## 概述

### 相关插件

缓冲时间条：

◆Drill\_GaugeOfBufferTimeBar UI - 缓冲时间条

◆Drill\_CoreOfGaugeMeter 系统 - 参数条核心

缓冲时间数字：

◆Drill\_GaugeOfBufferTimeNum UI - 缓冲时间数字

◆Drill\_CoreOfGaugeNumber 系统 - 参数数字核心

**两个插件的功能是实现地图界面中 实时剩余时间 可视化而存在的，仅提供时间显示。**

### 相似文档

以下为设计方法相似的插件文档，可以结合理解：

|  |  |
| --- | --- |
| 插件名称 | 文档名称 |
| Drill\_EnemySimpleHud  战斗UI - 简单生命框 | 5.战斗UI > 关于战斗简单生命框.docx |
| Drill\_GaugeForBoss  UI – 高级变量固定框 | 13.UI > 关于高级BOSS生命固定框.docx |
| Drill\_GaugeForVariable  UI – 高级变量固定框 | 13.UI > 关于高级变量固定框.docx |
| Drill\_GaugeOfBufferTimeBar  UI - 缓冲时间条  Drill\_GaugeOfBufferTimeNum  UI - 缓冲时间数字 | 13.UI > 关于缓冲时间条与数字.docx |

### 插件关系

插件的关系如下图。



## 缓冲时间条

### 结构

缓冲时间条的结构如下图所示。



**时间条物体：**表示一个放置在地图界面中的独立物体，具有自己的贴图和时间计时器。时间条物体可以绑定到事件/玩家/图块上。

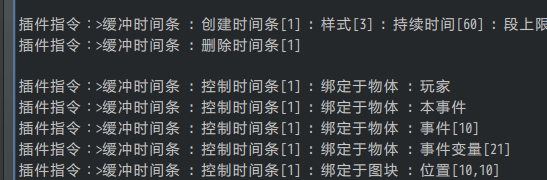
**时间计时器**：计时器是时间条物体内置的结构，不可控制。在绑定物体之后，开始计时。通过不断地给予参数条时间数据，使得参数条能够缓慢增长/缩短。

**时间参数：**参数条中对应的参数为时间，单位帧。（1秒60帧）

参数条的内容这里不赘述，去看看 ”1.系统 > 关于参数条.docx”。

### 时间条物体

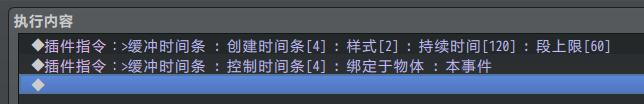
控制时间条物体需要通过插件指令控制。





举个例子，创建一个id为4的时间条物体，该物体绑定于当前事件。

绑定之后，时间计时器开始正式计时。当时间达到120帧(持续时间)后，结束计时，并自动消失。计时期间不受其他条件影响。



注意，如果创建了id相同的时间条物体，那么后一个物体会覆盖掉前一个物体。

另外， 段上限 用于控制时间条的 段 结构，120/60 = 2余0，时间条将会显示2段，即绿色部分和蓝色部分。更多 段 的详细内容去看看 ”1.系统 > 关于参数条.docx”。

### 外框

时间条的结构为 2框+1参数条 结构。



如果你要创建一个类似雕花的外框结构。你需要先画一个黑白遮罩遮住参数条，再给参数条添加一个玻璃外框的结构。

参数条样式： + 

时间条的外框：



### 配置流程

缓冲时间条的配置流程如下图。



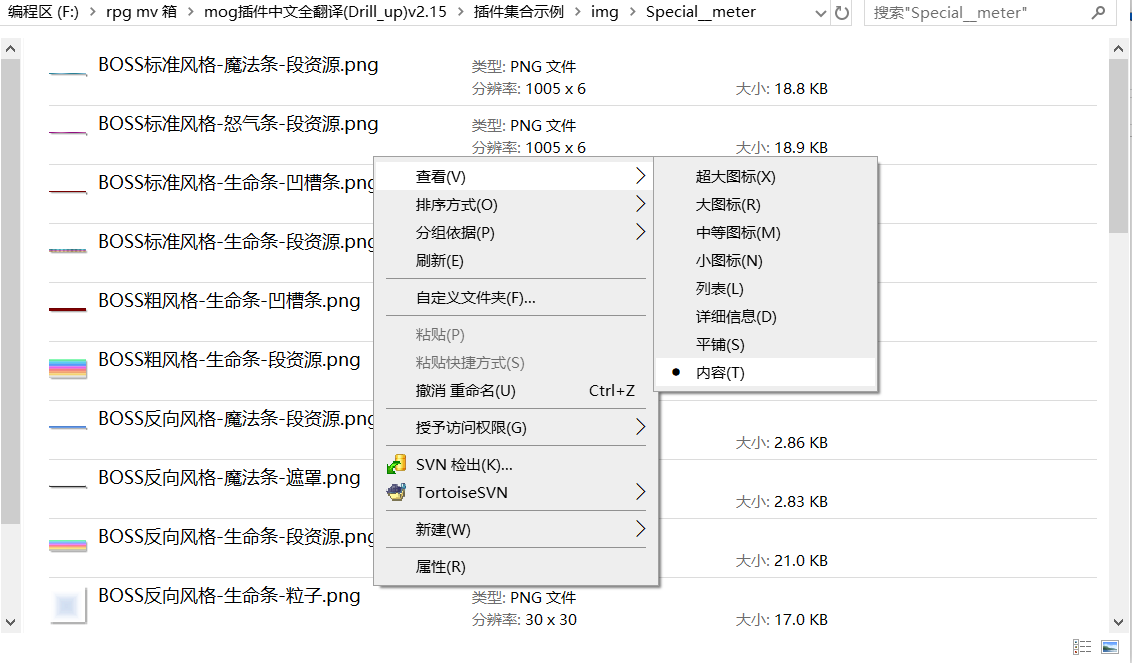
**”时间条[4]”表示 时间条id为4的物体，这个物体可以控制绑定到指定的事件上**。

图中 参数条样式 -> 时间条样式 是一对多的关系，也可以一对一设计。

图中 时间条样式 -> 时间条物体 是一对多的关系，也可以一对一控制。

图中 时间条物体 -> 事件 是多对一的关系，你可以创建多个时间条物体，并且绑定到同一个事件上，但是，时间条物体只能绑定到一个事件上。

**在文件夹中查看时，建议将内容平铺，方便看图片名称和分辨率**。



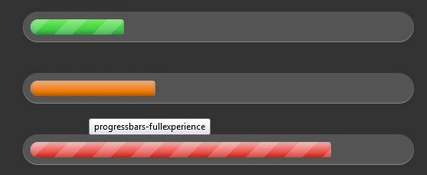
### 从零开始设计（DIY）

在开始设计前，你可以先了解一下 [时间条的配置流程](#时间条的配置流程) 。

#### 1. 设置一个目标

首先要想明白自己想创建一个什么样的时间条，这一点非常重要，示例中提供了很多的参数条，但是其画风不一定适应你的游戏。

比如，我从百度上随便找了一个看起来不错的时间进度条。



#### 2. 结构规划/流程梳理

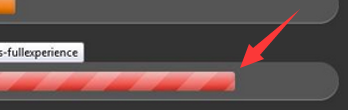
你需要将 参数条与框 的结构分离。因为二者在不同的插件中配置。

初步观察，可以发现，外面灰色的部分，可以作为一个单纯的玻璃背景，或许玻璃背景透明会有更好的效果。

红色的条就是参数条了因为参数条是圆角的矩形，所以需要用遮罩，由于图片是静态的，不确定是否流动。不过，加个流动效果效果会更好，流速稍微慢点。



图中的圆口，插件是无法实现的，因为插件的底层是通过矩形切割形成的，只能为硬边缘。

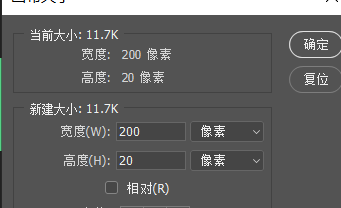


#### 3. 起草资源

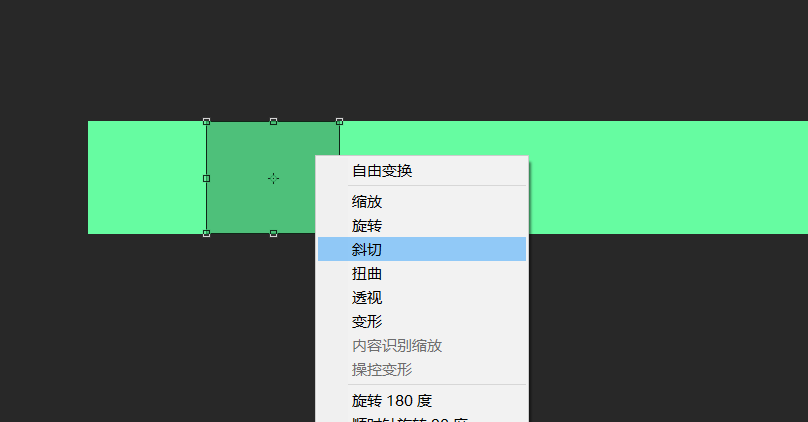
有了大概的思路，接下来就可以开始画画 遮罩图片和参数条图片 资源了。

不过示例中都配置了八层颜色，赤橙黄绿青蓝紫银，画好后多配几层也无妨。

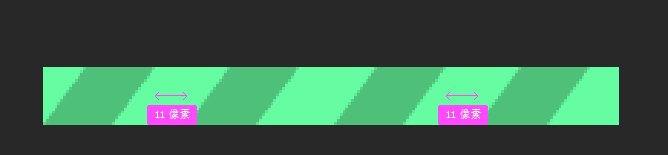
这里我打算开启流动效果，然后设置 段长度划分 ，并设置段长度100。这样，资源的宽度200就够了。



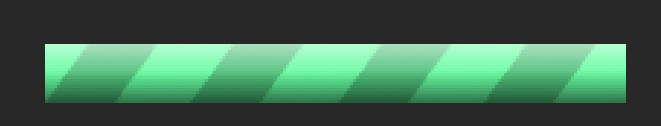
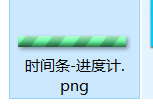
参数条中的平行四边形并不难画，只要画一个长方形，然后斜切就可以了。



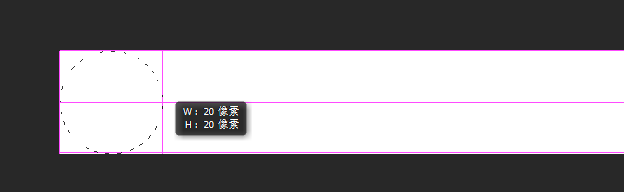
初步的设计如下。



拉个黑白渐变，完成 段 的部分。

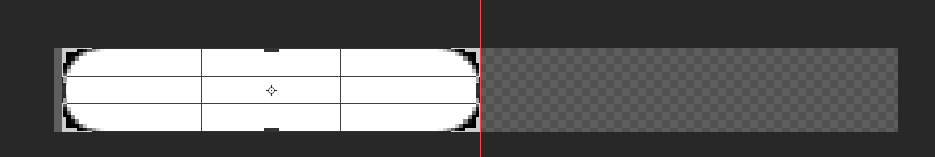
接下来画遮罩，画一个圆，选择反向，涂黑，然后相裁剪就可以了。

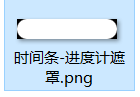




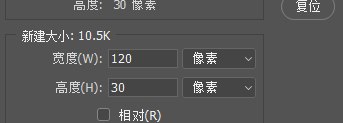


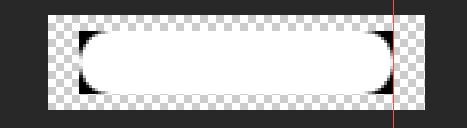
**这里忽然想起，因为我设置了流动，所以资源图片要把200改成100**。

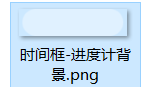




接下来画外框。外框就在遮罩的基础上，稍微加宽一下就好了。

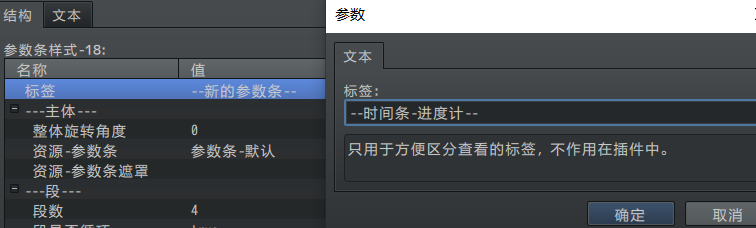




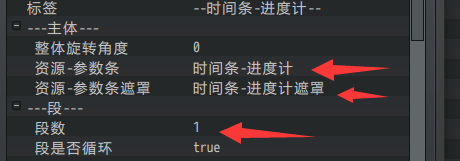


#### 4. 配置样式

首先，配置参数条的样式。标签名没想到有什么好的名字，就简单叫 进度计 吧。



配置最基本的几个参数。其他配置默认都是关闭状态。





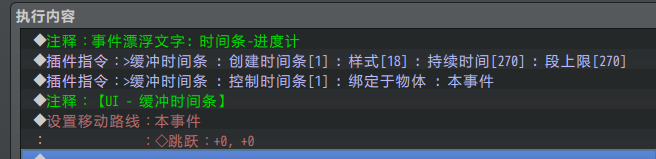
接下来，配置时间条的样式，时间条样式只要对应上id，然后把外框配上就可以了。前景不需要，把前景资源配置成空。





#### 5. 放置一个实例

样式配好后，需要通过插件指令放置一个时间条实例。这里的时间条样式是18。因为目前只画了一段，所以段上限和持续时间都设置为270就行。



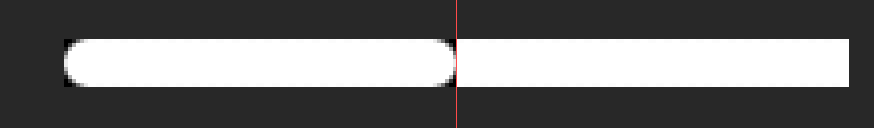
效果出来了，感觉一般般。流动速度慢了，而且背景的框太白了，看起来比较刺眼。还有，太粗了。Σ(°Д°;



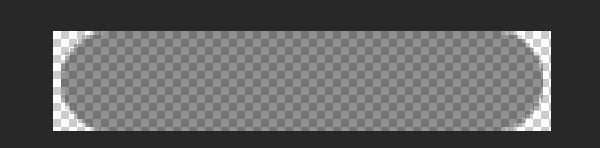
#### 6. 细节调整

将之前的工程的高度调小，重新画遮罩，确保时间条是细长的。

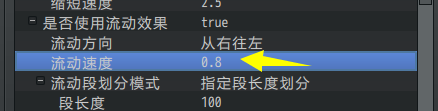




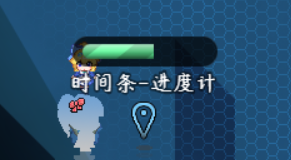
把原来的白背景换成黑背景。



速度设为之前的4倍：



这个比之前的要完美多了。（果然 黑色+细长 = 高级 吗……）

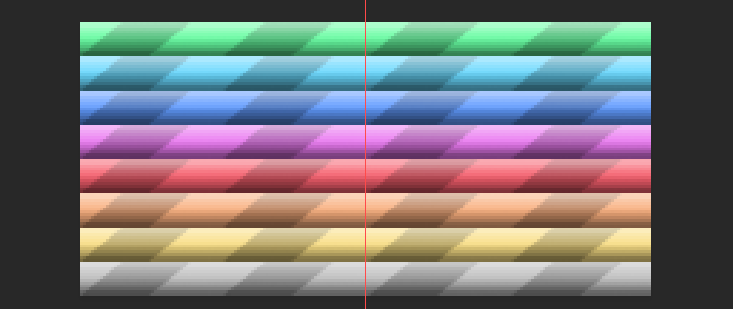


#### 7. 收尾部分

功能已经完成了大半，接下来就是完成 多段 的部分了，如果有其他想法，加粒子效果、游标之类的都可以。

下图的多段，单独看起来会非常突兀，因为上方的白色亮部和下方的黑色暗部撞色了，不过反正在实际的游戏中看不见，所以不用担心美观问题。





（这个时间条画好后，再回去看原始的简易时间条，发现丑死了。W(ﾟДﾟ)w真是没有对比就没有伤害。）



## 缓冲时间数字

### 结构

缓冲时间数字的结构如下图所示。（使用机制与时间条相似，原理稍微有些区别。）



**时间数字物体：**表示一个放置在地图界面中的独立物体，具有自己的贴图和时间计时器。时间数字物体可以绑定到事件/玩家/图块上。

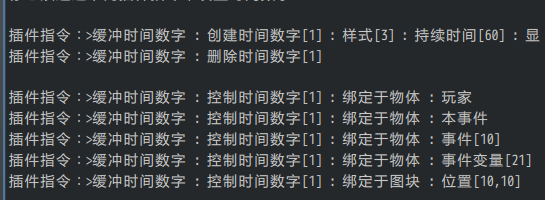
**时间计时器**：计时器是时间数字物体内置的结构，不可控制。在绑定物体之后，开始计时。通过不断地给予参数数字时间数据，使得参数数字能够持续变化不同的数字。

**时间参数：**参数数字中对应的参数为时间，单位帧。（1秒60帧）

参数数字的内容这里不赘述，去看看 ”1.系统 > 关于参数数字.docx”。

### 时间数字物体

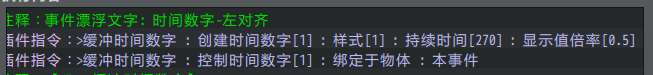
控制时间数字物体需要通过插件指令控制。





举个例子，创建一个id为1的时间数字物体，该物体绑定于当前事件。

绑定之后，时间计时器开始正式计时。当时间达到270帧(持续时间)后，结束计时，并自动消失。计时期间不受其他条件影响。



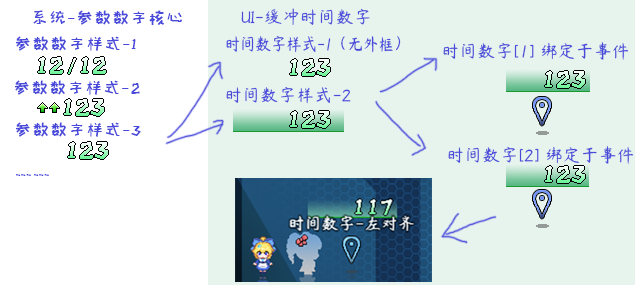
注意，如果创建了id相同的时间数字物体，那么后一个物体会覆盖掉前一个物体。

另外，显示值倍率表示显示的时间数字的比。

实际消耗了270帧，而显示的数字为 270 \* 0.5 = 135。

### 配置流程

缓冲时间数字的配置流程如下图。



**”时间数字[2]”表示 时间数字id为2的物体，这个物体可以控制绑定到指定的事件上。**

参数数字的原理和参数条一样，都可以绑定到同一个事件上。

在文件夹中查看时，建议将内容平铺，方便看图片名称和分辨率。



### 从零开始设计（DIY）

在开始设计前，你可以先了解一下 [时间数字的配置流程](#时间数字的配置流程) 。

#### 1. 设置一个目标

先思考一下你的游戏中，什么样的时间数字比较合理。通常情况下，数字的风格有两个就可以了，其余的都是同一种风格不同的结构、大小、图标配置。

这里我打算设计一个偏像素的极简的表示数字，百度找到了下图的参考：



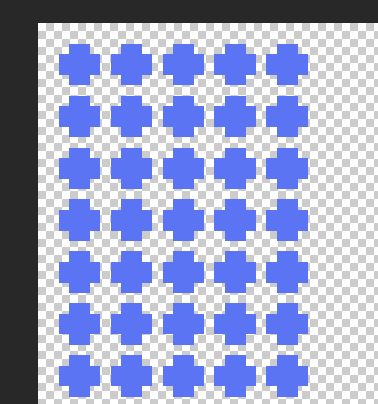
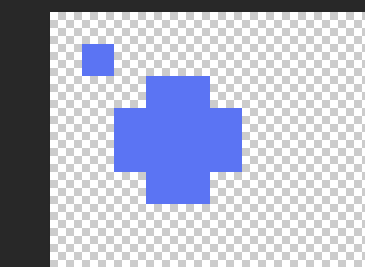
#### 2. 结构规划/流程梳理

你需要将 参数数字与框 的结构分离。因为二者在不同的插件中配置。

初步观察，可以发现，黑色不明显的原点部分，可以作为背景；绿色显眼的部分，可以当做数字。

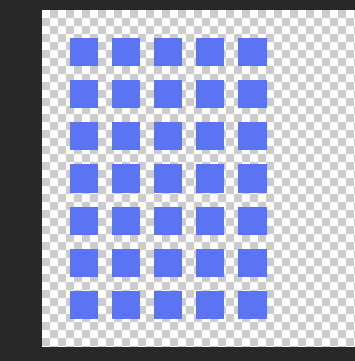
#### 3. 起草资源

先从第一个符号开始吧，一个圆圈就画一点点大。然后平铺开来。



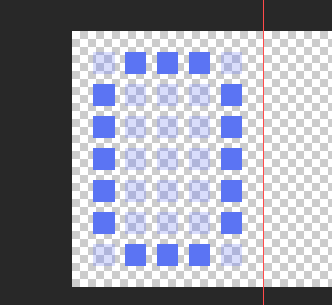
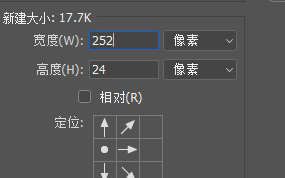
使用4x4可能太大了，这里换小。

四方形的点，一共24像素，刚好符合地图的半个图块。

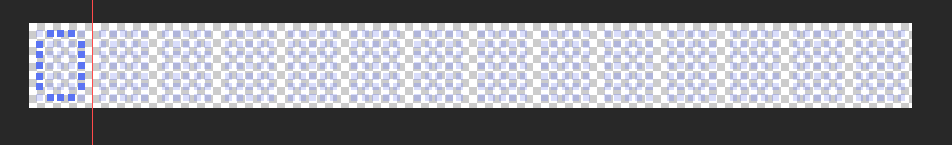


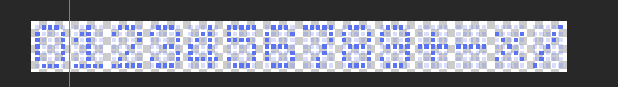
复制图层，然后修改一下透明度，画一个0。

一个字符占用的空间为 18x24像素。18x14 = 252。

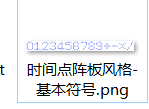
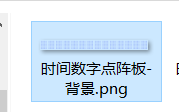
 

设置好后，把像素点平铺。





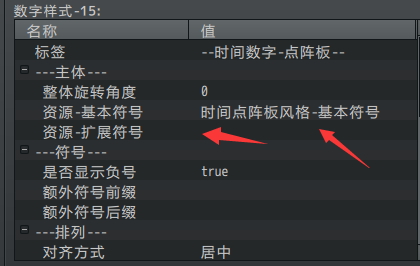
把透明背景暂时去掉，保存文件。另外，保留背景，存一份背景文件。

#### 4. 配置样式

首先，在参数数字核心中，配置参数数字的样式。名字叫点阵板。

由于扩展符号还没开始画，这里设置为空。



需要注意的是，如果对齐方式为居中，那么点阵板背景和点阵数字就不能很好地吻合了，这里设为右对齐。

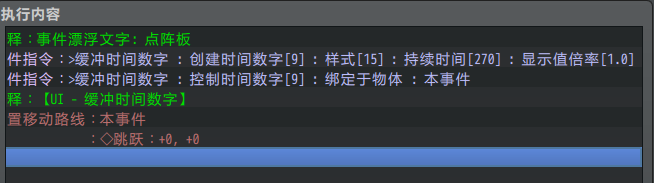


接下来，在 缓冲时间数字 插件里面，配置对应的样式。



#### 5. 放置一个实例

样式配好后，需要通过插件指令放置一个时间数字实例。这里的时间条样式是15。



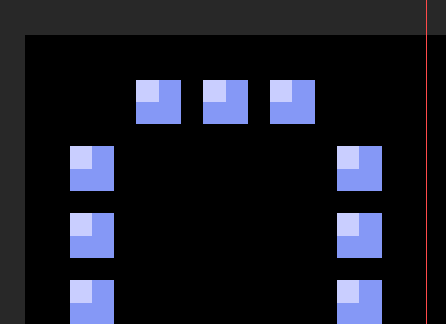
效果出来了，点阵数字是右对齐的，而且完美贴合背景。

不过，亮度似乎不够，而且背景右边的部分可以直接全部去掉。



#### 6. 细节调整

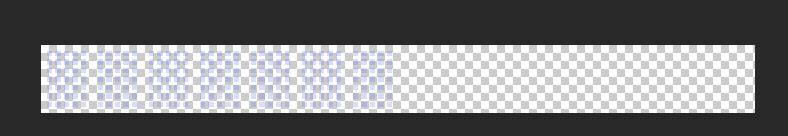
把之前的四个方块全部调亮，然后加上一个光点，使得这个数字看起来更加显眼。



完成之后，发现亮多了。



接下来打开背景，把背景的左边部分的内容全去掉。



效果完成。



#### 7. 收尾部分

功能已经完成了大半，由于背景画的有限，图中只有7位数字可以显示，你可以扩充背景，使其看起来能显示更多位数。

顺带一提，参数数字的数量是有上限的，默认为20个，不过达到20个数字的情况几乎没有。（boss最大的生命位数也只有16位，完全能容纳）



另外，扩展符号还未做，这里可以把扩展符号给补上。



最后，你也可以考虑将参数数字换成时间格式。



但要注意，缓冲时间数字，是以每帧+1或者-1的方式流动的，与时间格式的单位最好对应上。