

Qt不规则窗体的实现 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/181/2021\\_2022\\_Qt\\_E4\\_B8\\_8D\\_E8\\_A7\\_84\\_E5\\_88\\_c103\\_181691.htm](https://www.100test.com/kao_ti2020/181/2021_2022_Qt_E4_B8_8D_E8_A7_84_E5_88_c103_181691.htm) Skin(表皮) 是制作比较酷的软件界面的有利工具. 一个软件可以同时使用多种Skin 以取得不同的外观, 使同一个软件有截然不同的风格. 用户可以根据自己的喜好选择 不同的风格. 本节介绍使用 Qt 制作 Skin 的方法. 软件界面的风格变化可以通过两种机制完成，一种是通过设置主题(Themes)，它使用 界面库本身所具有的对界面组件(Components)的控制能力切换显示风格；另外一种是通过提供不同系列的图片来切换显示界面，即这里所讲的 Skin。制作表皮有几个重要的因素值得考虑：1. 使用无边界的窗口在 XWindow 下，无边界的窗口是指不受窗口管理器管理的边界不规则的窗口。由于不受窗口管理器管理，所以软件窗口界面没有附加的标题条(Title Bar)等。在Qt中，建立无边界窗口的最简单的方法是设置 QWidget 的 WFlags 的值是 WStyle\_NoBorder。它定义在 qnamespace.h 中。不规则窗口的特点则要求对整个 窗口使用图像掩码。使用 X 窗口形状的扩展(X Shape Extension)来达到要求。在 Qt 中可以直接使用

```
, QPixmap bm.bm =  
*(Pixmaps[MASK]).setMask(bm).setBackgroundPixmap(*Pixmaps  
[BACKGROUND]).2. 窗口的移动由于上述窗口不受窗口管理  
器的管理的特性，所以移动窗口需要特殊处理，一般的方法  
是截取根Widget的鼠标按钮事件，自己处理鼠标点击和移动  
的事件。void SkinDemo::mouseMoveEvent(QMouseEvent  
*e){QPoint newpos = e->globalPos().QPoint upleft = pos0 newpos
```

```
- last.move(upleft).}void
```

```
SkinDemo::mousePressEvent(QMouseEvent *e){last =
```

```
e->globalPos().pos0 = e->globalPos() - e->pos().}这里我们取得的鼠标位置是绝对位置，即相对于根窗口的位置，同时也记录下窗口左上角的位置，当鼠标移动时，取得新的绝对位置，则窗口左上角的新位置应该是原来位置与鼠标移动的位置之差。
```

3. 按钮的制作在例子( qt-skin-example.tar.gz) 中，我们重新定义了鼠标按下和鼠标移动的事件处理函数，并且含有按钮的所有信息，主工作区的所有信息等。主工作区的信息是它的位置和尺寸，主窗口的信息是它所使用的背景图片和图片的掩码(用来制造不规则窗口)，所以整个主窗口的大小可以由图片的尺寸来决定。几个按钮的信息包含它们的位置，它们的大小由图片的大小来决定。对于表皮中的图像按钮，

设置它的父类是 QPushButton，这种按钮由两幅图片构成，一幅图片是正常状态(Normal)，一幅图片是按钮按下时的状态(Activated)。有时也可以设置成四种状态，即增加禁止状态(Disabled)和鼠标指针进入时的状态 (Hovered)。Qt3 的手工代码实现：

```
CODE:#include #include #include #include #include
class myclass:public QWidget{ public: myclass(). protected: void
mouseMoveEvent(QMouseEvent *e). void
mousePressEvent(QMouseEvent *e). private: QPixmap *pixmap.
QBitmap *bitmap. QPoint last,pos0.}.myclass::myclass(){
setGeometry(0,0,120,120). pixmap=new
QPixmap("/doc/test/a/tmp.png"). bitmap=new
QBitmap("/doc/test/a/mask.png").
setPaletteBackgroundPixmap(*pixmap). setMask(*bitmap).}void
```

```
myclass::mouseMoveEvent(QMouseEvent *e){  
if(e->state() == LeftButton) { QPoint newpos = e->globalPos().  
QPoint upleft = pos0 newpos - last. move(upleft). }}void  
myclass::mousePressEvent(QMouseEvent *e){ last =  
e->globalPos(). pos0 = e->globalPos() - e->pos().} int main(int  
argc, char *argv[]){ QApplication a(argc, argv). myclass w.  
a.setMainWidget(&w). w.show(). return a.exec().} 100Test 下  
载频道开通，各类考试题目直接下载。详细请访问  
www.100test.com
```