

原 SDL2源代码分析8：视频显示总结

2014年11月18日 00:25:14 阅读数：10470

=====

SDL源代码分析系列文章列表：

[SDL2源代码分析1：初始化（SDL_Init\(\)）](#)

[SDL2源代码分析2：窗口（SDL_Window）](#)

[SDL2源代码分析3：渲染器（SDL_Renderer）](#)

[SDL2源代码分析4：纹理（SDL_Texture）](#)

[SDL2源代码分析5：更新纹理（SDL_UpdateTexture\(\)）](#)

[SDL2源代码分析6：复制到渲染器（SDL_RenderCopy\(\)）](#)

[SDL2源代码分析7：显示（SDL_RenderPresent\(\)）](#)

[SDL2源代码分析8：视频显示总结](#)

=====

本文简单总结一下SDL显示视频的源代码。

SDL显示视频的结构体

SDL显示视频涉及到下列结构体：

SDL_Window：代表了窗口

SDL_Renderer：代表了渲染器

SDL_Texture：代表了纹理

SDL_Rect：一个矩形框，用于确定纹理显示的位置。

上述几个结构体之间的关系如下图所示。

PS：该图源自于文章《[最简单的基于FFmpeg+SDL的视频播放器 ver2（采用SDL2.0）](#)》

□

由图可见，YUV/RGB像素数据首先加载至SDL_Texture，然后通过SDL_Render渲染至SDL_Window。其中SDL_Rect可以指定显示的位置。

SDL显示视频的流程

SDL显示视频的流程如下图所示。

□

更清晰的图片链接（右键保存）：<http://my.csdn.net/leixiaohua1020/album/detail/1795751>

从图中可以看出，整体的流程可以概括为如下步骤：

1. 初始化：SDL_Init()
2. 创建SDL_Window：SDL_CreateWindow()
3. 创建SDL_Render：SDL_CreateRenderer()
4. 创建SDL_Texture：SDL_CreateTexture()
5. 更新SDL_Texture：SDL_UpdateTexture()
6. 渲染SDL_Texture：SDL_RenderCopy()
7. 显示：SDL_RenderPresent()
8. 返回步骤4继续执行

上图中显示了SDL播放视频的时候API的调用流程。下文总结一下在不同的系统以及渲染技术下，这些SDL的API和系统底层API之间的调用关系。

SDL-Windows-Direct3D

SDL在Windows系统下，使用Direct3D渲染视频的时候的函数调用关系如下图所示。

PS：白色背景函数为SDL的API；蓝色背景的函数为Win32的API；紫色背景的函数Direct3D的API。

□

更清晰的图片链接（右键保存）：<http://my.csdn.net/leixiaohua1020/album/detail/1795753>

从图中可以看出，SDL在Windows下使用Direct3D渲染视频的时候。函数之间的调用关系如下所列：

SDL_CreateWindow()调用了如下Win32的API：

```
CreateWindow()
SetWindowText()
ShowWindow()
SetWindowPos()
```

SDL_CreateRenderer()调用了如下Direc3D的API：

```
Direct3DCreate9()
IDirect3D9_GetDeviceCaps()
IDirect3D9_CreateDevice()
IDirect3DDevice9_SetFVF()
IDirect3DDevice9_SetRenderState()
IDirect3DDevice9_SetTextureStageState()
IDirect3DDevice9_SetTransform()
IDirect3DDevice9_CreatePixelShader()
```

SDL_CreateTexture()调用了如下Direc3D的API：

```
IDirect3DDevice9_CreateTexture()
```

SDL_UpdateTexture()调用了如下Direc3D的API：

```
IDirect3DTexture9_LockRect()
memcpy()：这个不算D3D的，用于拷贝像素数据。
IDirect3DTexture9_UnlockRect()
```

SDL_RenderCopy()调用了如下Direc3D的API：

```
IDirect3DDevice9_BeginScene()
IDirect3DDevice9_SetRenderState()
IDirect3DDevice9_SetSamplerState()
IDirect3DDevice9_SetTexture()
IDirect3DDevice9_SetPixelShader()
IDirect3DDevice9_DrawPrimitiveUP()
```

SDL_RenderPresent()调用了如下Direc3D的API：

```
IDirect3DDevice9_EndScene()
IDirect3DDevice9_Present()
```

SDL-Windows-OpenGL

SDL在Windows系统下，使用OpenGL渲染视频的时候的函数调用关系如下图所示。

PS：白色背景函数为SDL的API；蓝色背景的函数为Win32的API；紫色背景的函数OpenGL的API。

□

更清晰的图片链接（右键保存）：<http://my.csdn.net/leixiaohua1020/album/detail/1795755>

从图中可以看出，SDL在Windows下使用OpenGL渲染视频的时候。函数之间的调用关系如下所列：

SDL_CreateWindow()调用了如下Win32的API：

```
CreateWindow()
SetWindowText()
ShowWindow()
SetWindowPos()
```

SDL_CreateRenderer()调用了如下OpenGL的API：

```
glCreateProgramObject()
glCreateShaderObject()
glShaderSource()
glCompileShader()
GetObjectParameteriv()
AttachObject()
LinkProgram()
```

UseProgramObject()

SDL_CreateTexture()调用了如下OpenGL的API：

glGenTextures()
glBindTexture()
glTexParameteri()
glTexImage2D()

SDL_UpdateTexture()调用了如下OpenGL的API：

glBindTexture()
glTexSubImage2D()

SDL_RenderCopy()调用了如下OpenGL的API：

glActiveTexture()
glBindTexture()

SDL_RenderPresent()调用了如下OpenGL的API：

SwapBuffers()

SDL-Windows-Software

SDL在Windows系统下，使用Software渲染视频的时候的函数调用关系如下图所示。

PS1：白色背景函数为SDL的API；蓝色背景的函数为Win32的API。

PS2：Software渲染目前还没有透彻分析。

□

更清晰的图片链接（右键保存）：<http://my.csdn.net/leixiaohua1020/album/detail/1795757>

从图中可以看出，SDL在Windows下使用Software渲染视频的时候。函数之间的调用关系如下所列：

SDL_CreateWindow()调用了如下Win32的API：

CreateWindow()
SetWindowText()
ShowWindow()
SetWindowPos()

SDL_CreateRenderer()调用了如下Win32的API：

CreateCompatibleBitmap()
GetDIBits()
CreateCompatibleDC()
CreateDIBSection()
SelectObject()

SDL_UpdateTexture()调用了memcpy()填充像素数据。

SDL_RenderPresent()调用了如下Win32的API：

BitBlt()

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/41207787>

文章标签：[SDL](#) [OpenGL](#) [Direct3D](#) [GDI](#) [源代码](#)

个人分类：[SDL](#)

所属专栏：[开源多媒体项目源代码分析](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com