

大家的 Ren' py (初级)

hello ren'py!

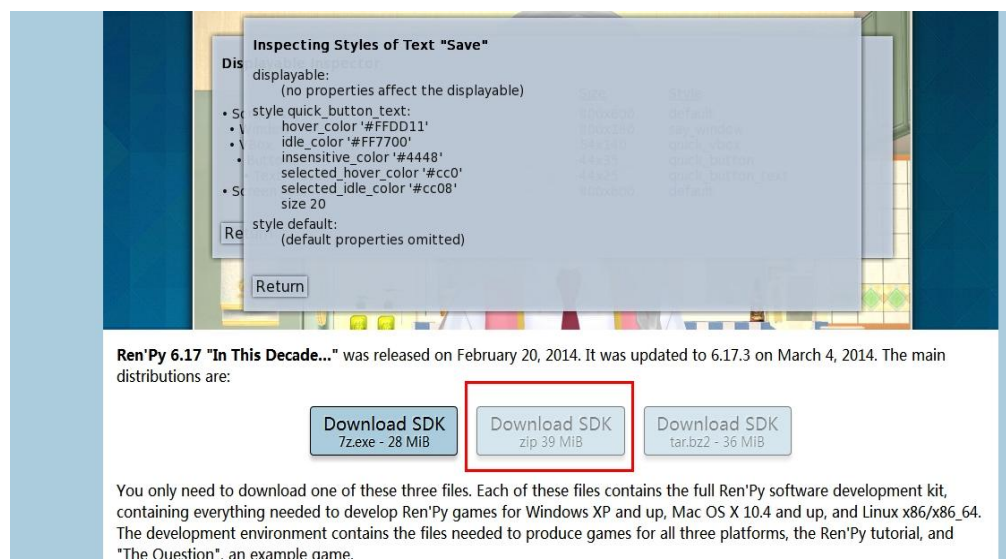
第一章，你好 Ren'py!

1.1 获取 ren'py

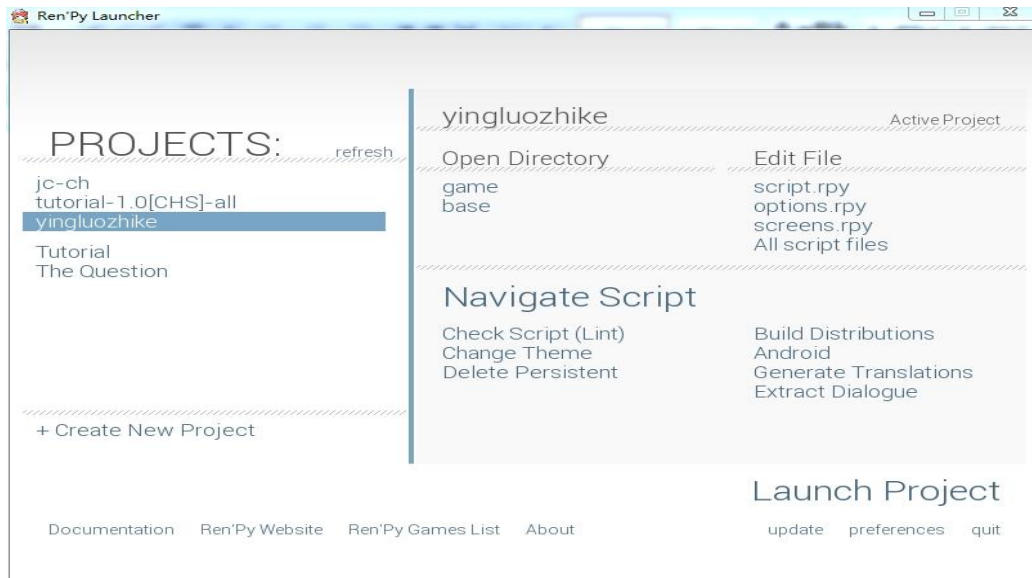
大家可以从官网来下载 ren'py 的客户端，地址为

<http://www.renpy.org/latest.html>

俺下载的是中间红框框的那个。



下载后解压，直接点击 ren'py.exe 就可以运行啦。然后就会是这样的界面：



鸟文的界面是不是看起来很不舒服？不过没关系的，我们的教程不要理那么多，直接使用吧，我打了汉化补丁后遇到了一些小问题，所以换回了鸟文，洋皮华心，逼格就是这么高^_^大家可以根据自己的喜好设置。

ren'py 的界面非常简单，左上角是工程目录，选中自带的示例项目【The Question】，然后点击右下角大大的 Lunch Project 按钮，它就启动了。当然我们这里选由幻玄工作室汉化的版本~

这个 demo 可以给大家展示很多 ren'py 的许多基本功能，如果你能吃透这个 demo 的话，基本也就能做出一款很惊艳的 gal 游戏啦。



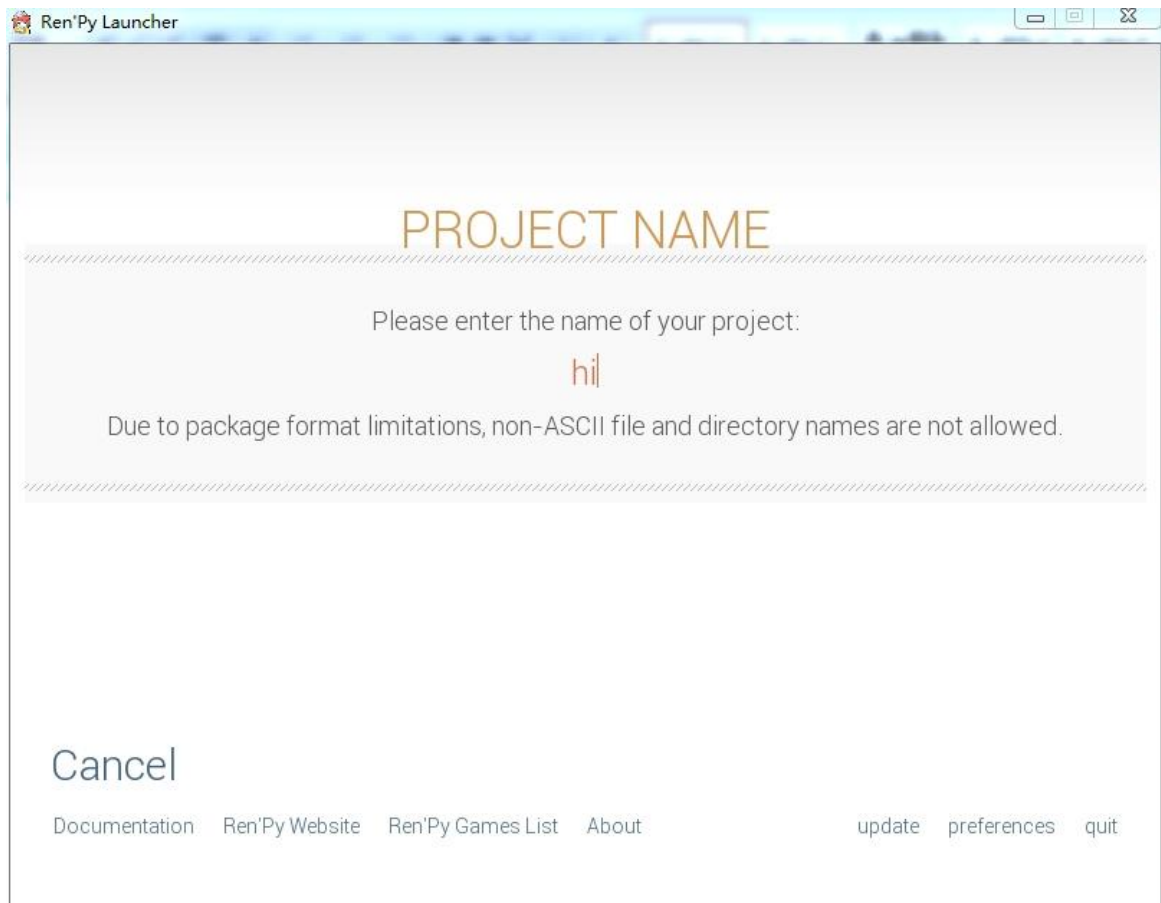
怎么样，由可爱的艾琳酱带领我们游览一番后，是不是觉得很心动？

那么点击 **Create New Project** 开始我们的冒险吧！

1.2 第一次

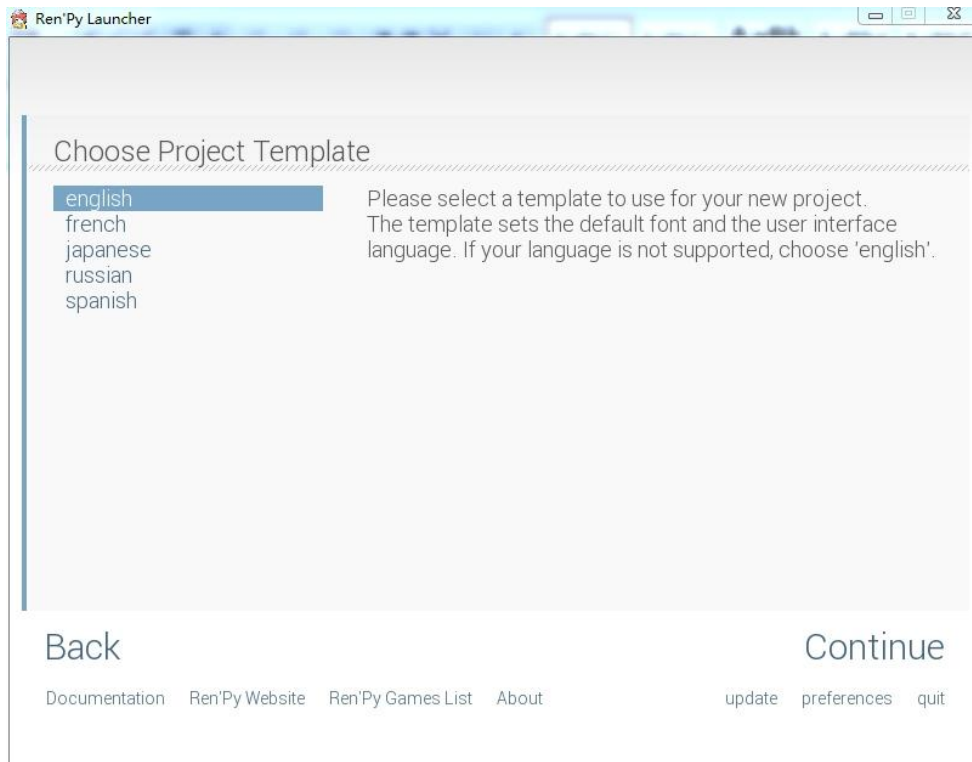
怎么样建立工程呢？大家来看图说话吧！

如下图，首先当然是填写工程名咯，那么我就写上“hi”，然后潇洒的敲击回车。



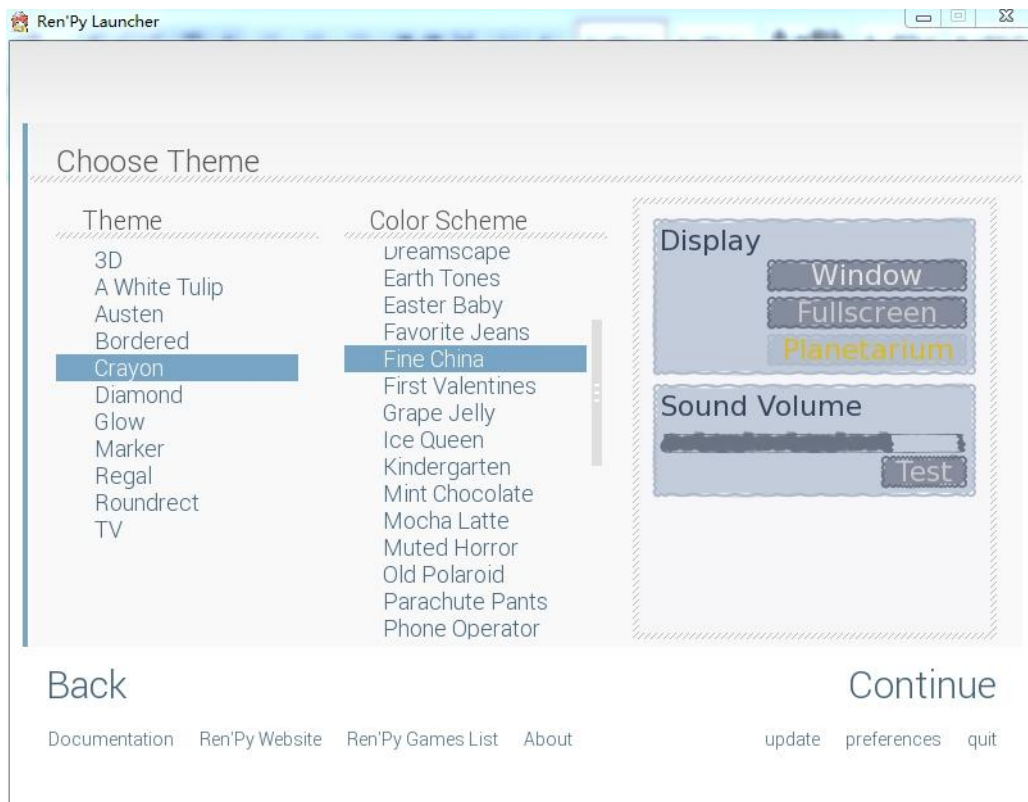
接着就是选择模版。暂时来说还木有中文模板，所以我们看看五种给定的语言后，无奈的选择淫文版吧。

选择好了以后点击 Continue。

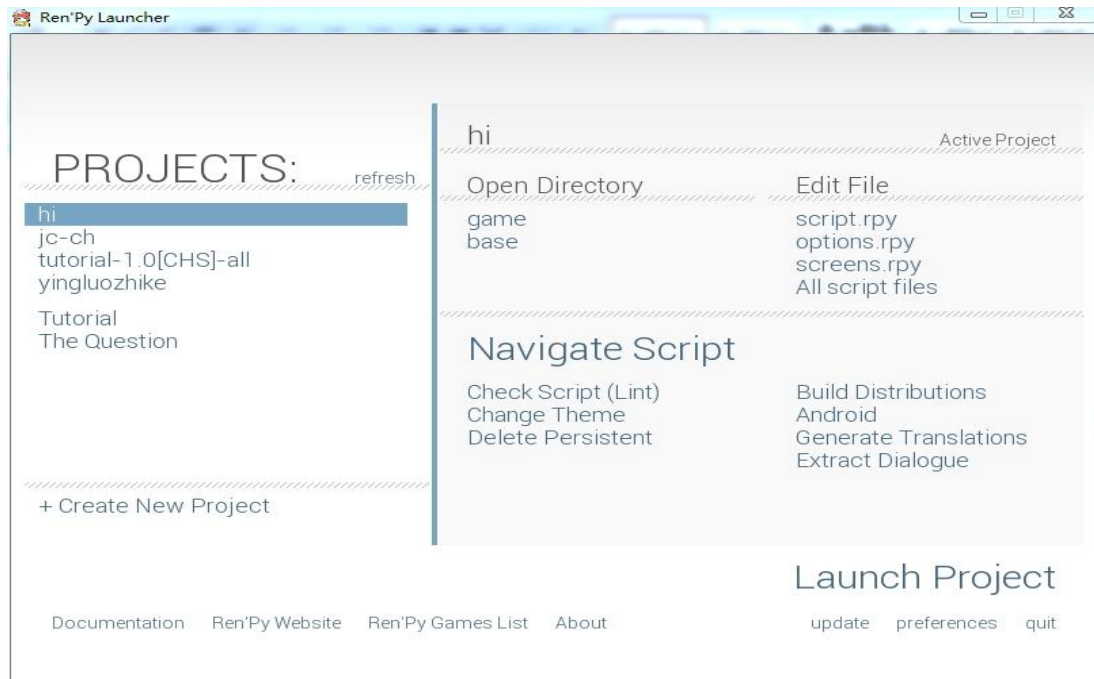


然后是选择主题设计，按个人喜好啦，俺是民族主义者所以选了个带大 China 的。

然后继续 Continue。

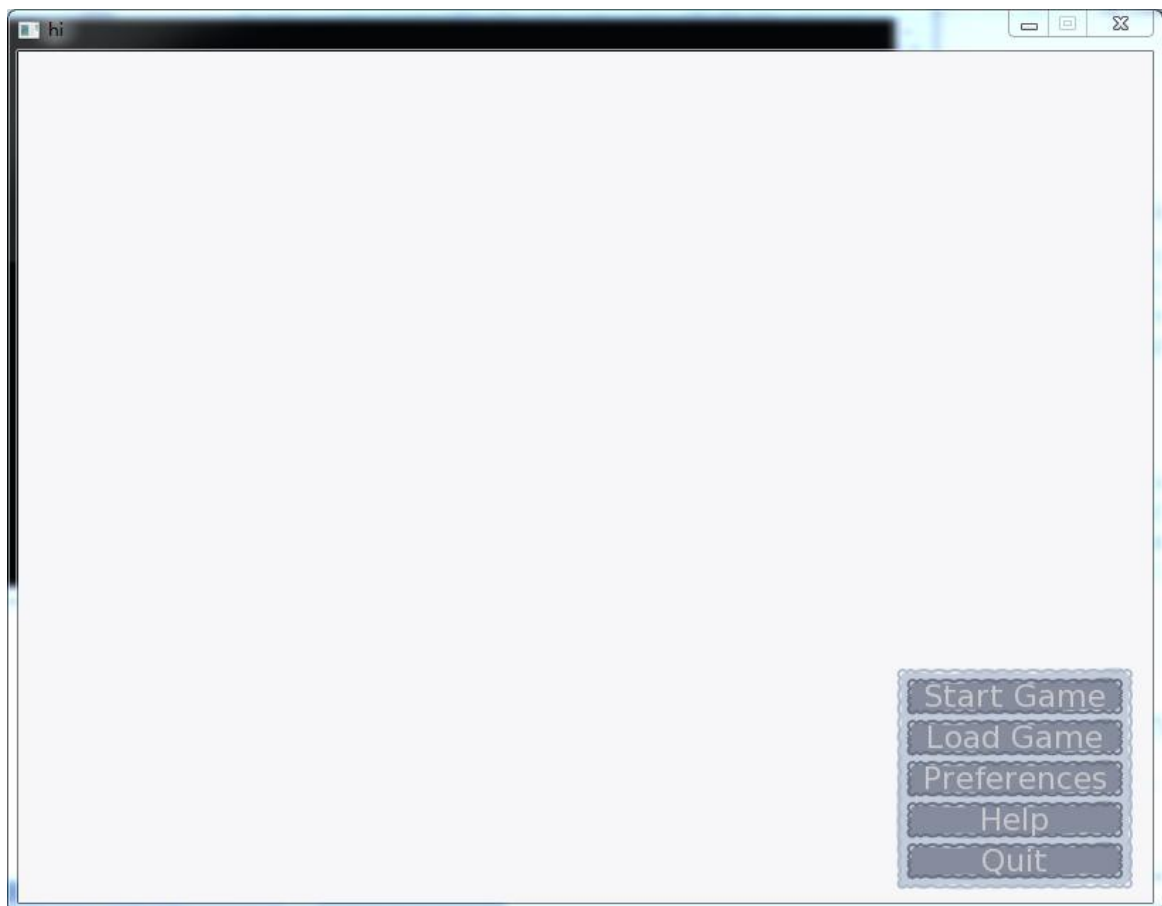


然后就是回到主界面了，同时在工程目录那里有了 hi~ 目测创建成功。那么二话不说就点 Launch Project 吧！



然后...

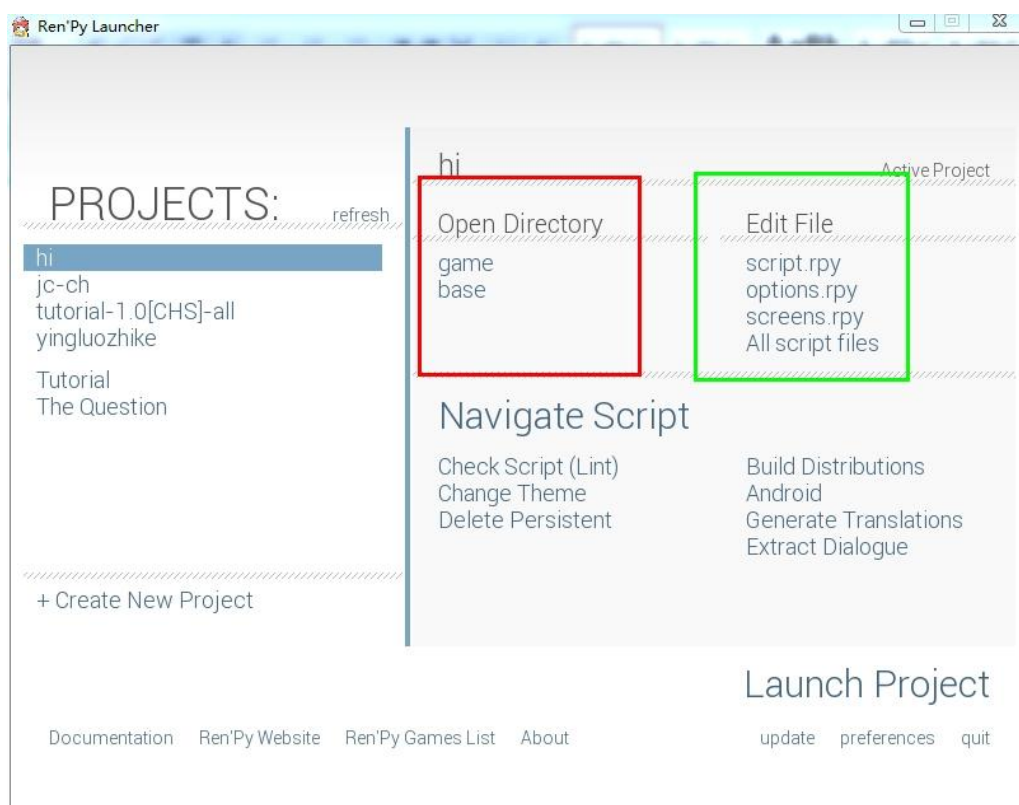
虽然丑了一点，但它仍然是你的第一个 ren'py 程序~
哦，丑陋但是美好的第一次！



1.3 动手改动吧

你在上一步中点击 **Start Game** 了吗？不要告诉我没点，说谎的孩子是要被打 PP 的哦。

那么点了的话你就知道，你的第一次很短，2 句话的功夫就退出了。那么我就动手改动一下，让它成长起来吧。人经历过第一次的话，当然会有所成长的^_^



如上图所示，我们来关注一下红框框和绿框框里都是什么。

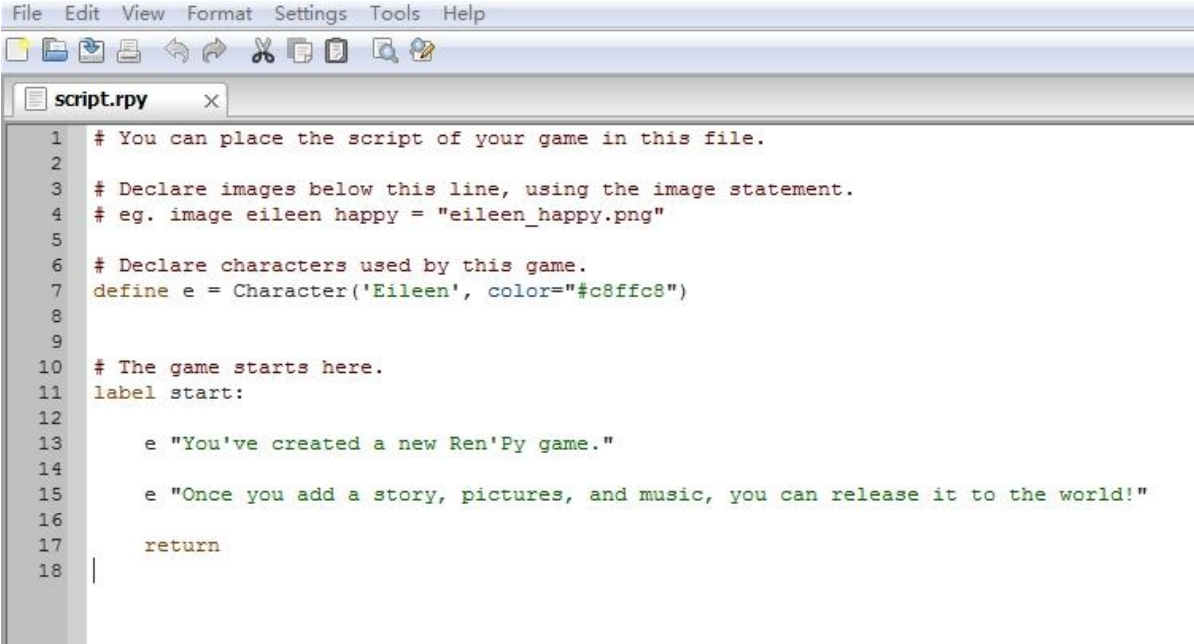
红框框里是你的工程目录，**game** 是所有资源所在，**base** 是工程根目录。也就是说，你的图片，文本，淫乐等等素材都应该放在 **game** 目录下。

绿框框里从上到下分别是你的游戏脚本（控制游戏流程），配置文档（修改游戏配置，详细信息见附录），屏幕属性（对话框什么的）和

所有文件。

如果你除了之前所做的再没有做其他改动的话，那么点击绿框框中的某一个的话，它会让你下载一个文本编辑器。俺选的是 **Editra**。当然俺更喜欢直接进入目录用 **editPlus** 来编辑，看个人喜好了~

在教程的这个阶段，我们就不要管其他的了，来专注一下 **script.rpy** 吧。打开它看一看。



```
File Edit View Format Settings Tools Help
script.rpy x
1 # You can place the script of your game in this file.
2
3 # Declare images below this line, using the image statement.
4 # eg. image eileen happy = "eileen_happy.png"
5
6 # Declare characters used by this game.
7 define e = Character('Eileen', color="#c8ffc8")
8
9
10 # The game starts here.
11 label start:
12
13     e "You've created a new Ren'Py game."
14
15     e "Once you add a story, pictures, and music, you can release it to the world!"
16
17     return
18 |
```

就这样简单的几句话，构成了一个游戏流程，是不是很不错？

那么让我们进入下一章，一起开始 **GAL** 世界的快乐大冒险吧！

第二章， 里世界

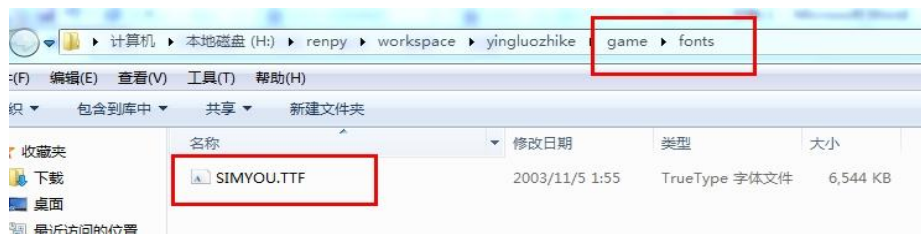
这一章我们就来学习一下如何编写完整的游戏剧情脚本，世界在向着你招手。

“hello world!”

2.1 请说中文

当我们迫不及待的改动了 `script.rpy` 中的鸟语后，我们发现了一个重要的问题，那就是所有中文都是这样的□□□。怎么解决呢？

首先我们先找一个字体文件，打开控制面板，选择一个中文字体文件，我们就选幼圆好了。复制它，然后在 `game` 下新建一个 `fonts` 文件夹，把字体文件丢进去，如下图：



当然你也可以直接丢到 `game` 目录下，不过为了美观和管理方便，我们还是应该将资源分类摆放。

小贴士：你应该按照良好的习惯将资源分类存放。你应该按照良好的习惯将资源分类存放。你应该按照良好的习惯将资源分类存放。因为很重要所以说了三遍。

按笔者建议，我觉得大家按照 `krkr` 的目录方式比较好，这样日后与其他开发人员交流、合作时也会比较方便。

然后在 `script.rpy` 的最上面加上这样一句话

```
init:
```

```
$ style.default.font = "fonts/SIMYOU.TTF"
```

这样我们就将默认的文本字体设置为幼圆了，然后再次启动游戏，你会发现自己的文字正确显示了。

这个世界，终于可以和我们的正常沟通啦！

那么还有同学想问，如果是我想在不同的地方使用不同的字体应该怎么做呢？比如我想让这个世界居民的姓名为仿宋字体，讲话内容为幼

圆该怎么办呢？

请往下看^_^

2.2 每个人都有不同的颜色

在我们的 GAL 世界中，有很多很多的角色，他们每个人都是不同的，所以我们应该赋予他们每个人姓名不同的颜色，让他们与众不同。

我们设定这个世界中有三个人，男主角小绿，女一号小红，女二号小黄，那么我们就在最开始的地方定义它们：

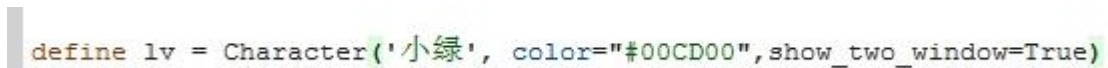
A screenshot of a text editor window titled 'script.rpy'. The window contains 12 lines of code. Lines 1-3 are comments in Chinese. Line 4 is a comment about the font. Line 5 is the 'init:' keyword. Line 6 is an indented line setting the default font to 'SIMYOU.TTF'. Lines 7-8 are comments about character colors. Lines 9-11 are 'define' statements for three characters: 'lv' (green), 'hong' (red), and 'zi' (purple), each with a specific color code. Line 12 is a blank line.

```
1
2 # 这是我们创造的世界，不要放弃哦！
3
4 # 普通话是温润的幼圆
5 init:
6     $ style.default.font = "fonts/SIMYOU.TTF"
7
8 # 这个世界的居民有着属于每个人自己的色彩.
9 define lv = Character('小绿', color="#00CD00")
10 define hong = Character('小红', color="#FF0000")
11 define zi = Character('小紫', color="#BA55D3")
12
```

这样每个人都有了属于自己的色彩。

其实除了色彩，我们还可以赋予我们的子民更多东西。查看文档我们看到还能指定这些常用的属性：

1, show_two_window:把他的名字放在对话框上方

A screenshot of a text editor window showing a single line of code. The line defines a character 'lv' with the color '#00CD00' and the attribute 'show_two_window=True'.

```
define lv = Character('小绿', color="#00CD00", show_two_window=True)
```

2, show_side_image:给这个臭屁的家伙在对话框旁边加上头像，xalign 和 yalign 指的是头像放置的位置，它是 0 到 1 之间的数字，代表在屏幕位置的比例，比如 0.5 就是指在屏幕的中间。

```
define lv = Character('小绿', color="#00CD00",  
    show_side_image=Image("actor/iren/eileen_side.png",xalign=0.0,yalign=1.0))
```

3, `what_underline`: 为文字加上下划线。

```
define lv = Character('小绿', color="#00CD00",what_underline=True)
```

4, `window_left_margin`: 文本框距离屏幕左边的位置, 值只能为数字。

```
define lv = Character('小绿', color="#00CD00",window_left_margin=200)  
...
```

5, `window_yminimum`: 文本框的最小高度, 如果文本高度大于设定值, 会变高, 值只能为数字。

```
define lv = Character('小绿', color="#00CD00",window_yminimum=300)
```

6, `what_slow_cps`: 逐字显示, 值只能是数字

```
define me = Character('今夜不谈风月', color="#436EEE", what_slow_cps=20)
```

还有更多的属性, 亲爱的小伙伴们可以自己探索, 也可以给我留言提问。当然如果我在以后的使用中如果觉得还有什么比较常用的话也会加在这里的。

总之, 请赋予自己的角色更多的不同吧。

2.3 说话的方式有很多

我们的 **GAL** 世界既然已经有了居民, 那们赶紧的让他们活跃起来吧。

那么怎么样让他们说话呢?

每一行文本就是文本框中的文字, 他们用英文双引号包裹。

如果是小绿说的话，我们可以这样：

```
The game starts here.  
abel start:  
    "小绿" "我是小绿，我很环保，是个ntr!"
```

这里红框框中是一个空格哦

前面的双引号包裹姓名，后面的双引号包裹他的话。

当然因为这样做实在太麻烦了，所以我们就可以用之前定义好的角色，直接这样：

```
define lv = Character('小绿', color="#00CI  
define hong = Character('小红', color="#FI  
define zi = Character('小紫', color="#BA5:  
  
# The game starts here.  
label start:  
    lv "我是小绿，我很环保，是个ntr!"
```

如果是只有一对引号的话，那就不会有姓名，只会显示文本，可以用来表达心理。

我们知道，世界中每个人的性格都是不同的，他们说话的口气，心情也会不同。可是怎么样来表达呢？那么我们就使用类似 **html** 标签的东西来让他们说话的内容丰富起来~

一般有这些常用的标签。

1, **a** 超链接，它可以指向一个网址

```
me "欢迎访问我的{a=http://weibo.com/u/2014054663/home}微博{/a} 来提出建议^_^"
```

2, **b** 加粗，它可以把标签内的文本加粗

```
"加{b}粗{/b}了~"
```

3, color, 它可以改变标签内的文本颜色

```
"{color=#ff0000}汹涌的红色{/color}"
```

4, cps, 文本显示速度

```
"{cps=20}文本显示速度是20|{/cps}"
```

5, font 文本字体

```
"我还是幼圆--- {font=fonts/simfang.ttf}变成仿宋啦!{/font}."
```

6, i 斜体

```
"我是好孩子, 站的正直。{i}我是坏孩子, 走路都走不直。{/i}"
```

7, image 文本中添加图片

```
"心情不错, 发张自拍吧喵~{image=actor/iren/eileen_side.png}"
```

8, s 删除线

```
"微博上的公知是 {s}傻B{/s}."
```

9, size 字体大小, 值只能是数字

```
"{size=26}|字体大小是24, 默认单位为px{/size}"
```

10, u 下划线

```
"{u}下划线{/u}"
```

常用的大概这些吧? 其他更多的效果让我们在进阶教程里面再学习

吧, 路要一步一步走滴。

2.4 为世界赋予美丽的风景吧

直到现在为之，我们的世界还是只有文字木有图像。那么这一节就让我们动手来为这个世界赋予诗一般美丽的画面吧！

首先要做的第一步，是整理好我们手中的图片素材。有些小伙伴有一个误区，认为图片的大小长宽无所谓，认为在程序中强制设定其长宽就好，包括以前的俺。但其实这是错误的，当你在动手构建你的世界的时候，应该把所有图片都调整为合适的宽度和高度。

那么得到一张图片后，我们就把它加进我们的世界中吧！

为了方便管理，我们在 `game` 文件夹下新建一个 `bg` 的目录，里面专门存放背景图片。

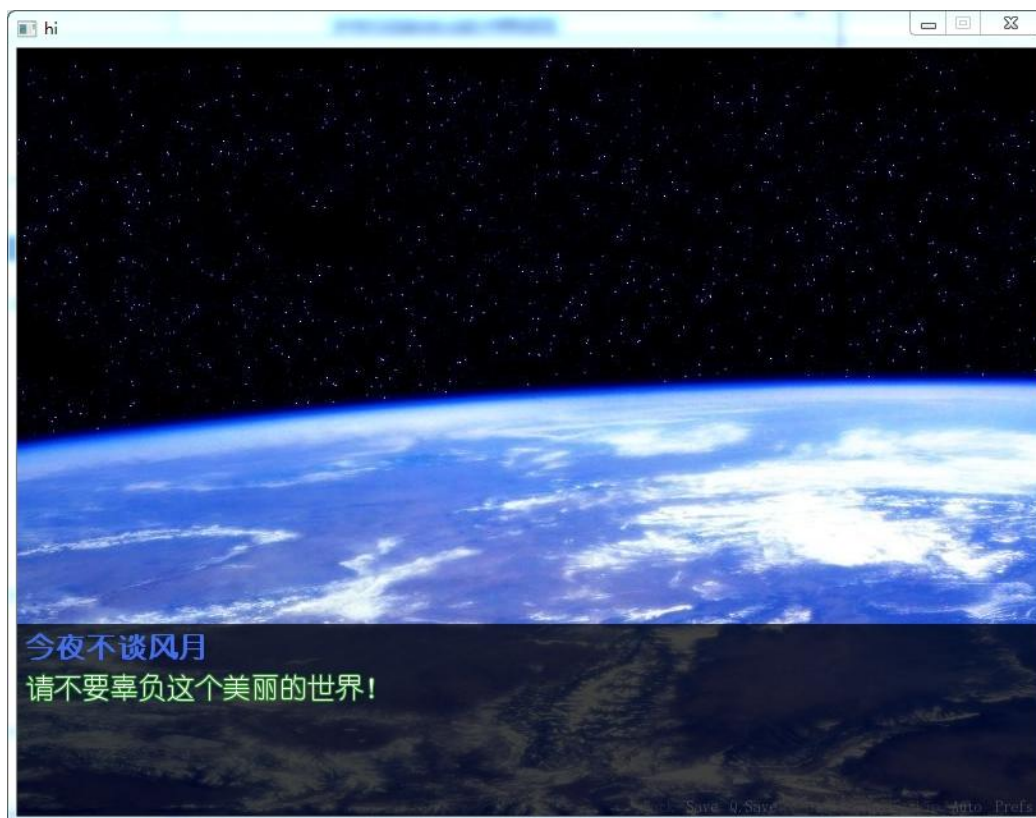
接着，我们就在代码中定义它。同样为了方便管理，我们新建一个叫做 `bg_init.rpy` 的文件，里面这样写：



然后想在合适的地方使用它的话，这样做：

```
scene bg space  
  
me "请不要辜负这个美丽的世界！"
```

然后就可以看到效果啦！至于文字的那些效果的话，在后面我会说怎么做的，在初级教程中，大家应该关心怎么样实现最基本的功能才是第一位的^_^



怎么样，定义了这样一张美丽的图片，是不是很心动呢？可是你的世界应该是很丰富的，里面有花草树木，高山流水，那就需要很多很多张图片。

难道你要一行行、一张张的这样定义吗？

答案当然是否定的。笔者在进入大学学习编程的第一课时，我们的老师（一个很老的程序员）就说过，优秀的程序员都很懒，他们会尽量让机器做更多的事。

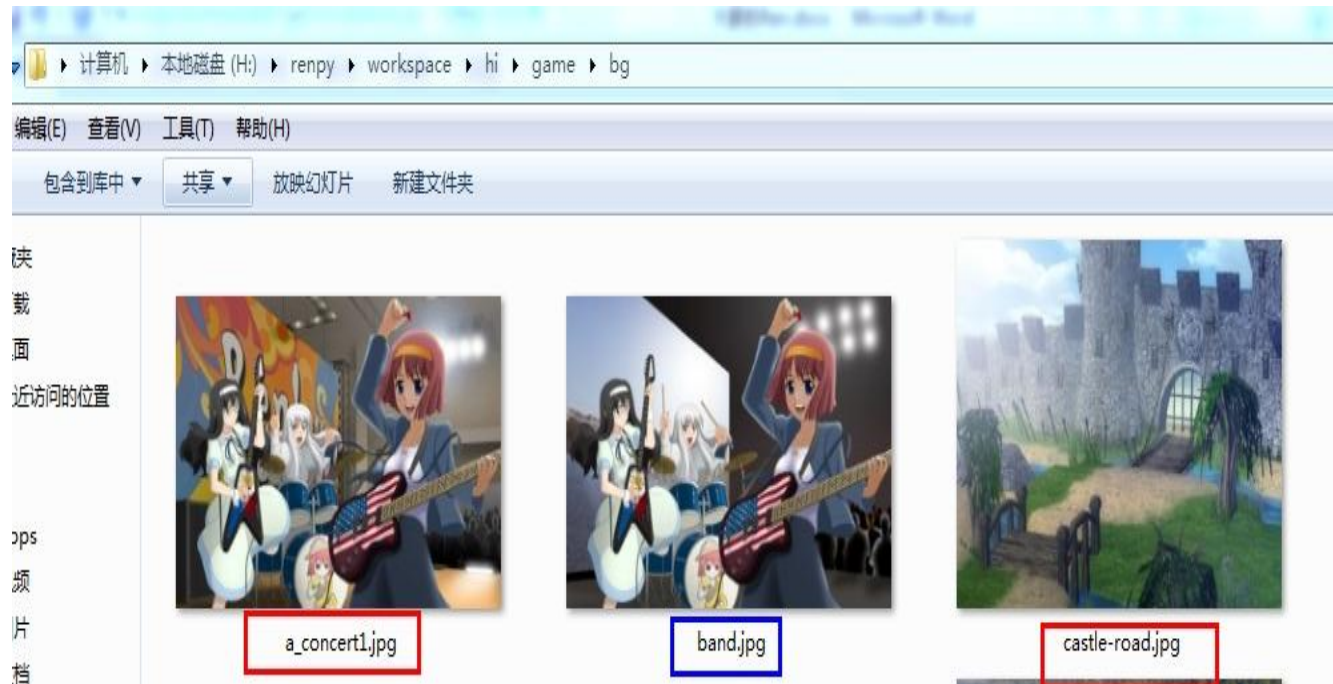
那么，就让神奇的 `ren'py` 来帮我们自动定义图片吧！

首先第一步，你需要规范你的图片，它有两个要点：

①命名必须是两段，即由减号（-） 或者下环线（ ）或者其他什么的隔开

②放在 `game` 或者 `game` 的子目录下

大家看下面这样的：



这里有三张图片，红色框框里的就是可以自动加载的图片，蓝色框框里的就不可以，因为它的名称只有一段。

第二步，我们打开 **options.rpy**。之前已经说过，它里面设置我们游戏的各种配置，比如窗口大小什么的。而里面的内容，其实可以写在任意的地方，同时我们也可以随意改动。

因为我们现在要做的自动加载图片算是基本配置，所以我们就写在这里好了。拖拉到 **options.rpy** 的最下面，加上这样两句话：

```
#####  
## More customizations can go here.  
  
#自动加载图片  
#为自己的图片命名的时候必须有加载的这些符号  
config.automatic_images = ["/","_","-"]  
  
#为了在显示时只需要方便的指定文件名而不用加上文件夹的名称，  
#把所有图片用到的文件夹名称加进去吧  
config.automatic_images_strip = ["bg","actor","iren"]
```

抛开注释，上面那句话的意思是如果图片名称中有那三个符号并且大

于等于 2 段，那么就会自动加载。（这样的理解在严格意义上说是错的，但无关大局，我们就这样理解并使用好了）

名称中是不能有斜杠（/）的，我们加进去是因为我们将图片并不是直接放在 **game** 目录下，而是 **game** 的子目录下。

第二句话不是必须的，加不加它有什么区别呢？

如果不加它，在上图两个红色方框框中，我们的图片会被自动命名为 **bg a concert1** 和 **bg castle road**，发现了没，就是会在图片名称前面加上文件夹的名称。而为了使用的时候直接使用图片名称而不去关心它在哪个目录，我们就使用第二句话把所有图片的目录都给切除掉，在中括号[]中按照格式把自己的图片目录都加进去吧！

然后图片就会自动声明为 **a concert1** 和 **castle road** 了~

PS: 有一点非常的重要，那就是文件夹名称一定不能包含在图片名称的小段中！如果包含了的话，要判断好图片被切后的名称哦。

最后，我们就可以直接在游戏中使用图片而不需要自己动手定义啦！

```
scene castle road  
show eileen happy  
me "如果你不理解也没关系，会使用就可以啦！"
```

效果如下：



怎么样，我们的世界是不是变得多姿多彩起来了呢。现在就让我们在这个世界中走一走，看一看，还有什么美丽的景色吧。

那就需要用到切换场景的标签(label) `scene` 了，最简单的用法是它会改变背景图，就像这样：

```
scene castle road  
me "现在是城堡外面，我们去其他地方看看吧！"  
scene cave 1  
me "到山东了，老板来个大饼！"
```

游戏中点一下鼠标，场景就嗖的切换了~



点击 ing...



大家要注意，使用 `scene` 的时候，它会把你的人物或其他的东西也给清除。

那么我们就可以使用 `show` 来切换啦。`show` 和 `scene` 的最基本区别是，`show` 不会擦除当前图片，只会覆盖，显示人物的话就使用它，这个我们下节讲。

可是我们觉得，直接这样切换会不会很突兀呢？（注意我前面那个“嗖”的拟态词了吗？）好吧，让我们来为场景切换加上简单的转场效果吧。需要注意的是，如果使用 `scene` 来转场，`fade` 和 `dissolve` 可能没什么，但是其他效果，如新场景从侧方滑入，应该使用 `scene` 特有的效果。区分好它和 `show` 之间的效果哦。

配合转场效果的话，只需要加入一个 `with` 关键字 加 效果名称就可以了，如下所示：

```
scene castle road
with fade

me "现在是城堡外面，我们去其他地方看看吧！"

scene cave 1
with slideleft #新场景从右方以平缓的方式滑入

me "到山东了，老板来个大饼！"
```

简单但非常实用也够用的效果主要有以下这些（更多基本效果请参见附录）：

简单转场效果名称	实现的效果
1, fade	淡入淡出
2, dissolve	溶解

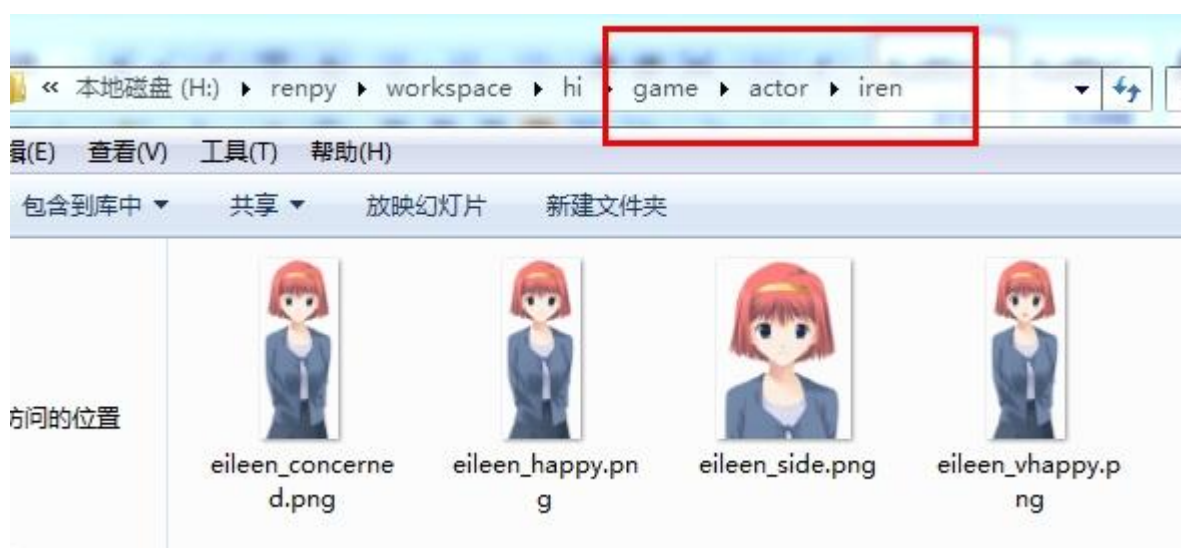
3, pixellate	马赛克式
4, vpunch	上下震动（额...不算转场效果吧）
5, hpunch	水平震动（同上。。。）
6, blinds	百叶窗
7, squares	小方格
8, wipeleft	向左擦除
9, wiperight	向右擦除
10, wipeup	向上擦除
11, wipedown	向下擦除
12, slideleft	新场景向左滑入
13, slideright	新场景向右滑入
14, slideup	新场景向上滑入
15, slidedown	新场景向下滑入
16, slideawayleft	旧场景往左划出同时展示新场景
17, slideawayright	旧场景往右划出同时展示新场景
18, slideawayup	旧场景往上划出同时展示新场景
19, slideawaydown	旧场景往下划出同时展示新场景
20, irisin	向里绽放
21, irisout	向外绽放

更高级的转场效果请观看俺的进阶教程~

2.5 居民们出现了

现在我们的世界有了美丽的风景，可是空无一人，很是荒凉。那就让我们玩一回女娲造人，让居民们出生在这里吧！

第一步，当然就是把居民们的照片放在合适的位置咯。为了以后方便维护，我们在 **game** 目录下新建一个 **actor** 文件夹，然后在 **actor** 下为不同的角色们分别创建一个文件夹，比如我这里有一个角色叫艾琳，那就建个叫 **iren** 的目录，再把所有 **iren** 的照片丢进去。（为了能够自动加载图片，别忘记改上一节教过的配置文件哦）



接着我们就可以直接在脚本中使用 **show** 让角色出现了，就像下面这样：

```
scene castle road
with fade
show eileen happy
me "大家好，这不是我们第一次见面也请多多关照~"
```

可是我们人物有很多个啊，一次让他们出现两个人或者是三个人。可以这样做：

```
scene castle road
show eileen happy at left
show meili happy at right

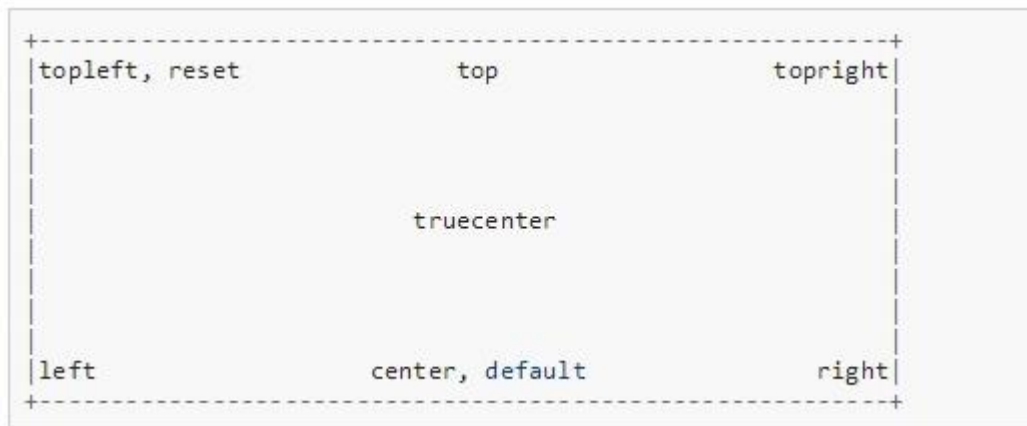
me "我们有两个人哦~"
```

效果如下~



请不要吐槽素材的大小…那是俺随便找的。

同学们发现了没,如果在 **show** 图片名称 后面跟上【**at 位置**】的话,就可以简单的改变人物出现的位置啦。一般来说,简单的定位使用 **left**、**right**、**center** (仅水平居中) 和 **truecenter** (左右上下都居中) 和下图的示例就够了:



同样的，和场景转换一样，人物、前景的出现和改变也可以使用很炫的转场效果哦。使用方法也是在后面（必须在 **at** 后）加上 **with** 转场效果，如下图：

```
scene castle road
show eileen happy at left
with fade#淡入淡出
show meili happy at right
with easeinleft#从左边滑入，如果没有红框框里的语句，则两人都是从左边划入
#注意，场景和艾琳都是以淡入淡出的方式出现的哦
```

"她们两人分别用不同的酷炫效果登台了！"

使用的时候要仔细注意 **with** 的位置哦。人物常用同时也能用的转场效果因为和场景的转场效果类似，大家一个个来试试吧：

fade, dissolve, pixellate, move, moveinleft, moveinright, moveintop, moveinbottom , moveoutleft , moveoutright , moveouttop , moveoutbottom, ease, easeinleft, easeinright, easeintop, easeinbottom, easeoutleft, easeoutright, easeouttop, easeoutbottom, zoomin, zoomout, zoominout。

这些难道不够用吗？如果想要更多有个性的转场效果，请观看进阶教程。

那么 **show** 指令是用来显示一张图片的，**hide** 指令当然就是用来隐藏一张图片的咯。

需要注意的是 **hide** 只需要指定图片的第一段名称就可以了~

```
scene castle road
show eileen happy at left
show eileen vhappy at right
```

"如果图片名称第一段相同的话，只会显示后面的那一张。"
"所以这里不需要手动hide的。"

```
hide eileen
hide eileen happy#这一句其实不需要
#hide后只需要跟上图片的第一段名称即可|
```

多使用几次领会以下吧^_^

2.6 声音是世界的语言

在 Ren'py 中，播放声音默认有三个频道，**music**，**sound** 和 **voice**。他们互不干涉，即你播你的我放我的。使用的时候我们也应该注意，该放音乐的就用 **music**，音效就用 **sound**，配音就用 **voice**，不要混淆了。

那么接下来，就让我们一步步为这个世界添加绚丽的声音吧！