

1 库文件下载

<http://www.minigui.org/zhcn/download/>

iphone-like-demo-1.0.0.tar.gz

libminigui-gpl-3.2.tar.gz minigui 核心库文件

mg-samples-3.2.tar.gz minigui 例子文件

minigui-res-be-3.2.tar.gz minigui 资源文件

libmgplus-1.4.0.tar.gz minigui 高级图形功能

libmgi-2.0.4.tar.gz minigui 的输入法

libmgutils-1.2.0.tar.gz

libmgeff-1.2.0.tar.gz

libmgncs-1.2.0.tar.gz

mgncs4touch-1.2.0.tar.gz

freetype-2.3.9-fm20100818.tar.gz

zlib-1.2.8.tar.gz

jpegsrc.v7.tar.gz

libpng-1.2.37.tar.gz

libchipmunk-5.3.1.tar.gz

mgallery-1.0.tar.gz

sqlite-autoconf-3120200.tar.gz

环境:

Ubuntu 14.04.5 LTS 64 位

MiniGui3.0.12

Qt-3.3.0

2 lib 安装

首先安装编译 32 位程序的库，因为 minigui 编译成 32 位程序比较容易，如果编译换成 64 位程序，会有很多错误需要修改，安装之前，建议**先把本文看完**。

```
sudo apt-get install g++-multilib ia32-libs libc6-dev-i386 linux32
```

可能有些库没有，但是控制台会提示替代能替代的库，安装就好

libpng, libjpeg, zlib, freetype 这四个库都是解压后执行下面的命令，安装到/usr/local/lib

```
CC="gcc -m32" ./configure  
make  
sudo make install
```

3 qvfb 安装

安装依赖的第三方库

```
sudo apt-get install libx11-dev libxext-dev libxtst-dev
```

```
sudo apt-get install build-essential subversion git-core  
libncurses5-dev zlib1g-dev gawk flex quilt libssl-dev xsltproc  
libxml-parser-perl mercurial bzip2 ecj cvs unzip ia32-libs
```

3.1 qt-4.8.6

不需要安装 qt-4.8.6，此处记录我艰难的安装过程，直接安装 qt-3.3.0 即可。

源码下载链接 <http://download.qt.io/archive/qt/4.8/4.8.6/>

Name	Last modified
↑ Parent Directory	
qt-opensource-windows-x86-vs2010-4.8.6.exe	06-May-2014 12:52
qt-opensource-windows-x86-vs2008-4.8.6.exe	06-May-2014 12:51
qt-opensource-windows-x86-mingw482-4.8.6-1.exe	03-Jun-2014 10:31
qt-opensource-mac-4.8.6-1.dmg	03-Jun-2014 10:30
qt-opensource-mac-4.8.6-1-debug-libs.dmg	03-Jun-2014 10:30
qt-everywhere-opensource-src-4.8.6.zip	06-May-2014 12:51
qt-everywhere-opensource-src-4.8.6.tar.gz	06-May-2014 12:51

选择文件 qt-everywhere-opensource-src-4.8.6.tar.gz，解压并进入文件夹，安装到/home/anruli/software/qt-4.8.6 这个文件夹下

```
CC="gcc -m32" CXX="g++ -m32" ./configure -opensource -prefix  
/home/anruli/software/qt-4.8.6
```

CC="gcc -m32" CXX="g++ -m32"是编译成 32 位的，实际上不需要编译成 32 位程序，去掉即可，之后出现开源许可，输入 yes 接受，通过库文件检查之后输入

```
make && make install
```

编译通过，安装完成后，需要将 qt 的库和可执行文件路径配置到环境变量中，否则在安装 qvfb 时会报错

```
sudo gedit ~/.bashrc
```

在文件最后面后面加上

```
export PATH=/home/anruli/software/qt-4.8.6/bin:$PATH
```

使环境变量生效

```
source ~/.bashrc
```

进入 qt4.8.6 源码目录下的 tools/qvfb，执行 qmake 命令

```
qmake
```

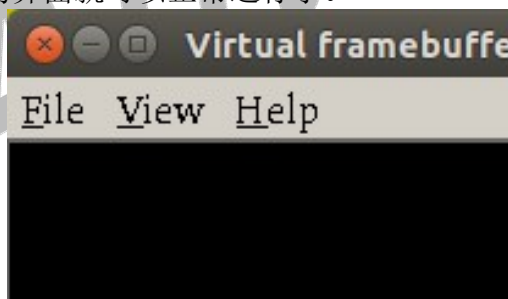
如果出现错误找不到 qmake 命令，则是因为环境变量配置没有生效，生效之后报错看下面的错误，qmake 执行成功，执行 make 命令，开始编译，编译前可以在 /tools/qvfb/main.cpp 中更改 width、height 和 depth 的值

```
make
```

编译成功后，将源码目录 bin 下的 qvfb 文件拷贝到 qt 安装目录 bin 下，双击 qvfb 就可以运行，也可以在 bin 目录下执行

```
./qvfb
```

qvfb 安装成功。在需要使用 minigui 前，先打开 qvfb 模拟工具，让其在后台运行，这样 minigui 的界面就可以正常运行了。



错误:

```
qmake: could not exec '/usr/lib/x86_64-linux-gnu/qt4/bin/qmake': No such file or directory
```

解决:

```
sudo 打开/usr/lib/x86_64-linux-gnu/qt-default/qtchooser/default.conf
```

```
修改第 一 行 /usr/lib/x86_64-linux-gnu/qt4/bin 为 /home/anruli/software/qt-4.8.6/bin
```

错误:

```
undefined reference to `png_create_write_struct`  
undefined reference to `png_create_info_struct`  
undefined reference to `png_set_compression_level`  
undefined reference to `png_set_write_fn`
```

解决:

打开/tools/qvfb/Makefile 文件, 找到 LIBS 变量, 增加 `-L/usr/local/lib -lpng`

错误:

```
/usr/bin/ld: skipping incompatible  
/home/anruliu/workspace/minigui/qt-everywhere-opensource-src-4.8.6/src/t  
ools/bootstrap/libbootstrap.a when searching for -lbootstrap  
/usr/bin/ld: cannot find -lbootstrap
```

解决:

打开/src/tools 目录下的 `idc`, `moc`, `rcc`, `uic`, `uic3` 文件夹里面的 Makefile 文件, 把 `LINK=g++` 改为 `LINK=g++ -m32`

打开/src 目录下的 `corelib`、`xml`、`network`、`sql`、`testlib`、`gui`、`qt3support`、`xmlpatterns`、`multimedia`、`svg`、`script`、`declarative`、`scripttools`、`plugins`、`imports` 的 Makefile 文件也修改, 其中 `plugins` 和 `imports` 的子目录下也有很多 Makefile, 注意不要漏了。还有很多很多处需要修改, 根据出错的路径更改即可!!!

因为 CC 和 CXX 在./configure 的时候都改成 -m32 的了, 所以 LINK 也要改。谁有更好修改 LINK 的方法请告诉我, 感激不尽!!! 要改那么多...手都快改肿了。

错误:

```
/usr/include/openssl/e_os2.h:56:33: fatal error: openssl/opensslconf.h: No  
such file or directory
```

```
#include <openssl/opensslconf.h>
```

解决:

将/usr/include/x86_64-linux-gnu/openssl/opensslconf.h 复制到
/usr/include/openssl 目录下

错误:

```
/usr/bin/ld: cannot find -lX11  
/usr/bin/ld: cannot find -lXtst  
/usr/bin/ld: cannot find -lXext
```

解决:



秘密

lX11 与 lXtst 没有安装，安装即可

```
sudo apt-get install libx11-dev:i386
sudo apt-get install libxtst-dev:i386
```

使用命令查看 lXext 有没有安装

```
locate Xext
```

```
/usr/include/X11/extensions/Xext.h
/usr/lib/i386-linux-gnu/libXext.so.6
/usr/lib/i386-linux-gnu/libXext.so.6.4.0
/usr/lib/x86_64-linux-gnu/libXext.a
/usr/lib/x86_64-linux-gnu/libXext.so
/usr/lib/x86_64-linux-gnu/libXext.so.6
/usr/lib/x86_64-linux-gnu/libXext.so.6.4.0
```

发现在 /usr/lib/i386-linux-gnu/ 有 libXext 库，但是没有链接，制作链接即可

```
sudo ln -s libXext.so.6 libXext.so
```

3.2 qt-3.3.0

源码下载链接 <http://download.qt.io/archive/qt/3/>

qt-x11-free-3.3.2.tar.bz2	13-Dec-2012 14:12	14M
qt-x11-free-3.3.1.tar.gz	13-Dec-2012 14:12	17M
qt-x11-free-3.3.1.tar.bz2	13-Dec-2012 14:12	14M
qt-x11-free-3.3.0.tar.gz	13-Dec-2012 14:12	17M
qt-x11-free-3.3.0.tar.bz2	13-Dec-2012 14:12	14M
qt-x11-free-3.2.3.tar.gz	13-Dec-2012 14:12	16M

选择文件 qt-x11-free-3.3.0.tar.gz，解压并进入文件夹，安装到 /home/anruli/software/qt-3.3.0 这个文件夹下

```
./configure --prefix /home/anruli/software/qt-3.3.0
```

之后出现开源许可，输入 yes 接受，通过库文件检查之后输入

```
make && make install
```

编译通过，安装完成后，需要将 qt 的库和可执行文件路径配置到环境变量中，否则在安装 qvfb 时会报错

```
sudo gedit ~/.bashrc
```

在文件最后面后面加上

```
export PATH=/home/anruli/software/qt-3.3.0/bin:$PATH
```

全志科技版权所有，侵权必究

Copyright © 2018 by Allwinner. All rights reserved

使环境变量生效

```
source ~/.bashrc
```

进入 qt3.3.0 源码目录下的 tools/qvfb, 执行 qmake 命令

```
qmake
```

如果出现错误找不到 qmake 命令, 则是因为环境变量配置没有生效, 生效之后报错看下面的错误, qmake 执行成功, 执行 make 命令, 开始编译, 编译前可以在 /tools/qvfb/main.cpp 中更改 width、height 和 depth 的值, 我改的值是

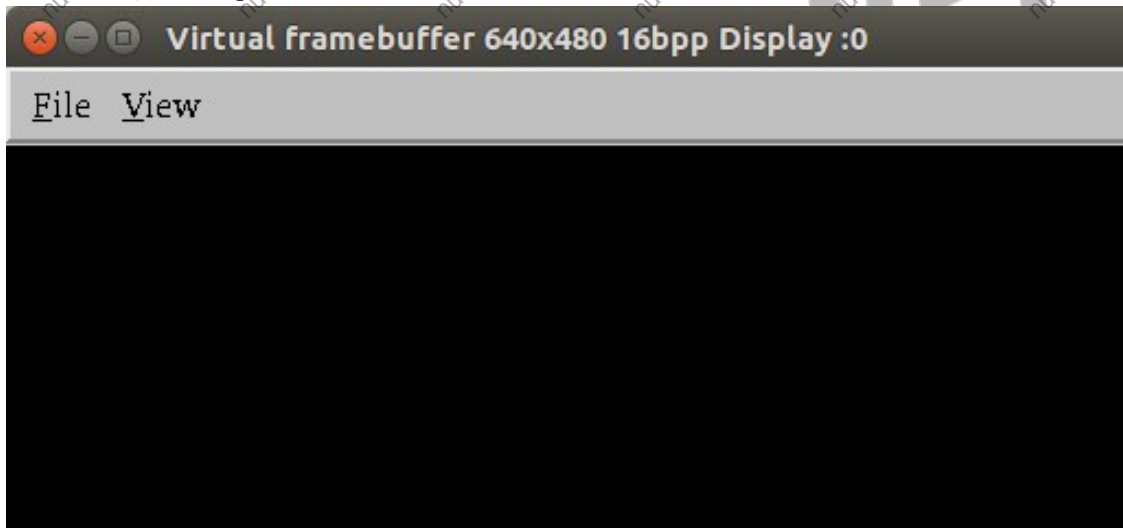
```
int width = 640;int height = 480;int depth = 16;
```

```
make
```

编译成功后, 将源码目录 bin 下的 qvfb 文件拷贝到 qt 安装目录 bin 下, 双击 qvfb 就可以运行, 也可以在 bin 目录下执行

```
./qvfb
```

qvfb 安装成功。在需要使用 minigui 前, 先打开 qvfb 模拟工具, 让其在后台运行, 这样 minigui 的界面就可以正常运行了。



错误:

```
../include/qvaluelist.h:88:13: error: 'ptrdiff_t' does not name a type  
typedef ptrdiff_t difference_type;
```

解决:

打开/include/qvaluelist.h 与 qmap.h, 在最上面加入#include <cstdlib>

错误:

```
error while loading shared libraries: libqt.so.3: cannot open shared object
```



秘密

file: No such file or directory

解决:

export LD_LIBRARY_PATH='/home/anruliu/workspace/minigui/qt-x11-free-3.3.0/lib'
把这库路径添加到环境变量中

错误:

```
qvfview.cpp: In constructor 'QVfView::QVfView(int, int, int, int, QWidget*, const char*, uint)':  
qvfview.cpp:118:15: error: cast from 'unsigned char*' to 'int' loses precision [-fpermissive]  
    if ( (int)data == -1 )
```

解决:

打开/tools/qvfb/qvfview.cpp,

把 if ((int)data == -1) 改为 if ((long)data == -1)

如果还遇到其他编译错误, 请查看本文上一小节 [qt-4.8.6 编译问题记录](#)

4 minigui 安装

4.1 minigui-res-be-3.2

进入到解压目录

```
CC="gcc -m32" ./configure  
make  
sudo make install
```

4.2 libminigui-gpl--3.2

进入到解压目录, 注意-m32, 把这个库编成 32 位的, 不然之后会遇到很多错误的!! 还有 enable-qvfbial 与 enable-videoqvfb, 表示启用 qvfb 功能!! 如果加上 --enable-pngsupport --enable-jpgsupport 表示支持 png 与 jpeg, 要显示飞漫公司启动动画则要加上 png 与 jpeg 支持, --disable-splash --disable-screensaver 不显示飞漫公司的启动动画

```
./autogen.sh  
./configure --with-runmode=ths --enable-qvfbial --enable-videoqvfb  
--disable-splash --disable-screensaver --enable-pngsupport  
--enable-jpgsupport --with-ttfsupport=ft2  
--with-ft2-includes=/usr/local/include/freetype2
```

```
make  
sudo make install
```

4.3 libmgplus-1.2.4

进入到解压目录

```
./autogen.sh  
./configure  
make  
sudo make install
```

4.4 libmgi-2.0.4

进入到解压目录

```
./configure  
make  
sudo make install
```

4.5 libmgutils-1.2.0

进入到解压目录

```
./autogen.sh  
./configure  
make  
sudo make install
```

4.6 libmgncs-1.2.0

安装这个库之前，先安装 libmgutils-1.2.0

进入到解压目录

```
./autogen.sh  
./configure  
make  
sudo make install
```

4.7 libmgeff-1.2.0

进入到解压目录

```
./autogen.sh
```

```
./configure  
make  
sudo make install
```

4.8 libchipmunk-5.3.1

进入到解压目录

```
cmake XXX(安装路径,通常这个路径指定为解压目录的路径)  
make  
sudo make install
```

错误:

```
[ 72%] Linking C executable chipmunk_demos  
/usr/bin/ld: CMakeFiles/chipmunk_demos.dir/MagnetsElectric.c.o:  
undefined reference to symbol 'cos@@GLIBC_2.2.5'  
//lib/x86_64-linux-gnu/libm.so.6: error adding symbols: DSO missing  
from command line  
collect2: error: ld returned 1 exit status  
Demo/CMakeFiles/chipmunk_demos.dir/build.make:620: recipe for target  
'Demo/chipmunk_demos' failed  
make[2]: *** [Demo/chipmunk_demos] Error 1  
CMakeFiles/Makefile2:179: recipe for target  
'Demo/CMakeFiles/chipmunk_demos.dir/all' failed
```

解决:

```
vim CMakeLists.txt 删除下面的内容  
if(BUILD_DEMOS)  
    add_subdirectory(Demo)  
endif(BUILD_DEMOS)
```

4.9 mgncs4touch-1.2.0

进入到解压目录

安装这个包之前,先安装 libmgncs-1.2.0 libchipmunk-5.3.1

```
./autogen.sh  
./configure  
make  
sudo make install
```

错误:

```
configure: error: Package requirements (chipmunk >= 5.3.1) were not met:
```



秘密

No package 'chipmunk' found

解决:

```
./configure          CHIPMUNK_CFLAGS=-I/usr/local/include/chipmunk
CHIPMUNK_LIBS=-lchipmunk
```

4.10 sqlite-autoconf-3120200

进入到解压目录

```
./autogen.sh
./configure
make
sudo make install
```

4.11 mg-samples-3.0.12

进入到解压目录

```
./autogen.sh
./configure
make
sudo make install
```

错误:

```
In file included from lf_skin.c:31:0:
lf_skin_res_en.h:18:8: error: unknown type name 'TVITEMINFO'
static TVITEMINFO ListViewInfo =
```

解决:

```
vim lf_skin/lf_skin_res_en.h
添加#include <minigui/ctrl/treeview.h>
```

错误:

```
taskbar.c:140:5: error: unknown type name 'COOLBARITEMINFO'
    COOLBARITEMINFO bar_item;
```

解决:

```
vim mginit/taskbar.c
添加#include <minigui/ctrl/coolbar.h>
```



秘密

错误:

```
gcc -g -O2 -Wstrict-prototypes -pipe -o worldtransform worldtransform.o  
@APP_LIBS@ /usr/local/lib/libmngplus.so -L/usr/local/lib -lpthread  
-lstc++ /usr/local/lib/libminigui_procs.so /usr/local/lib/libjpeg.so  
-lpng -lm -lpng12 /usr/local/lib/libfreetype.so -lz -Wl,--rpath  
-Wl,/usr/local/lib -Wl,--rpath -Wl,/usr/local/lib  
gcc: error: @APP_LIBS@: No such file or directory
```

解决:

```
vim mngplus/worldtransform/Makefile.am  
将 worldtransform_LDADD =-lmngplus @APP_LIBS@  
替换为 worldtransform_LDADD =-lmngplus
```

错误:

```
/usr/bin/ld: cannot find -lsqlite3  
collect2: error: ld returned 1 exit status
```

解决:

先安装 sqlite-autoconf-3120200 这个包

错误:

```
mginit.c: In function 'on_new_del_client':  
mginit.c:44:15: error: 'LCO_NEW_CLIENT' undeclared (first use in this  
function)
```

```
if (op == LCO_NEW_CLIENT) {
```

解决:

运行里面的应用出现的问题:

错误:

```
KERNEL>InitGUI (step 4): Can not start client (Please run mginit first)!
```

解决:

因为 libminigui-gpl 没有编译成多线程模式

错误:

```
execl: No such file or directory  
NEWGAL>PCXVFB: Wait too long for CLIENT.  
NEWGAL: Does not find matched engine: pc_xvfb.  
KERNEL>InitGUI: Can not get graphics engine information!
```

全志科技版权所有, 侵权必究

Copyright © 2018 by Allwinner. All rights reserved

解决:

```
sudo gedit /usr/local/etc/MiniGUI.cfg  
gal_engine=pc_xvfb 改为 gal_engine=qvfb  
ial_engine=pc_xvfb 改为 ial_engine=qvfb  
cursorpath=/usr/local/share/minigui/res/cursor/  
respath=/usr/local/share/minigui/res/
```

错误:

/lib/libminigui_ths.so: undefined reference to 'QVFB_bootstrap'

解决:

主要原因是 libminigui-gpl-3.0.12.tar.gz 下载错了, 重新下载
其次确保 libminigui 编译时指定了 --enable-qvfbial --enable-videoqvfb

5 配置

```
sudo gedit /usr/local/etc/MiniGUI.cfg  
gal_engine=pc_xvfb 改为 gal_engine=qvfb  
ial_engine=pc_xvfb 改为 ial_engine=qvfb  
cursorpath=/usr/local/share/minigui/res/cursor/  
respath=/usr/local/share/minigui/res/
```

将默认配置修改为使用 qvfb 模拟器, 配置资源文件目录, 将光标目录和源目录值修改为实际的安装目录, 保存退出. 我安装的时候 libminigui-gpl 没有指定路径, 所以在 MiniGUI.cfg 在默认路径下

4.12 cell-phone-ux-demo

进入到解压目录

```
./autogen.sh  
./configure CHIPMUNK_CFLAGS=-I/usr/local/include/chipmunk CHIPMUNK_LIBS=-lchipmunk
```

```
make
```

```
sudo make install
```

错误:

execl: No such file or directory

```
NEWGAL>PCXVFB: Wait too long for CLIENT.  
NEWGAL: Does not find matched engine: pc_xvfb.  
KERNEL>InitGUI: Can not get graphics engine information!
```

解决:

```
vim /解压目录/MiniGUI.cfg  
gal_engine=pc_xvfb 改为 gal_engine=qvfb  
ial_engine=pc_xvfb 改为 ial_engine=qvfb  
cursorpath=/usr/local/share/minigui/res/cursor/  
respath=/usr/local/share/minigui/res/
```

6 测试

进入/mg-samples-3.0.12/same, &表示后台执行
qvfb -width 640 -height 480 -depth 16 &
成功运行 qvfb 之后, 运行例子程序
./same

报错:

```
./notebook: error while loading shared libraries: libminigui_ths-3.0.so.12:  
cannot open shared object file: No such file or directory  
但是/usr/local/lib 目录下是有 libminigui_ths-3.0.so.12 的, 明显是因为找不到库路径, 需要增加搜索库路径
```

解决:

```
sudo vim /etc/ld.so.conf  
追加:include /usr/local/lib  
sudo ldconfig
```

报错:

```
NEWGAL: Does not find matched engine: qvfb.  
KERNEL>InitGUI: Can not get graphics engine information!
```

解决:



秘密

MiniGUI3.0.12版本默认情况下是不开 QVFB 图像引擎与 QVFB 输入引擎的，要加 `--enable-videoqxfb --enable-qvfbial` 配置项，编译 `libminigui-gpl--3.0.12` 的时候

报错：

NEWGAL: Video mode smaller than requested.

NEWGAL: Set video mode failure.

KERNEL>InitGUI: Can not initialize graphics engine!

解决：

运行 `qvfb` 的时候指定宽高和多少位和 `/usr/local/etc/MiniGUI.cfg` 文件中一样，或者修改 `MiniGui.cfg` 中 `defaultmode=640x480-16bpp` （保证这里的参数和 `qvfb` 配置中的分辨率一样）

报错：

KERNEL>InitGUI: Count not init mouse cursor!

KERNEL>InitGUI: Init failure, please check your MiniGUI configuration or resource.

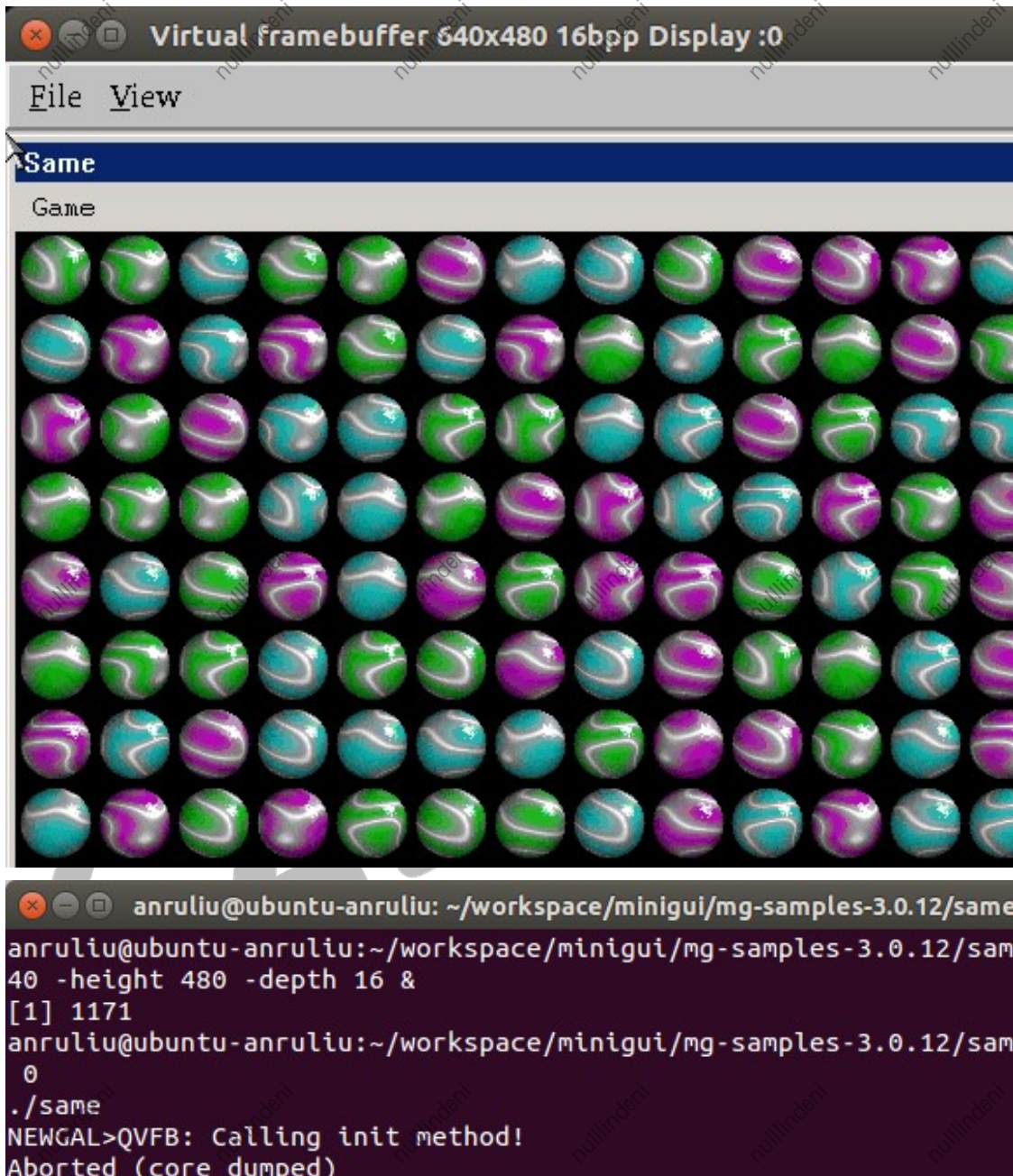
解决：

确保 `/usr/local/etc/MiniGUI.cfg` 中下面两个值，和实际安装路径一样

`cursorpath=/usr/local/share/minigui/res/cursor/`

`respath=/usr/local/share/minigui/res/`

问题解决后出现画面，很开心啊！！



注意看最后一句 **Aborted (core dumped)** 段错误，程序异常终止了！使用 `gdb` 调试

`(gdb) file same`

Reading symbols from same...done.

`(gdb) run`

Starting program: /home/anruliu/workspace/minigui/mg-samples-3.0.12/same/same

[Thread debugging using libthread_db enabled]

Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

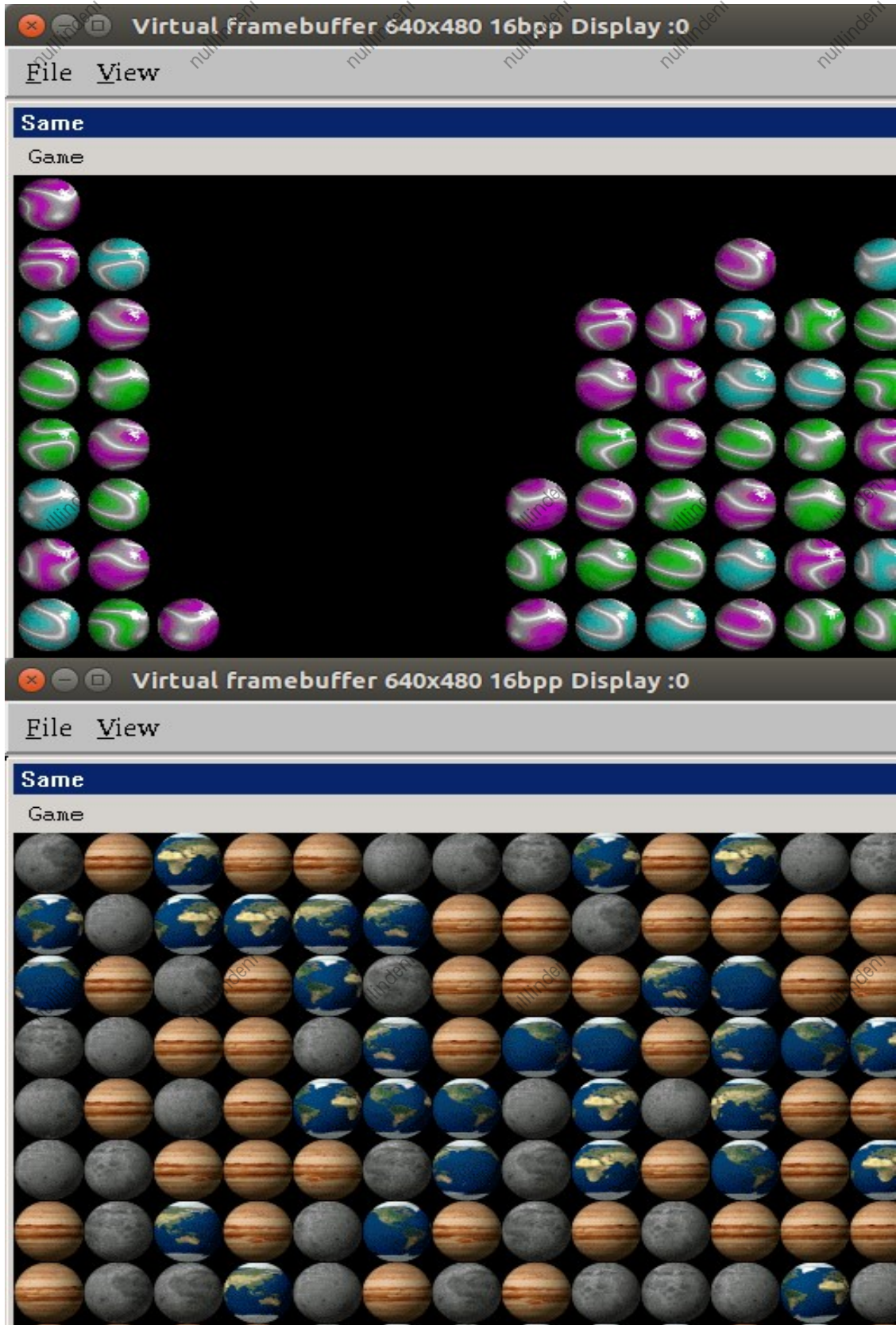


秘密

```
[New Thread 0xf7a9ab40 (LWP 30606)]
[New Thread 0xf70ffb40 (LWP 30607)]
[New Thread 0xf68feb40 (LWP 30608)]
Program received signal SIGSEGV, Segmentation fault.
0x00000000 in ?? ()
(gdb) backtrace
#0 0x00000000 in ?? ()
#1 0xf7eb59ec in GAL_VideoSync (rc=rc@entry=0xffffcc90) at video.c:1207
#2 0xf7e4645f in EndPaint (hWnd=hWnd@entry=134544608, hdc=hdc@entry=4160429420) at window.c:4814
#3 0x080496f1 in SameGameProc (hWnd=134544608, message=177, wParam=0, lParam=134544784)
    at same.c:607
#4 0xf7e718c3 in DispatchMessage (pMsg=pMsg@entry=0xffffcebc) at message.c:1061
#5 0x0804a158 in MiniGUIMain (argc=argc@entry=1, argv=argv@entry=0xffffcfff4) at same.c:732
#6 0x08048fb2 in main (args=1, argv=0xffffcfff4) at same.c:697
(gdb) frame 1
#1 0xf7eb59ec in GAL_VideoSync (rc=rc@entry=0xffffcc90) at video.c:1207
1207         video->VideoSync(video, rc);
```

从上面信息的最后一处可知，在 VideoSync(video, rc) 函数处程序崩溃了，打开 /libminigui-gpl-3.0.12/src/newgal/video.c 文件把 video->VideoSync(video, rc) 这句注释掉，然后重新编译安装 libminigui-gpl-3.0.12 就可以玩游戏了。

我分析由于是使用了 qvfb，打开 /libminigui-gpl-3.0.12/src/newgal/qvfb/qvfb.c，发现里面并没有 VideoSync 函数的实现，去调用它肯定会报错，而这个函数只在 /libminigui-gpl-3.0.12/src/newgal/fbcon/fbvideo.c 中有实现。如果使用 qt-4.8.6 的 qvfb 你会发现按钮不能点击，鼠标闪烁，但是如果换成 qt-3.3.0 的 qvfb，界面很流畅，按钮能点击，功能一切正常，可能是 minigui 很久没有更新，qt 一直在更新，兼容性不好导致的吧。





秘密

参考文献

- [1] [《MiniGui 开发: ubuntu-14.04 环境移植 MiniGui》](#)
- [2] [《 minigui 成功移植到 ubuntu64 位平台》](#)
- [3] [《MiniGUI3.0.12 绝对正确安装》](#)
- [4] [《minigui 安装及问题总结》](#)
- [5] [《MiniGui 开发: Ubuntu14.04 环境安装 qvfb 工具》](#)

÷

