Cocos2dx有五种屏幕适配模式：

EXACT\_FIT NO\_BORDER SHOW\_ALL FIXED\_HEIGHT FIX\_WIDTH

**EXACT\_FIT**

X轴与Y抽均拉伸到与屏幕大小相等

**NO\_BORDER**

保持比例，放缩较大的倍数 （screen/designResolution）

**SHOW\_ALL**

保持比例，放缩较小的倍数 （screen/designResolution）

**FIX\_WIDTH**

保持比例，同时保证宽度不变

**FIXED\_HEIGHT**

保持比例，同时保证高度不变

下面的例子中（图片大小为960\*640 屏幕窗口大小为500\*500 DesignResolutionSize = 960\*640）

同时查看各种尺寸的区别：

Size winSize = director->getWinSize();

Size visibleSize = director->getVisibleSize();

Size designResolutionSize = director->getOpenGLView()->getDesignResolutionSize();

Size screenSize = director->getOpenGLView()->getFrameSize();

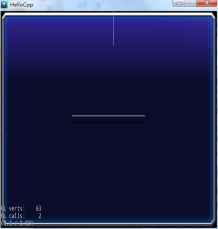
log("winSize width = %f, height = %f", winSize.width, winSize.height);

log("visibleSize width = %f, height = %f", visibleSize.width, visibleSize.height);

log("designResolutionSize width = %f, height = %f", designResolutionSize.width, designResolutionSize.height);

log("screenSize width = %f, height = %f", screenSize.width, screenSize.height);

**EXACT\_FIT**





**NO\_BORDER**



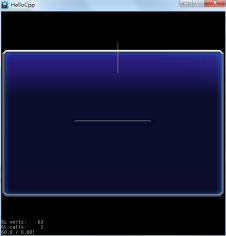


**SHOW\_ALL**





**FIX\_WIDTH**

****



**FIXED\_HEIGHT**





**源码分析：**

在上面的各种尺寸获取过程中：

winSize存于Director中

visibleSize、designResolutionSize与screenSize均存在与GLView中

注意getVisibleSize的获取方法：

Size GLView::getVisibleSize() const

{

if (\_resolutionPolicy == ResolutionPolicy::NO\_BORDER)

{

return Size(\_screenSize.width/\_scaleX, \_screenSize.height/\_scaleY);

}

else

{

return \_designResolutionSize;

}

}

**实际上，visibleSize就是ResolutionSize的可以看到的部分**

调用：

glview = GLViewImpl::createWithRect("HelloCpp", Rect(0, 0, 500, 500));

内部会调用：

void GLView::setFrameSize(float width, float height)

{

\_designResolutionSize = \_screenSize = Size(width, height);

}

也就是初始化状态View中的designResolutionSize与screenSize均为自己的设置，此时均为500\*500

调用

director->setOpenGLView(glview);

内部会调用：

\_winSizeInPoints = \_openGLView->getDesignResolutionSize();

也就是此时winSizeInPoints也设置为当前的designResolutionSize，此时为500\*500

调用

director->getOpenGLView()->setDesignResolutionSize(960, 640, ResolutionPolicy::EXACT\_FIT);

内部会调用：

\_designResolutionSize.setSize(width, height);

也就是此时designResolutionSize变为960\*640

内部还会调用：

director->\_winSizeInPoints = getDesignResolutionSize();

也就是此时winSizeInPoints变为960\*640

总结整个过程：

WinSize ScreenSize DesignResolutionSize

createGL 500\*500 500\*500

setOpenGLView 500\*500（通过designResolutionSize设置）

setDesignResolution 960\*640 960\*640

void GLView::setDesignResolutionSize(float width, float height, ResolutionPolicy resolutionPolicy)

{

CCASSERT(resolutionPolicy != ResolutionPolicy::UNKNOWN, "should set resolutionPolicy");

if (width == 0.0f || height == 0.0f)

{

return;

}

\_designResolutionSize.setSize(width, height);

\_resolutionPolicy = resolutionPolicy;

**updateDesignResolutionSize();**

}

**void GLView::updateDesignResolutionSize()**

{

if (\_screenSize.width > 0 && \_screenSize.height > 0

&& \_designResolutionSize.width > 0 && \_designResolutionSize.height > 0)

{

\_scaleX = (float)\_screenSize.width / \_designResolutionSize.width;

\_scaleY = (float)\_screenSize.height / \_designResolutionSize.height;

if (\_resolutionPolicy == ResolutionPolicy::NO\_BORDER)

{

\_scaleX = \_scaleY = MAX(\_scaleX, \_scaleY);

}

else if (\_resolutionPolicy == ResolutionPolicy::SHOW\_ALL)

{

\_scaleX = \_scaleY = MIN(\_scaleX, \_scaleY);

}

else if ( \_resolutionPolicy == ResolutionPolicy::FIXED\_HEIGHT) {

\_scaleX = \_scaleY;

\_designResolutionSize.width = ceilf(\_screenSize.width/\_scaleX);

}

else if ( \_resolutionPolicy == ResolutionPolicy::FIXED\_WIDTH) {

\_scaleY = \_scaleX;

\_designResolutionSize.height = ceilf(\_screenSize.height/\_scaleY);

}

// calculate the rect of viewport

float viewPortW = \_designResolutionSize.width \* \_scaleX;

float viewPortH = \_designResolutionSize.height \* \_scaleY;

\_viewPortRect.setRect((\_screenSize.width - viewPortW) / 2, (\_screenSize.height - viewPortH) / 2, viewPortW, viewPortH);

// reset director's member variables to fit visible rect

auto director = Director::getInstance();

director->\_winSizeInPoints = getDesignResolutionSize();

director->\_isStatusLabelUpdated = true;

director->setGLDefaultValues();

}

}