

luastg+ 迁移向导

——从luastg到luastg+，这是一个体力活

不兼容功能及迁移方法

语言不兼容性

luajit使用更为严格的语法，去除了lua5.1之前所用的可变参特性。

如下列代码将报错（找不到arg）：

```
function LoadImage(img,...)
    ImageList[img] = arg
    OriginalLoadImage(img,...)
end
```

- 影响范围：`core.lua`、其他使用`arg`访问可变参的地方
- 迁移方法：可以使用下述代码创建局部变量`arg`

```
local arg = { ... }
```

编码不兼容

luastg+使用UTF-8作为编码，这意味着所有的lua代码、所有的zip资源中的文件名都必须使用UTF-8作为编码。

注意：英文字符集包含在UTF-8中，这意味着只要修改有中文出现的文件即可，此外请勿存储为UTF-8 BOM。

- 影响范围：所有文件
- 迁移方法：使用文本编辑器更换文件编码并保存、使用好压切换到UTF-8代码页然后重新保存ZIP文件

第三方库使用

由于luajit的DLL名称为lua.dll，第三方库将无法链接。

- 影响范围：`require`方法
- 迁移方法：复制`lua.dll`为`lua51.dll`和`lua5.1.dll`并置于同级目录中

外部模块lua filesystem已被集成到核心

为方便使用，lfs已被集成到核心，现在起可以直接使用lfs而无需导入。

- 影响范围：`launch`
- 迁移方法：移除`require("lfs")`并删除`lfs.dll`

命令行参数传递至lstg.args

- 影响范围：`launch`
- 迁移方法：替换所有`arg`到`args`

wav文件兼容性问题

- 影响范围：所有波形文件
- 迁移方法：转码到44100Hz16bit立体声
- 举例：

使用ffmpeg进行转码当前目录下所有wav文件到`out*.wav`

请保存下述代码为`convert.bat`到se目录并执行

```
mkdir out
for %c in (*.wav) do (ffmpeg -y -i "%c" -acodec pcm_s16le -ac 2 -ar 44100 -f wav "out\%c")
```

mp3不再受支持

luastg+使用ogg作为唯一支持的音频格式，不再支持mp3格式。

- 影响范围：所有mp3文件
- 迁移方法：转码到ogg
- 举例：

参照上文，使用ffmpeg进行转码

```
mkdir out
for %c in (*.mp3) do (ffmpeg -y -i "%c" -acodec libvorbis -aq 8 -ac 2 -ar 44100 -f ogg "out\%c")
```

注：测试中发现使用ffmpeg转码出来的ogg文件不完全被兼容，如果出现无法载入的情况请考虑更换编码器。

游戏主循环分离

luastg+中FrameFunc被分离为FrameFunc和RenderFunc。渲染和逻辑发生分离，对迁移造成最大麻烦。

- 影响范围：`core.lua`和`ext.lua`、其他隐含问题（如变量未初始化等，这通常由编码不严格造成）
- 迁移方法：分离DoFrame中处理渲染的部分，将其置于全局函数RenderFunc中。
- 已知隐含问题：

于`ext.lua`中，`AfterRender`方法中的`ext.mask_color`未初始化导致在焦点丢失时引发错误。

音量不一致

luastg/luastg+的音量响度不一致，需要手动进行调整。

- 影响范围：音效播放、音乐播放

不兼容函数及迁移方法

LoadTTF行为变更

请参见API手册以获得更多信息。

- 影响范围：`LoadTTF`
- 迁移方法：修改`core.lua`中的`LoadTTF`函数
- 举例：

`core.lua`中的原始代码：

```
function LoadTTF(ttfname,filename,faceName,size,weight,...)
    ExtractRes(filename,'font\\'..ttfname)
    RegTTF('font\\'..ttfname)
    local deco=0
    local arg = {...}
    for i=1,#arg do deco=deco+ENUM_TTF_DECO[arg[i]] end
    lstg.LoadTTF(ttfname,faceName,0,size*screen.scale,weight,deco,setting.charset)
end
```

修改后的代码：

```
function LoadTTF(ttfname,filename,size)
    lstg.LoadTTF(ttfname,filename,0,size*screen.scale)
end
```

不再支持加粗、斜体等设置。上述修改后需要修改`font.lua`的加载部分：

```
LoadTTF('boss_name', 'THlib\\UI\\font\\default_ttf', '方正黑体_GBK', 10, 600)
LoadTTF('sc_name', 'THlib\\UI\\font\\default_ttf', '方正黑体_GBK', 13, 400)
LoadTTF('sc_pr', 'THlib\\UI\\font\\default_ttf', '方正黑体_GBK', 16, 400)
LoadTTF('dialog', 'THlib\\UI\\font\\default_ttf', '方正黑体_GBK', 16, 400)
```

修改为：

```
LoadTTF('boss_name', 'THlib\\UI\\font\\default_ttf', 10)
LoadTTF('sc_name', 'THlib\\UI\\font\\default_ttf', 13)
LoadTTF('sc_pr', 'THlib\\UI\\font\\default_ttf', 16)
LoadTTF('dialog', 'THlib\\UI\\font\\default_ttf', 16)
```

其他文件（如`editor.lua`）中也可能需要进行相关修改。

RenderTTF行为变更

请参见API手册以获得更多信息。

- 影响范围：`RenderTTF`
- 迁移方法：修改`core.lua`中的`RenderTTF`函数
- 举例：

`core.lua`中的原始代码：

```
function RenderTTF(ttfname,text,left,right,bottom,top,color,...)
    if lstg.viewmode=='world' then
        left,bottom=WorldToScreen(left,bottom)
        right,top=WorldToScreen(right,top)
    elseif lstg.viewmode=='3d' then
        error('Can not render true type font in 3d view mode.')
    end
    local fmt=0
    local arg = {...}
    for i=1,#arg do fmt=fmt+ENUM_TTF_FMT[arg[i]] end
    lstg.RenderTTF(ttfname,text,left*screen.scale+screen.dx,setting.resy-(top*screen.scale+screen.dy),right*screen.scale+screen
end
```

修改后的代码（去除手动坐标转换）：

```
function RenderTTF(ttfname,text,left,right,bottom,top,color,...)
    local fmt=0
    local arg = {...}
    for i=1,#arg do fmt=fmt+ENUM_TTF_FMT[arg[i]] end
    lstg.RenderTTF(ttfname,text,left,right,bottom,top,fmt,color)
end
```