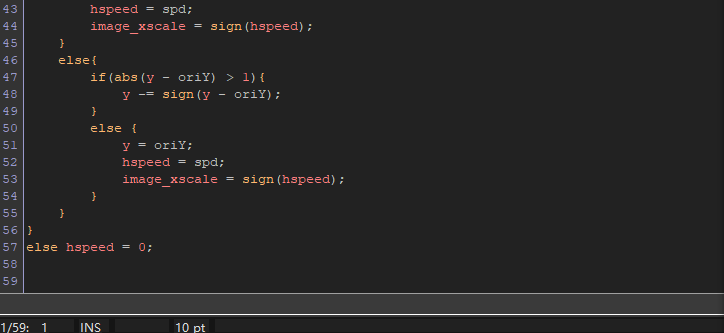
如果创建时就中弹则更新backHit 和faceHit的值

Step



同create事件，根据global.dayNight决定是否要切换幽灵的状态





逻辑概述

幽灵的高度每8帧更新一次，然后总体是一个上下循环浮动的效果

如果shadow = 1则幽灵静止不动

如果在夜晚幽灵正面中弹

贴图改为sprGhostHit

后退一小段距离

如果在夜晚幽灵背面中弹

贴图改为sprGhostTurn

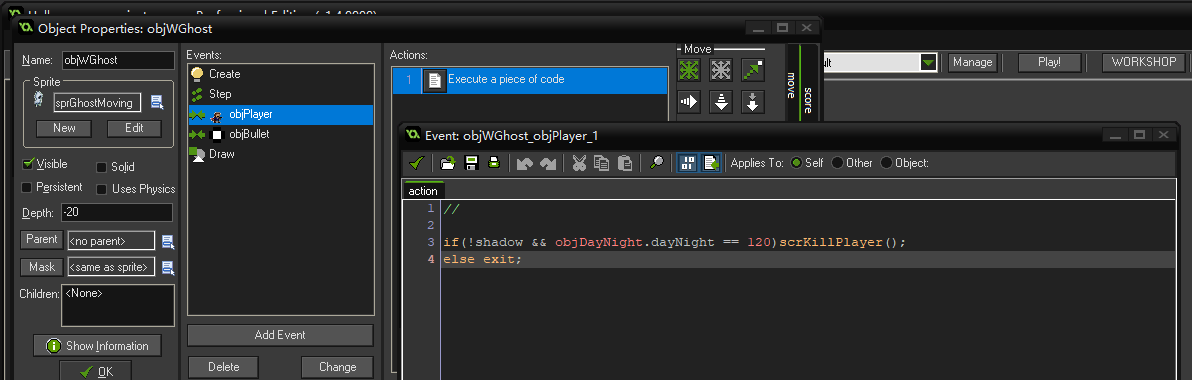
并往原方向后退一段距离

再转换hspeed的方向以及更新spd

否则幽灵在水平方向上一直是sprGhostMoving贴图面朝hspeed的方向匀速运动

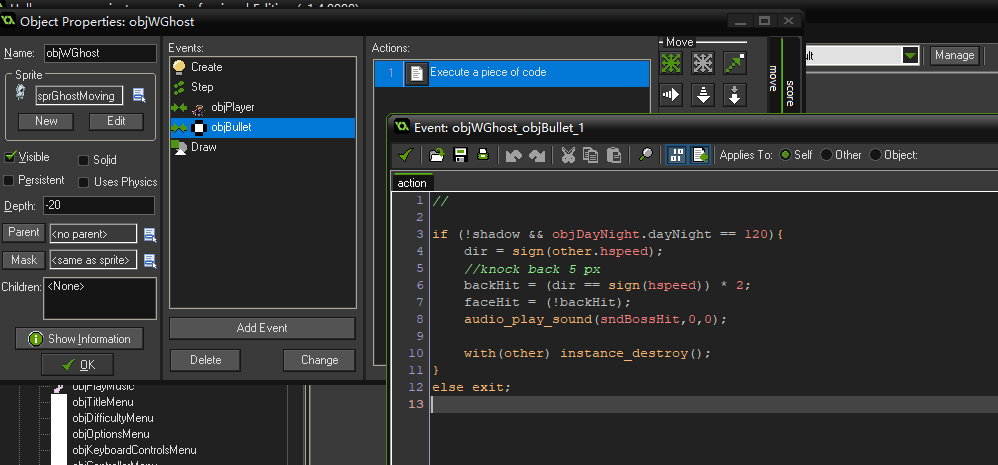
如果幽灵碰到砖则直接掉头，不经过sprGhostTurn的过渡

碰 objPlayer



如果已经完全切换为幽灵则是刺，否则无任何判定

碰objBullet



如果已经完全切换为幽灵状态

dir更新为子弹速度的方向

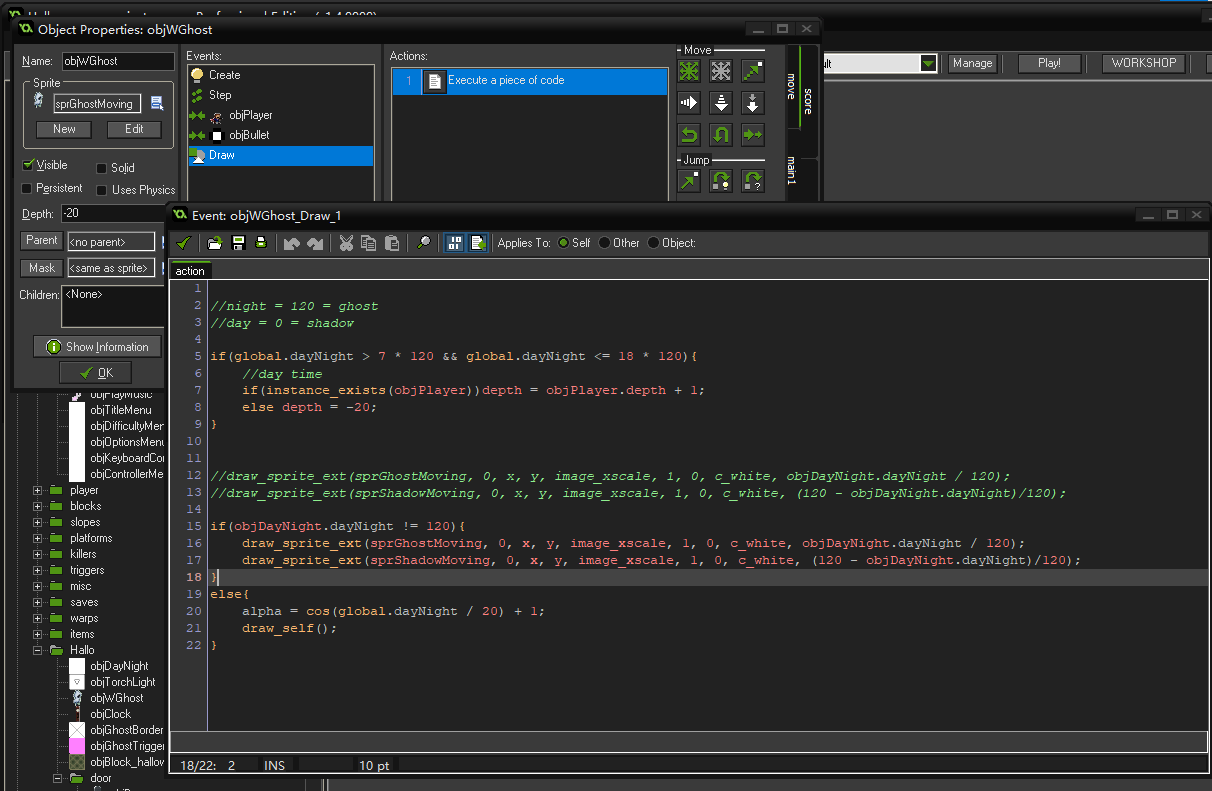
根据子弹的方向和当前幽灵的方向确定其实正面还是背面中弹

并更新faceHit和backHit值

播放中弹音效

清除子弹

Draw



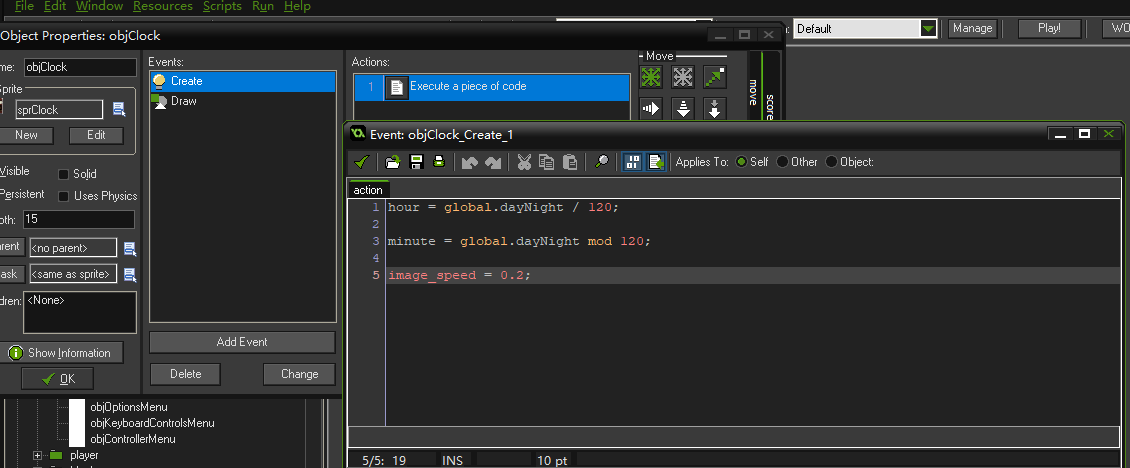
在白天的时候幽灵的深度应该在kid“下面”

在其它时候幽灵的深度-20（在kid“上面”）

然后就是根据objDayNight的dayNight值来控制两个幽灵贴图的渐变绘制，以及objDayNight的Draw GUI中幽灵的image\_alpha值（注意，幽灵的贴图是objDayNight特地管的）

objClock

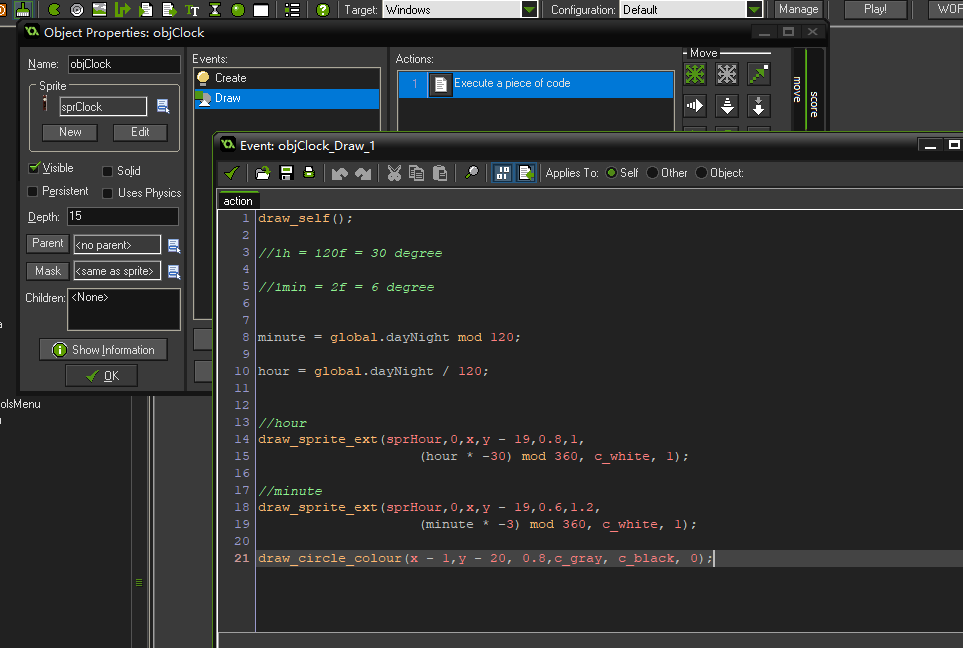
Create



根据global.dayNight确定对应的“小时”“分钟数”

以及钟摆的“摆动速度”

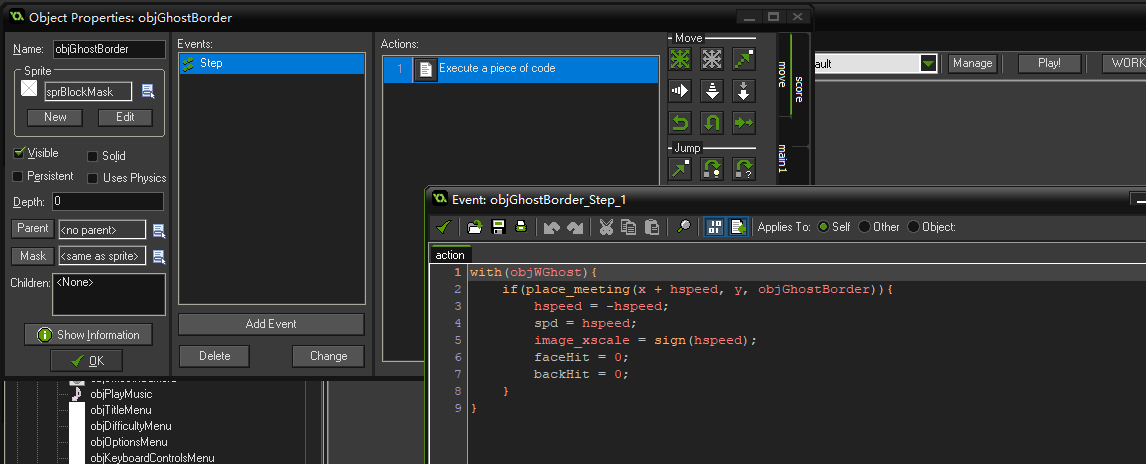
Draw



绘制objClock贴图（钟摆以及钟表的表面）后

根据“小时”“分钟”绘制时钟的指针

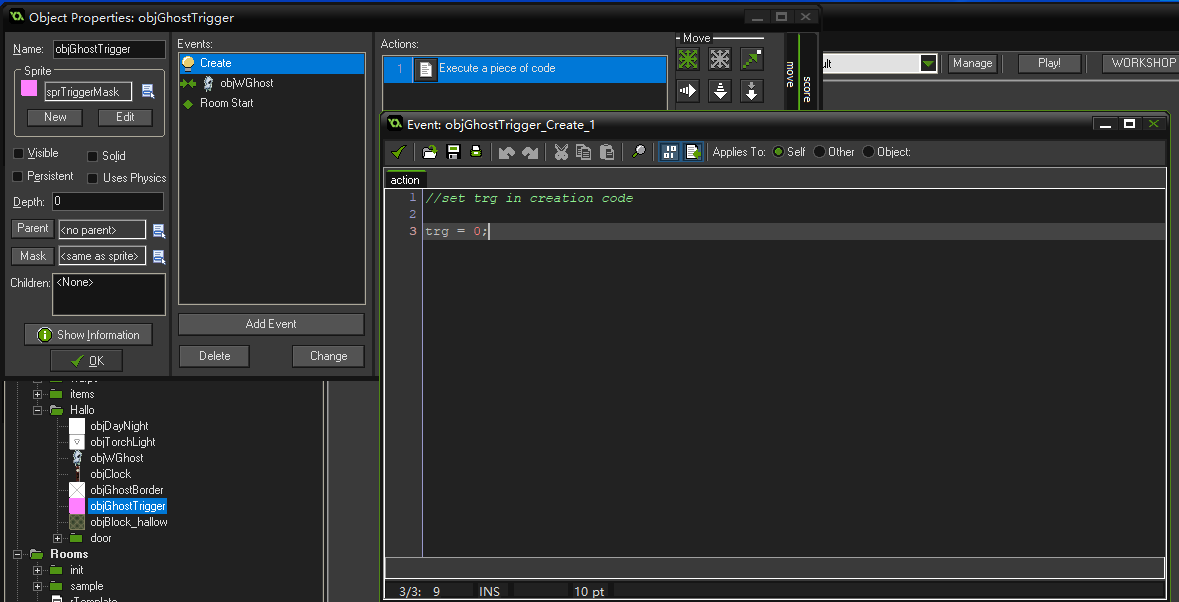
objGhostBorder



幽灵碰到它转向，但**不是砖**

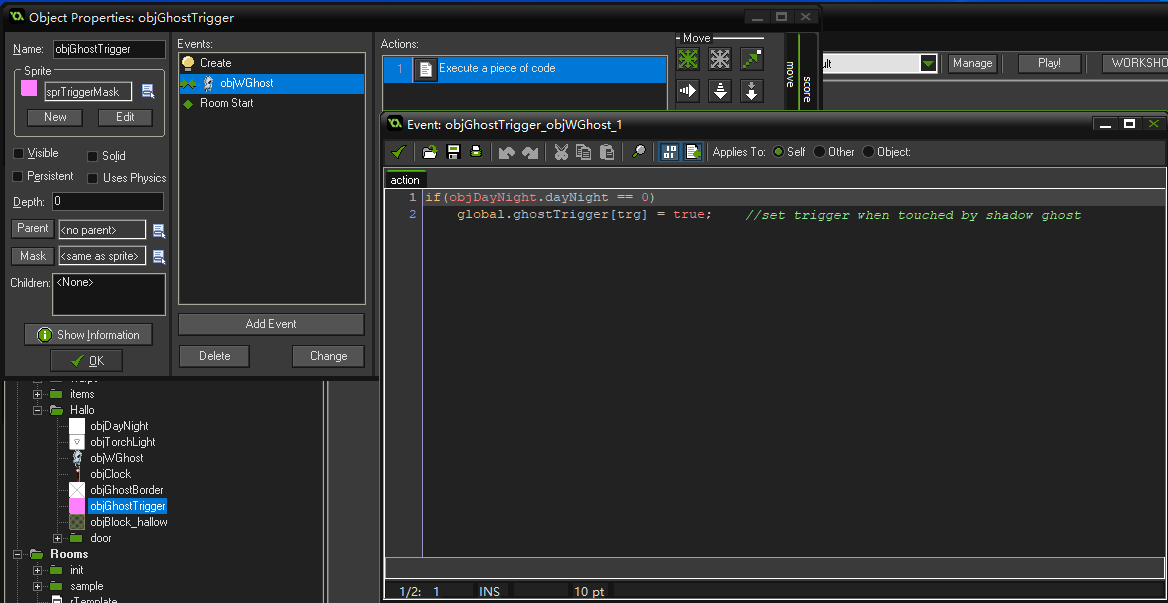
objGhostTrigger

Create



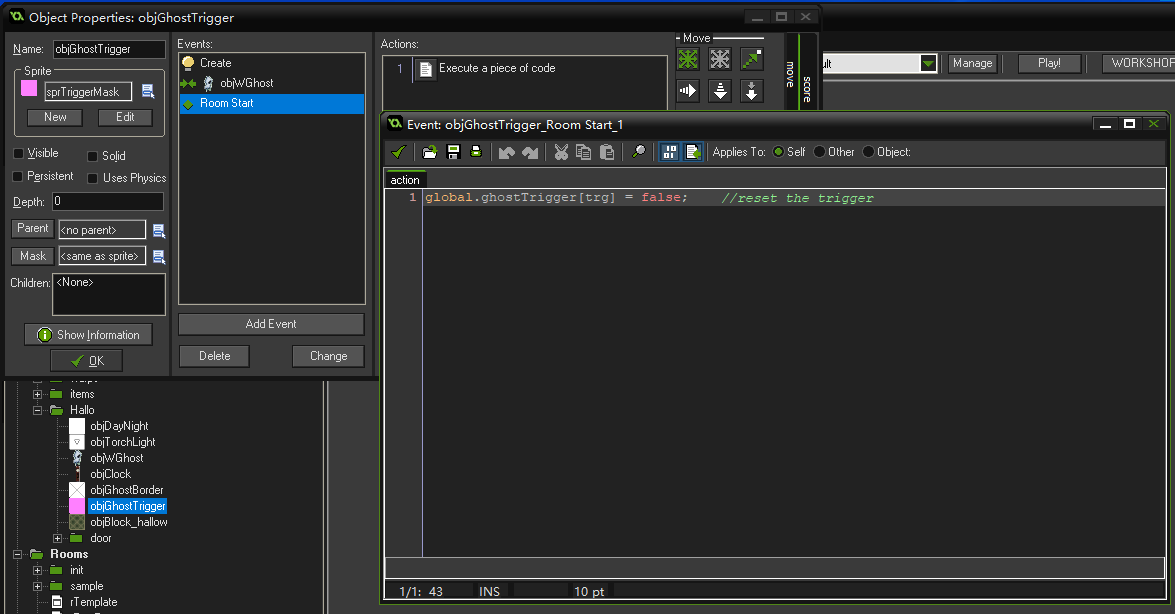
房间内需要新增trigger的时候creation code需要指定trg值

碰objWGhost



只有幽灵碰到它的时候才会触发事件

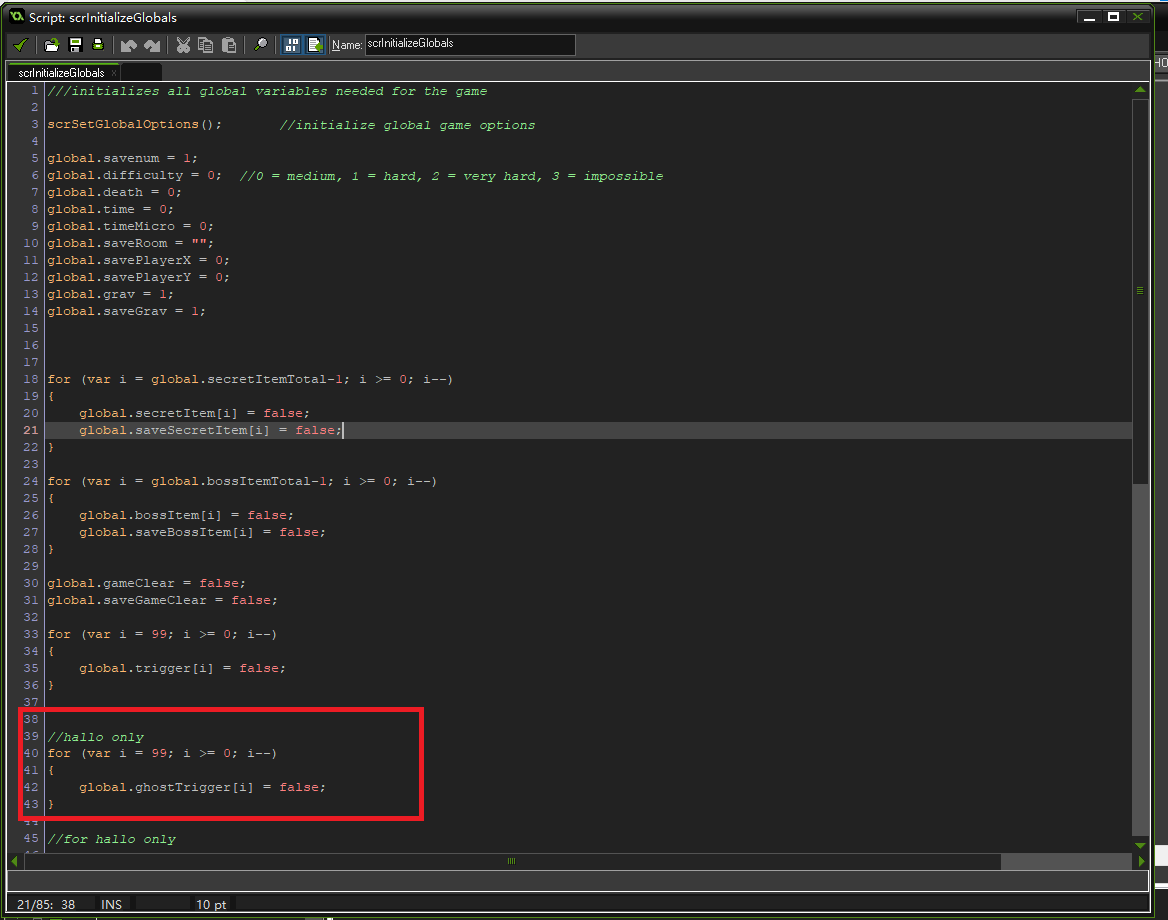
Room Start



初始化对应的global.ghostTrigger值为false，以防止未初始化的bug

备注：有在如下原生脚本中有修改

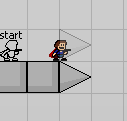
scrInitializeGlobals



objBlock\_hallow

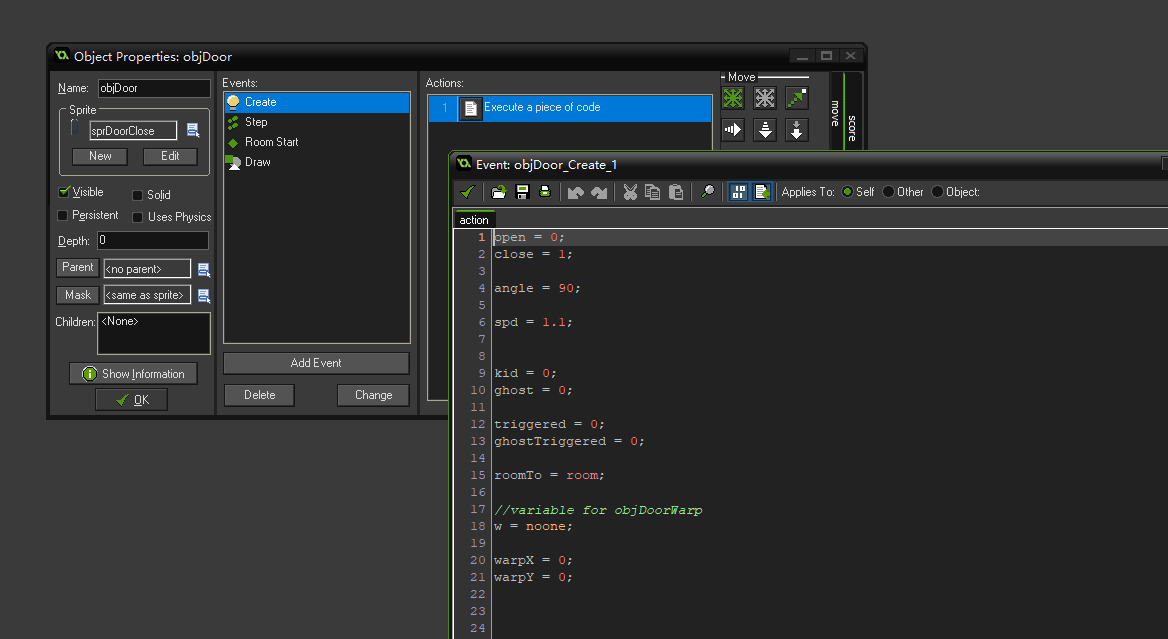
零梦自己搞的砖块，可以当objBlock看

但印象中自己搞的砖会导致

这种情况下kid会死亡

没测试过会不会触发这个问题，建议试试看

objDoor



摆刺时需要写入的变量为

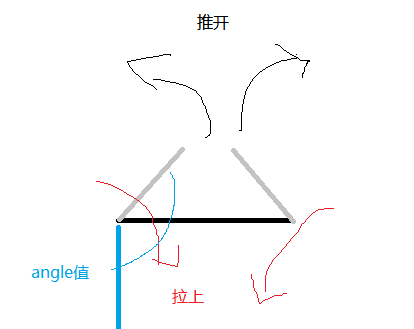
必填：roomTo,warpX,warpY

选填：spd,kid,ghost

open和close对应的是门的开关状态（即当前门是否已打开，当前门是否已关闭）

angle指门扉开启的最大角度

（默认“打开”指的是“往内推开”，“关闭”是“往外拉上”，如下图）



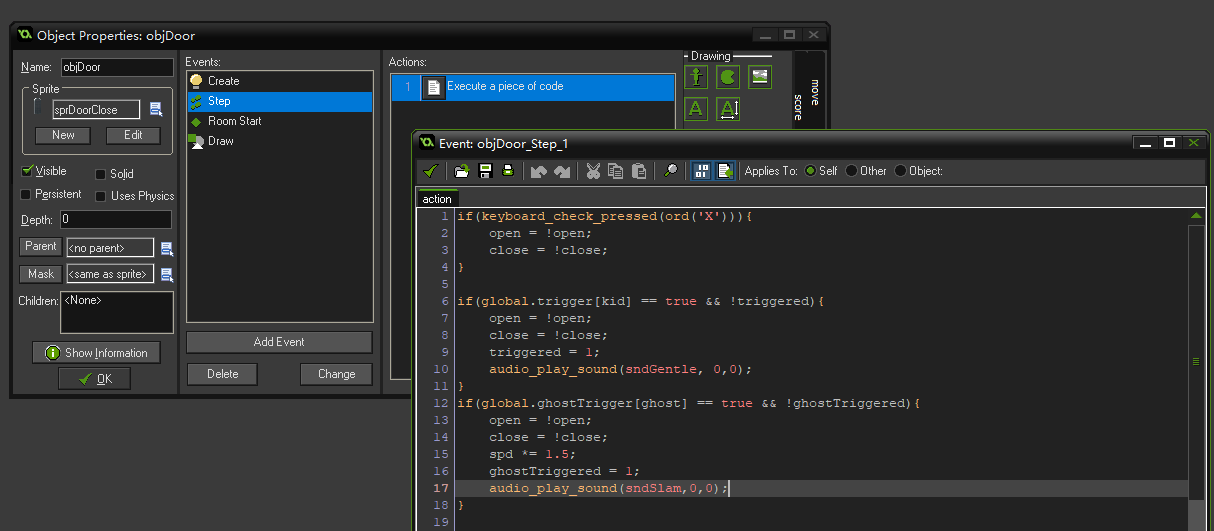
spd指门开关的速度

kid,ghost指代对应的objFreeTrigger和objGhostTrigger的trg号，即门是可以被kid和鬼 触发的

triggered和ghostTriggered分别对应是否已被kid和鬼触发过

当门打开的时候会生成一个objDoorWarp，kid碰到objDoorWarp之后就会播放特效并进入roomTo的房间的(warpX,warpY)坐标（暂不支持把鬼传送走）

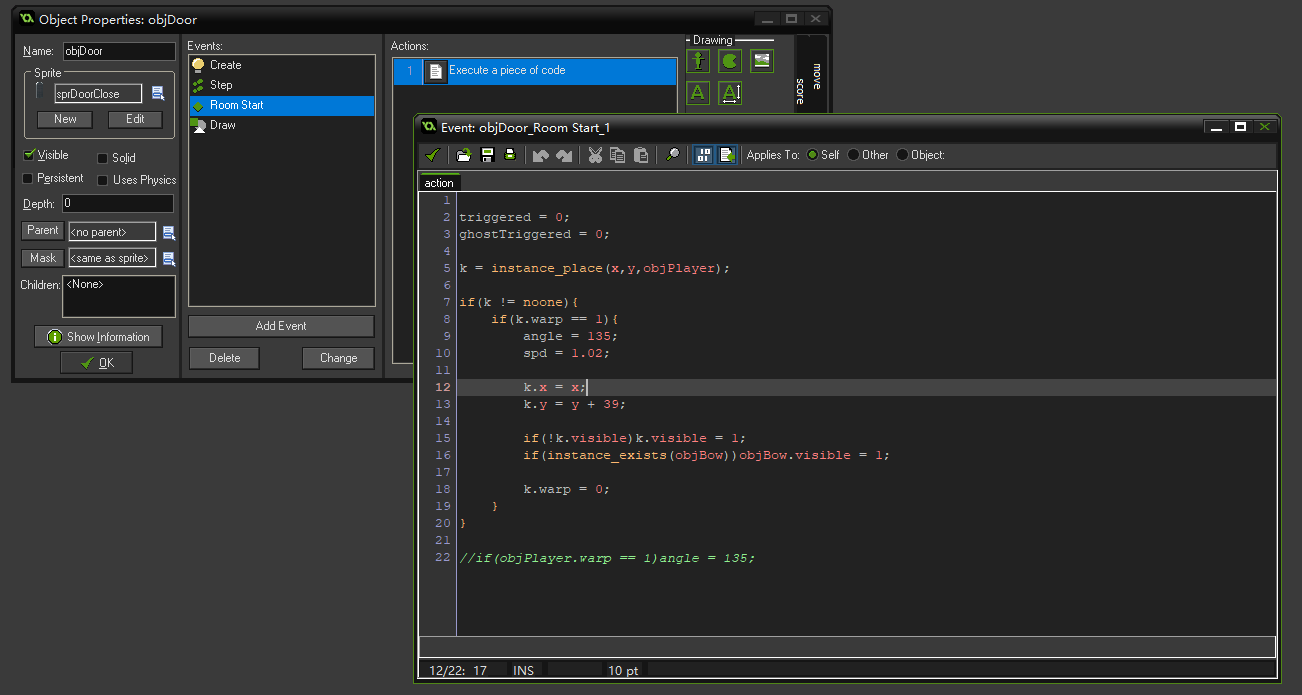
Step



**按X开关门是个debug function，实际使用的时候记得去掉**

分别设置kid和鬼碰到对应开关后门的反应，鬼触发的时候播放不同的音效

Room Start

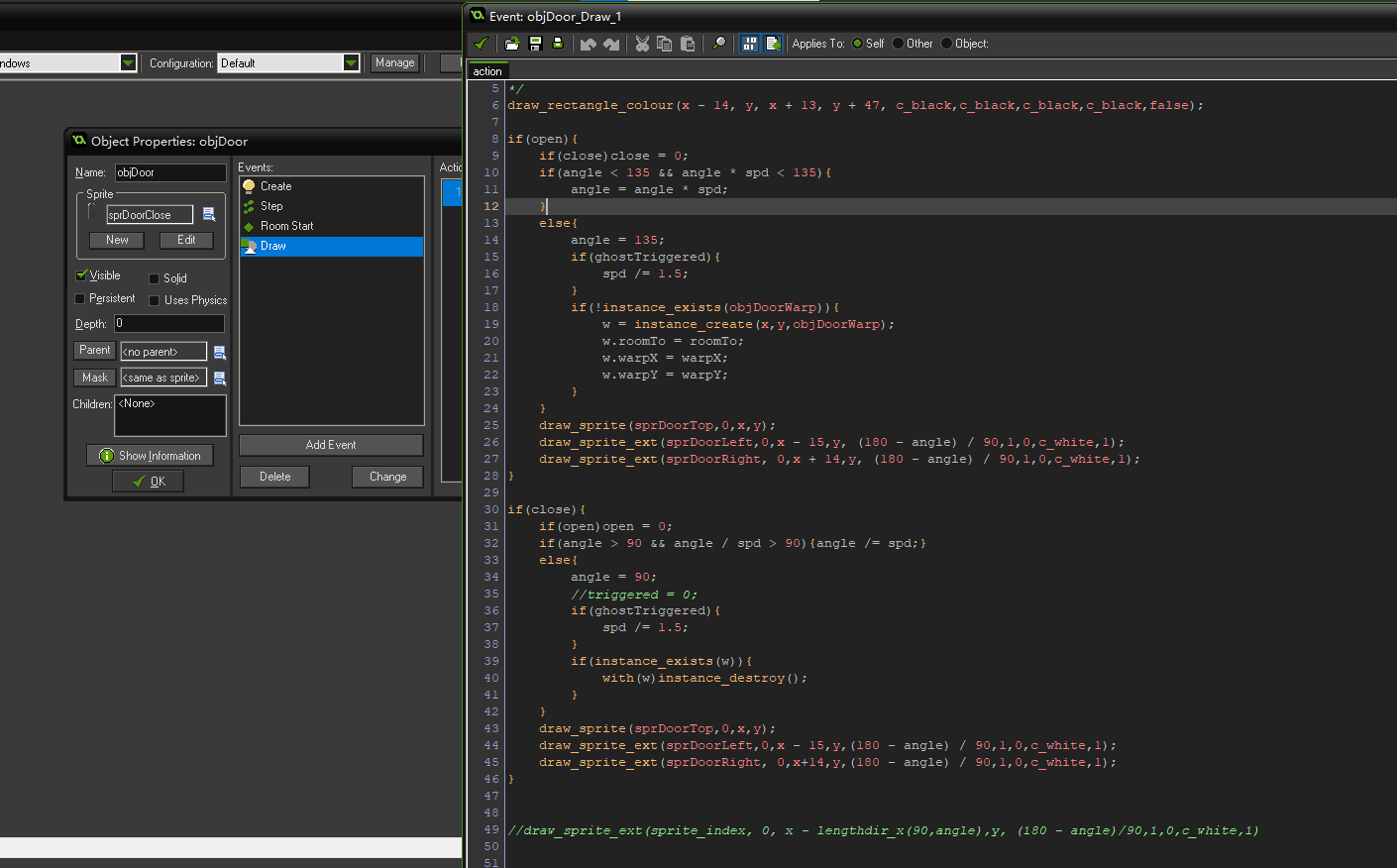


这里的设定是假如kid进门之后，起始点应该也是在一扇门处，造成一种“出门”后“进门”的效果

所以如果kid碰到objDoorWarp被传送到roomTo的（warpX,warpY）坐标后刚好碰到一扇门，则那扇门会自动设定为打开，并固定kid“站在”门缝处

**注意：objDoor的这种开门方式不会产生objDoorWarp，否则会无限传送**

Draw



门的默认贴图是完整的一扇门，以便直观地进行开发

但实际绘制的时候是分为sprDoorTop，sprDoorLeft和sprDoorRight三个部分

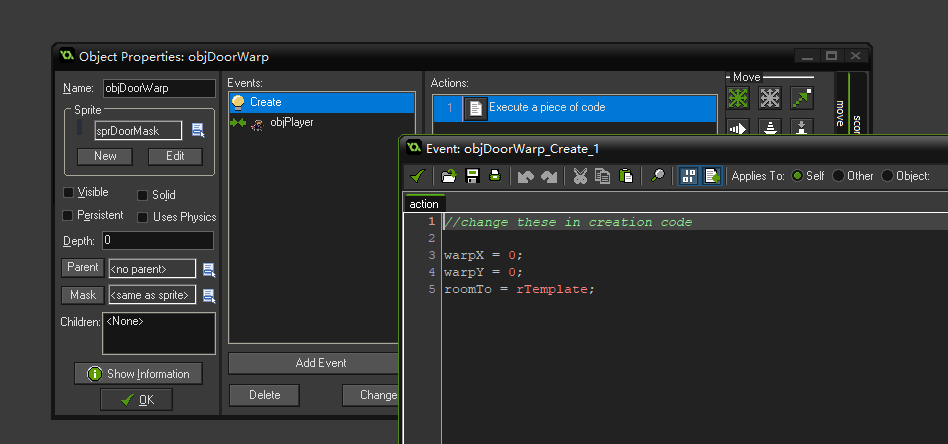
然后一开始绘制的是“门洞”的部分（黑色长方形）

如果门是打开状态的话会增加angle直到最大角度135为止，并据此绘制sprDoorLeft和sprDoorRight以造成“推开门”的效果，如果通过这种方式推开门的话就在门缝处生成objDoorWarp以供kid“进门”

如果门是打开状态的话会减少angle直到最小角度90为止，并据此绘制sprDoorLeft和sprDoorRight以造成“拉上门”的效果

objDoorWarp

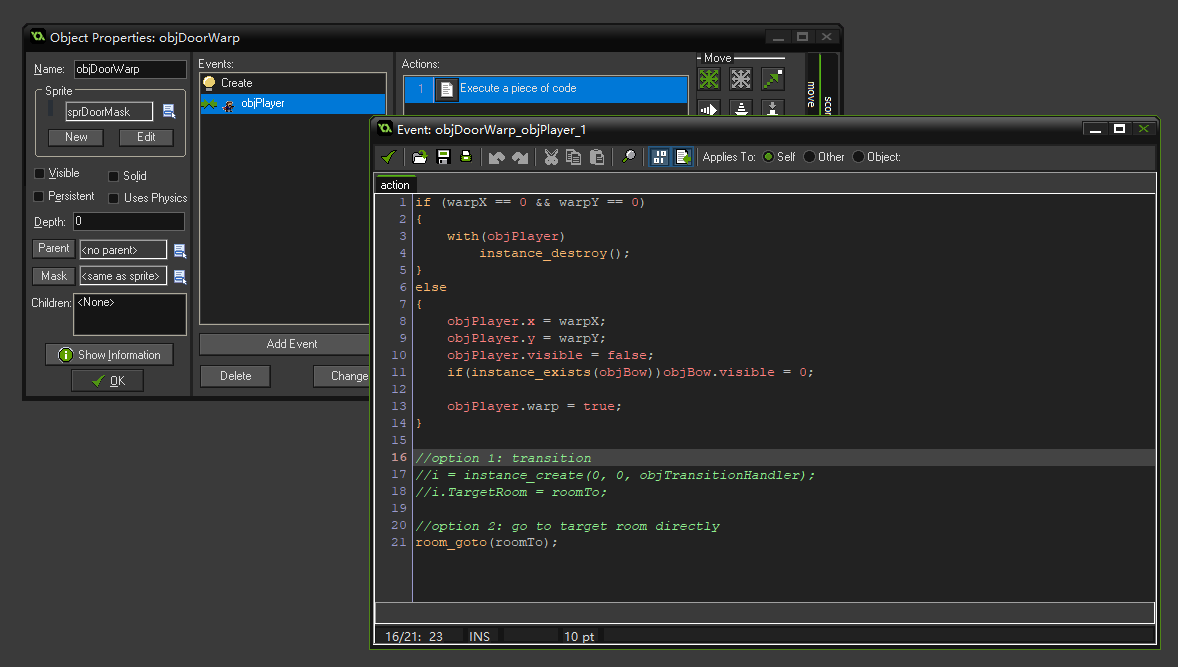
Create



其实就是照搬的果warp代码

碰到这个objDoorWarp之后会传送到roomTo，如果有指定warpX和warpY的话就传送到房间内的（warpX,warpY）坐标

碰objPlayer



如果有指定warpX和warpY的话（**建议指定的对应坐标处最好有扇门！！**）则传送到对应的一扇门处并强制打开该门

否则传送到下一面的playerStart处（**不建议，因为如果是进门的话最好是从门到门**）

可以选择使用注释处的两行代码来增加一个淡入淡出的过渡特效，或者像现在这样直接传送到下一个房间

objTransitionHandler

会由objDoorWarp产生的一个播放特效并传送kid到roomTo的对应坐标的obj，原理略