



## 十一月\_寒风

昵称: [平凡的编程者](#)  
 园龄: 6年2个月  
 粉丝: 12  
 关注: 0  
[+加关注](#)

2021年8月						
日	一	二	三	四	五	六
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

### 搜索


  



### 常用链接

[我的随笔](#)  
[我的评论](#)  
[我的参与](#)  
[最新评论](#)  
[我的标签](#)

### 我的标签

[C++\(8\)](#)  
[QT\(6\)](#)  
[OpenCV\(3\)](#)  
[视频播放\(2\)](#)  
[FFMPEG\(2\)](#)  
[contrib\(2\)](#)  
[bind\(2\)](#)  
[function\(2\)](#)  
[回调函数\(2\)](#)  
[Socket\(2\)](#)  
[更多](#)

### 随笔分类

[Android\(1\)](#)  
[C++\(9\)](#)  
[Delphi\(VCL\)\(1\)](#)  
[Python\(4\)](#)  
[QT\(7\)](#)

### 随笔档案

[2019年7月\(1\)](#)  
[2019年6月\(4\)](#)  
[2019年5月\(2\)](#)  
[2018年11月\(2\)](#)  
[2018年10月\(1\)](#)  
[2018年9月\(2\)](#)  
[2018年8月\(2\)](#)  
[2018年7月\(1\)](#)

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

随笔 - 20 文章 - 0 评论 - 15 阅读 - 10万

## QT+OPENCV实现录屏功能

本文使用QT+opencv来实现对指定窗体画面录制，并保存为avi文件。

### (1) 获取窗体界面

QScreen类有一个grabWindow函数，可以用来获取窗体的画面，这个函数使用很简单，就是传入窗体句柄和要截图的坐标。但是这个函数有一个缺陷，它是通过截取桌面画面的方式，而不是通过

窗体获取界面，所以当你的窗体被其他窗体遮挡时，就无法截取完整的窗体界面，如果你是要录制整个桌面画面，那用这个函数就可以了，下面的方法调用GDI函数来实现，即使窗体被遮挡时仍然能够获取到完整界面，但是窗体最小化时也一样无法获取。

```

/*
 * 函数功能: 获取窗体指定窗体图像
 * 参      数: hd:窗体句柄
 *             pm:保存获取到的图片
 *             x: 截取的起始x坐标,
 *             y: 截取的起始y坐标,
 *             w: 截取的宽度
 *             h: 截取的高度
 */
bool GetGDIBitmap(HWND hd, QPixmap &pm, int x, int y, int w, int h)
{
    if(hd==NULL)
        return false;
    HDC hDC;
    hDC=GetDCEX(hd, NULL, DCX_PARENTCLIP );
    HDC hMemDC;           //内存缓冲设备环境
    HBITMAP hbmMem, hbmOld; //内存缓冲设备环境中的位图
    RECT rc;
    rc.left=x;
    rc.top=y;
    rc.right=x+w;
    rc.bottom=y+h;
    //判断边境值
    RECT clientrc;
    ::GetClientRect(hd, &clientrc);

    int xc =0;
    int cx =0;
    int cy =0;

    if(rc.bottom>clientrc.bottom || rc.bottom<0)
        rc.bottom=clientrc.bottom;

    if(rc.right>clientrc.right || rc.right<0)
        rc.right=clientrc.right;

    // 24位图的BITMAPINFO
    BITMAPINFO *pBITMAPINFO = (BITMAPINFO*)malloc(sizeof(BITMAPINFOHEADER));
    memset(pBITMAPINFO, 0, sizeof(BITMAPINFOHEADER));
    BITMAPINFOHEADER *pInfo_Header = (BITMAPINFOHEADER *)pBITMAPINFO;
  
```

2018年6月(2)

2017年8月(2)

2017年5月(1)

## 文章分类

Delphi(FMX)(1)

## 最新评论

1. Re:QT实现支持加密的Sqlite数据库引擎

我按照步骤构建第一个项目时显示“qt无法找到执行档,请指定一个”

--CoyZheng

2. Re:QT实现支持加密的Sqlite数据库引擎

您好,我下载了您的工程运行测试程序后一直提示driver load fail,我已经把plugin文件夹放到相应的地方了

--CoyZheng

3. Re:Python监听键盘和鼠标事件

为什么我运行后没有反应?没有报错但是无任何反应

--一缕清风戏微尘

4. Re:Python窗体操作函数

```
PS C:\Users\lzg> & C:/Users/lzg/AppData/Local/Programs/Python/Python37/python.exe e:/python_study/Fo...
```

--尘土飞扬118

5. Re:QT实现支持加密的Sqlite数据库引擎

您好,我按照您的方法尝试了后能够在release下进行构建,但是没有生产dll,您遇到过这个问题吗?麻烦指导一下。谢谢。

--奋斗的小黄人

## 阅读排行榜

1. Python键盘按键模拟(17222)
2. Python监听键盘和鼠标事件(14121)
3. QT+FFMPEG实现视频播放(13159)
4. 开源OCR识别库-Tesseract介绍(1723)
5. QT源码分析: QObject(5958)

## 评论排行榜

1. QT+FFMPEG+SDL2.0实现视频播放(3)
2. QT实现支持加密的Sqlite数据库引擎(3)
3. QT5.10+MinGW+OpenCV3.4.2编译(2)
4. Android Tcp操作(2)
5. Python窗体操作函数(1)

## 推荐排行榜

1. QT源码分析: QObject(4)
2. Python键盘按键模拟(1)
3. QT源码分析: QTcpServer(1)
4. QT5.10+MinGW+OpenCV3.4.2

```
pInfo_Header->biSize = sizeof(BITMAPINFOHEADER);
pInfo_Header->biWidth = rc.right - rc.left;
pInfo_Header->biHeight = (rc.bottom - rc.top);
pInfo_Header->biPlanes = 1;
pInfo_Header->biBitCount = 24;
pInfo_Header->biCompression = BI_RGB;

hMemDC=CreateCompatibleDC(hDC); //创建内存兼容设备环境
//创建内存兼容位图
hbmMem=CreateCompatibleBitmap(hDC,pInfo_Header->biWidth,pInfo_Header->biHeight);

hbmOld=(HBITMAP) SelectObject(hMemDC,hbmMem);
//将内存设备环境中的内容绘制到物理设备环境 hDC
BitBlt(hMemDC,0,0,pInfo_Header->biWidth,pInfo_Header->biHeight,hDC,cx+rc.left,rc.top,rc.right,rc.bottom,SRCCOPY);
HBITMAP hBitmap=(HBITMAP) SelectObject(hMemDC,hbmOld);

// 获得数据buf
DWORD bufSize=(pInfo_Header->biWidth * 3 + 3) / 4 * 4 * pInfo_Header->biHeight;
BYTE * pBuffer = new BYTE[bufSize];

int aHeight=pInfo_Header->biHeight;

if(!::GetDIBits(hMemDC, hBitmap, 0, aHeight, pBuffer,pBITMAPINFO, DIB_RGB_COLORS))
{
    return false;
}

bool bret=BitmapToPixmap(hBitmap,pm);

ReleaseDC(hd,hDC);
//释放资源
DeleteObject(hbmMem);
DeleteObject(hbmOld);
DeleteDC(hMemDC);
free(pBITMAPINFO);
::DeleteObject(hBitmap);
delete [] pBuffer;
return bret;
}

/*
 * 函数功能: 将bitmap转为QPixmap
 */
bool BitmapToPixmap(HBITMAP hBitmap, QPixmap &pm)
{
    HDC hDC;
    //设备描述表
    int iBits;
    //当前显示分辨率下每个像素所占字节数
    WORD wBitCount;
    //位图中每个像素所占字节数
    //定义调色板大小, 位图中像素字节大小, 位图文件大小, 写入文件字节数
    DWORD dwPaletteSize=0,dwBmBitsSize,dwDIBSize;
    BITMAP Bitmap;
    //位图属性结构
    BITMAPFILEHEADER bmfHdr;
    //位图文件头结构
    BITMAPINFOHEADER bi;
    //位图信息头结构
    LPBITMAPINFOHEADER lpbi;
    //指向位图信息头结构
    HANDLE hDib, hPal;
    HPALETTE hOldPal=NULL;
    //定义文件, 分配内存句柄, 调色板句柄

    //计算位图文件每个像素所占字节数
    hDC = CreateDC(L"DISPLAY",NULL,NULL,NULL);
    iBits = GetDeviceCaps(hDC, BITSPIXEL) * GetDeviceCaps(hDC, PLANES);
    DeleteDC(hDC);
    if (iBits <= 1)
        wBitCount = 1;
```

编译(1)

5. VS2013+OpenCV3.4.2编译(1)

```
else if (iBits <= 4)
    wBitCount = 4;
else if (iBits <= 8)
    wBitCount = 8;
else if (iBits <= 24)
    wBitCount = 24;
else
    wBitCount = 24;
//计算调色板大小
if (wBitCount <= 8)
    dwPaletteSize=(1<<wBitCount)*sizeof(RGBQUAD);

//设置位图信息头结构
GetObject(hBitmap, sizeof(BITMAP), (LPSTR)&Bitmap);
bi.biSize = sizeof(BITMAPINFOHEADER);
bi.biWidth = Bitmap.bmWidth;
bi.biHeight = Bitmap.bmHeight;
bi.biPlanes = 1;
bi.biBitCount = wBitCount;
bi.biCompression = BI_RGB;
bi.biSizeImage = 0;
bi.biXPelsPerMeter = 0;
bi.biYPelsPerMeter = 0;
bi.biClrUsed = 0;
bi.biClrImportant = 0;

dwBmBitsSize = ((Bitmap.bmWidth*wBitCount+31)/32)*4*Bitmap.bmHeight;
//为位图内容分配内存
hDib = GlobalAlloc(GHND,dwBmBitsSize+dwPaletteSize+sizeof(BITMAPINFOHEADI
lpbi = (LPBITMAPINFOHEADER)GlobalLock(hDib);
*lpbi = bi;
// 处理调色板
hPal = GetStockObject(DEFAULT_PALETTE);
if (hPal)
{
    hDC = ::GetDC(NULL);
    hOldPal=SelectPalette(hDC, (HPALETTE)hPal, FALSE);
    RealizePalette(hDC);
}
// 获取该调色板下新的像素值
GetDIBits(hDC,hBitmap,0,(UINT)Bitmap.bmHeight,(LPSTR)lpbi+sizeof(BITMAPINI
//恢复调色板
if (hOldPal)
{
    SelectPalette(hDC, hOldPal, TRUE);
    RealizePalette(hDC);
    ::ReleaseDC(NULL, hDC);
}
// 设置位图文件头
bmfHdr.bfType = 0x4D42; // "BM"
dwDIBSize=sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER)+dwPaletteSize+
bmfHdr.bfSize = dwDIBSize;
bmfHdr.bfReserved1 = 0;
bmfHdr.bfReserved2 = 0;
bmfHdr.bfOffBits = (DWORD)sizeof(BITMAPFILEHEADER)+(DWORD)sizeof(BITMAPINI

std::vector<uchar>buffer;
uchar *p=(uchar*)&bmfHdr;
// 写入位图文件头
buffer.insert(buffer.end(),p,p+sizeof(BITMAPFILEHEADER));
// 写入位图文件其余内容
p=(uchar*)lpbi;
buffer.insert(buffer.end(),p,p+sizeof(BITMAPINFOHEADER)+dwPaletteSize+dwBn
//清除
GlobalUnlock(hDib);
GlobalFree(hDib);
pm=QPixmap::fromImage(QImage::fromData(buffer.data(),buffer.size()));
return true;
}

```



## (2) 录制画面

```
bool g_needstop =false;
void Record()
{
    RECT rect;
    //获取窗体位置大小
    GetWindowRect(hd,&rect);
    cv::Size frameSize;
    frameSize.width=rect.right-rect.left;
    frameSize.height=rect.bottom-rect.top;
    cv::VideoWriter VideoWriter;
    if(!VideoWriter.open("d:\\1.avi",CV_FOURCC('M', 'J', 'P', 'G'),40,frameSi
        return;
    while(!g_needstop)
    {
        QPixmap pm;
        GetGDIBitmap(hd,pm,0,0,frameSize.width,frameSize.height);
        VideoWriter.write(ImageToMat(pm.toImage()));
    }
    VideoWriter.release();
}

Mat ImageToMat(QImage img,QString imgFormat)
{
    if(img.isNull())
        return Mat();
    QByteArray ba;
    QBuffer buffer(&ba);
    buffer.open(QIODevice::WriteOnly);
    img.save(&buffer,imgFormat.toLatin1().data());
    _InputArray arrSrc(ba.data(), ba.size());
    Mat mat = cv::imdecode(arrSrc, CV_LOAD_IMAGE_COLOR);
    return mat;
}
```

## (3)播放视频

```
void Play()
{
    cv::VideoCapture Capture;
    if(!Capture.open("d:\\1.avi"))
        return;
    Mat frame;
    //逐帧读取画面
    while(Capture.read(frame))
    {
        //转成QImage格式用于显示
        QImage img = MatToImage(frame);
        emit Frame(img);
        QThread::msleep(40);
    }
    Capture.release();
    emit PlayFinsh();
}

QImage MatToImage(Mat mat)
{
    if(mat.type() == CV_8UC1)
    {
        QImage image(mat.cols, mat.rows, QImage::Format_Indexed8);
        // Set the color table (used to translate colour indexes to qRgb values)
        image.setColorCount(256);
        for(int i = 0; i < 256; i++)
        {
            image.setColor(i, qRgb(i, i, i));
        }
    }
}
```

```
    }
    // Copy input Mat
    uchar *pSrc = mat.data;
    for(int row = 0; row < mat.rows; row++)
    {
        uchar *pDest = image.scanLine(row);
        memcpy(pDest, pSrc, mat.cols);
        pSrc += mat.step;
    }
    return image;
}
// 8-bits unsigned, NO. OF CHANNELS = 3
else if(mat.type() == CV_8UC3)
{
    // Copy input Mat
    const uchar *pSrc = (const uchar*)mat.data;
    // Create QImage with same dimensions as input Mat
    QImage image(pSrc, mat.cols, mat.rows, mat.step, QImage::Format_RGB888);
    return image.rgbSwapped();
}
else if(mat.type() == CV_8UC4)
{
    qDebug() << "CV_8UC4";
    // Copy input Mat
    const uchar *pSrc = (const uchar*)mat.data;
    // Create QImage with same dimensions as input Mat
    QImage image(pSrc, mat.cols, mat.rows, mat.step, QImage::Format_ARGB32);
    return image.copy();
}
else
{
    qDebug() << "ERROR: Mat could not be converted to QImage.";
    return QImage();
}
}
```



分类: C++, QT

好文要顶

关注我

收藏该文



平凡的编程者

关注 - 0

粉丝 - 12

+加关注

0

0

« 上一篇: [Python窗体操作函数](#)

posted on 2019-07-18 17:23 [平凡的编程者](#) 阅读(3379) 评论(0) [编辑](#) [收藏](#) [举报](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

编辑 预览

B



支持 Markdown