

摇篮文的书写规则

在游戏的 StreamingAssets 文件夹下有一个 localization 文件夹存放了游戏中所有文本的各语言译文。其中名称为 “_”（单个下划线）的子文件夹是日语，里面有一个名为 __tx_whole.txt 的文本文件（其他文件夹下的同内容文件无效）记录了诺艾儿习得新魔法时屏幕上闪过的所有横排摇篮文，（截至 0.26c）如下：

```
/* -正在初始化 [魔法名称? ]- */
-Initializing [&1]-
/* 正在为ID是.....的PCD注册 */
Registering for:
    Personal casting device
Device ID:
    H4NIW4-TON3
/* -个人信息- */
-Personal Info-
/* 姓名、性别、种族、年龄 */
Name: Noel Cornehl
Race: Female/Elf
Tosi: 105
/* 所属、年级 */
Belongs: Bermit Public Univ.
Class: Cadet 3
/* -注意- */
-Notice-
/* 因为败给了一只魔族，她失去了.....她的第二个设备是..... */
She have Lost the PCD[ST3LL4-M4GC]
due to defeat to a monster.
Her 2nd device: [H4NIW4-TON3]
/* -解码中- */
-Decoding Data-
/* -正在检查合格性- */
-Checking conformity-
/* 魔力容量、精神清洁度、同步性、回路脆性、身体等级*/
Mana Capacity: &1 [CLEAR]
Mental Clearance: 304% [CLEAR]
Synchronicity: 405% [CLEAR]
Circuit Fragility: 201% [CLEAR]
Physical Grades: A [CLEAR]
/* -认证进程- */
```

```

-Authentication process-
/* -正在检查一致性- */
-Checking consistency-
/* 开发ID、魔力消耗、咏唱时长、效率、公平性? */
Dev ID: &1
Mana Consume: &2
Cast Time: &3
Efficiency: &4
Fairness: &5
/* -插槽信息- */
-Socket Info-
/* 四个哈希值? */
1. SNTHNTFN
2. U41AK010
3. KA5GHALT
4. 1RTI4NL3
/* 关于为什么这次的演出这么夸张, 有点在乎这件事 */
nande konnnani oogesana
ensyutu nanoka
kininaru tte hanasi
/* 这个世界的魔法是兽人们管理生产的 */
kono sekaino mahou ha
zyuuzin tachi ga kanri
seisan site imasu
/* 愿市民能够安心使用已经确立了安全性的魔法 */
anzen sei ga kakuritsu
sareta mahou wo simin ga
ansin site tsukaeru you ni
/* 出于安全考虑每个魔法认证设备只能使用一次 */
mahou media ha security no
tsugou de ikkai sika
tukaenai youni natteimasu
/* 学生在校习得魔法时必须要有和人数相等的登录设备 */
gakkou de seito ga mahou wo
oboeru toki ha ninzuu bunn no
touroku media ga hitsuyou
/* 使用魔法时请小心不要对着路人施放哦 */
mahou wo tsukau tokiha
hito ni mukete utanai youni

```

ki wo tsuke te tsukatte ne

上面的几处百分比看上去很奇怪，这是因为游戏程序中有一个将非负整数转换为摇篮文的函数：

```
public static string nel_num(int t, bool zero_empty = false){ // 即下面的 NEL.nel_num
    if (t == 0)    // 单独的零是句点'.'
        return !zero_empty ? "." : "";
    if (t < 10)    // 一到九是单个字符'1'~'9'
        return t.ToString();
    if (t < 100)   // 十是字符'0'，十一到十九是'01'~'09'，21~99是'201'~'909'
        return (t < 20 ? "" : (t / 10).ToString()) + "0" + NEL.nel_num(t % 10, true);
    if (t < 1000)  // 百是字符'!'，111是'!01'，999是'9!909'
        return (t < 200 ? "" : (t / 100).ToString()) + "!" + NEL.nel_num(t % 100, true);
    if (t < 10000) // 千是字符'#'，1111是'#!01'，9999是'9#9!909'
        return (t < 2000 ? "" : (t / 1000).ToString()) + "#" + NEL.nel_num(t % 1000, true);
        // 万是字符'$'，11111是'$#!01'，99999是'9$9#9!909'
    return t < 100000 ? (t < 20000 ? "" : (t / 10000).ToString()) + "$" + NEL.nel_num(
        t % 10000, true) : NEL.nel_num(t / 100000, true) + NEL.nel_num(t % 100000, true);
}
```

修改前述的 txt 文件，即可在游戏中（按 F7 键热刷新）看到 49 个 ASCII 字符¹，包括 13 个数字（一到十、百千万）、26 个拉丁字母（不区分大小写）、9 个符号以及空格（全角宽度）。

从上述摇篮文中的最后六段话可以看出，日语的罗马字也可以用摇篮文转写，但是拼写规则很不统一。比如同时存在 toshi（とし，年龄）、ensyutu（えんしゅつ，演出）、zyuuzin tachi（じゅうじんたち，兽人们）、kakuritsu（かくりつ，确立）、simin ga ansin site（市民が安心して）、tsugou（つごう，出于）、ninzuu（にんずう，人数）。特别是つかい（使用）一词的变形中 tu 和 tsu 混用，非常混乱。

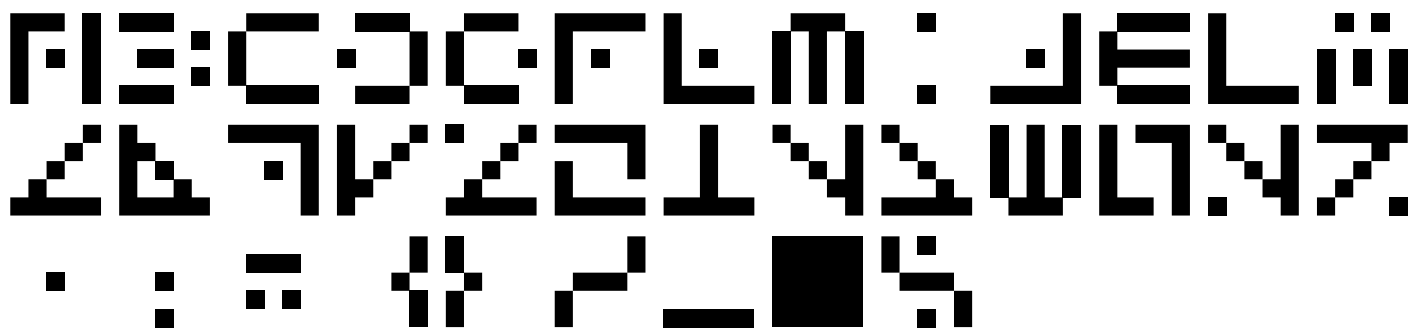
另一方面，日语用摇篮文转写以后需要用空格分词，は、へ、を会无视读音一律转写为 ha, he, wo，同时还会混入 media 和 security 这样的英文单词，后果是像 site、take、made 这样的词会看不出是哪种语言。

最后，摇篮文中正整数的写法也像极了日语，和汉语的最大区别是**所有“零”以及不在个位或万位的“一”都要省略**。不过上面的程序代码写得有问题，只能正确表示小于十万的正整数。

将 StreamingAssets 文件夹下的 Pxl 子文件夹中的 _icons.pxls.dat 文件用 AssetStudio 等第三方软件解包后可以得到一个 _icons.pxls 文件（多余的后缀名如 bytes 要删去），再用游戏作者自研的像素画工具 PixelLiner 打开，在左下角找到名为 nel_character_2 的姿势（posture, pose），就能看到上述 48 个字符（空格除外）的字形。此时按下 Ctrl+Shift+P（macOS 系统为 Command+Shift+P）打开保存对话框，将最左边的列数由 4 改为 13，并选中调色板右边的“背景色”勾选框，最后点击最下面的 OK 就能得到类似下图但黑白相反的图片（摇篮文像素体）。

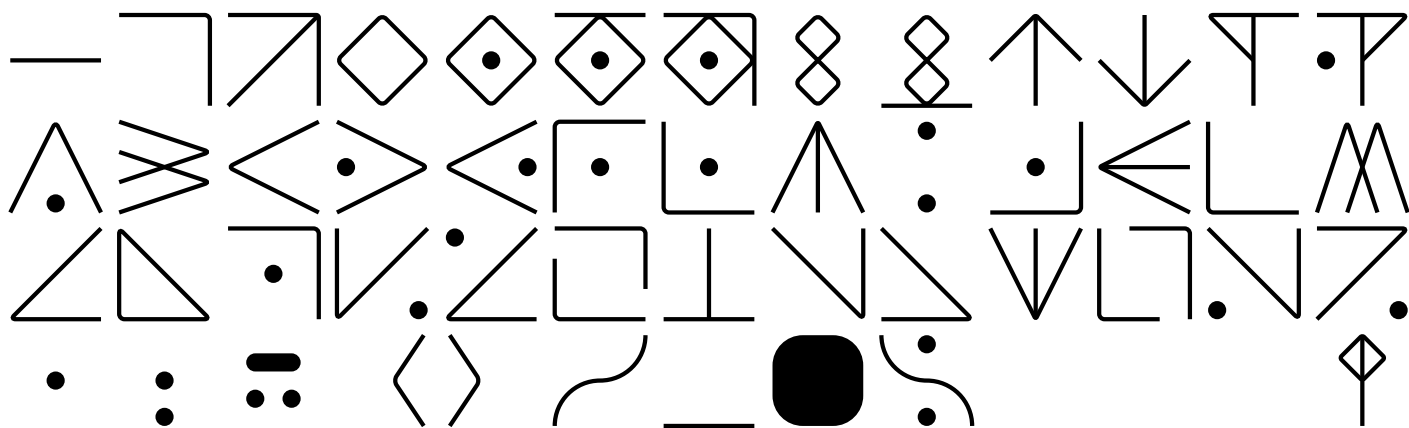


¹只有 0.20 和 0.21 中“圣光爆发”的习得动画可以反复观看，然而这两个版本中 J 和 K 弄反了。



上图中第一行是一到九、十百千万，中间两行是 A 到 Z，最后一行依次是 “.: - □ / _ , %”。可以看出，容易被混淆的有 FGJP 和 QRYZ 这两组。和 W 上下对称的居然不是 M 而是 H，需要特别注意。

除了像素体，摇篮文也有印刷体。因为在游戏中并不完整，因此下图作为玩家社区推断的结果仅供参考。印刷体比起像素体的主要区别是多了一些非对角倾斜的线条（尤其是 B 和 M），不过某些位于边角的圆点也许往中心靠一些更适合手写。另外右下角那个符号是法杖，没有对应的像素体。



为了方便地在 \LaTeX 中输入摇篮文，我自定义了名为 `cradle` 的宏包，使用它后可以直接用命令 `\pixelXX` 或 `\cradleXX` 输入某个摇篮文字符的像素体或印刷体，其中 `XX` 为字符的 ASCII 码（两位十进制数）。命令 `\cane` 输入印刷体的法杖符号。此外有四个调节字号的命令，它们是（花括号内为默认值，可以修改）：

```
\renewcommand\pixelcm{.72}      % 像素体总宽度(cm)含两侧空白
\renewcommand\pixelratio{1.2}    % 像素体总宽度除以非空白宽度，必须大于1
\renewcommand\cradlecm{.72}     % 印刷体总宽度(cm)含两侧空白
\renewcommand\cradleratio{1.2}  % 印刷体总宽度除以非空白宽度，必须大于1
```

为了进一步的使用方便，可以用以下 JavaScript 代码在浏览器中直接将一个 ASCII 字符串转换成上述命令序列：

```
var f = function (a, isPixel = false) {
  return Array.from(a).map(function (c) {
    switch (c) { // 括号一律视为方的
      case '(': case '<': case '{': c = '['; break;
      case ')': case '>': case '}': c = ']'; break;
    }
  })
}
```

```

    c = c.codePointAt(0);
    if (97 <= c && c <= 122) c -= 32; // 小写一律视为大写
    if (!(44 <= c && c <= 58 || 65 <= c <= 91 || [33,35,36,37,93,95].includes(c)))
        c = 32; // 不存在于 48 个中的字符一律视为空格
    return (isPixel ? '\\pixel' : '\\cradle') + c;
  }).join('');
} // 使用方法: console.log(f('Yin Shua')) 或 console.log(f('Xiang Su', true))

```

例句如下:

```
console.log(f('Ixia and Alma are 2 female elves'))
```

```
:[]:Λ Λ<> ΛLΛΛ ΛZ< 7 F<MAL< <L><□
```

```
console.log(f('Noel Cornehl is their best friend', true))
```

```
ΛBCL CBZΛCML :□ LMC:Z 3:C2L FZ:CΛ□
```