



# 计算机图形学小白入门

——从0开始实现OpenGL

GDI绘图环境构建



授课：赵新政  
资深三维工程师

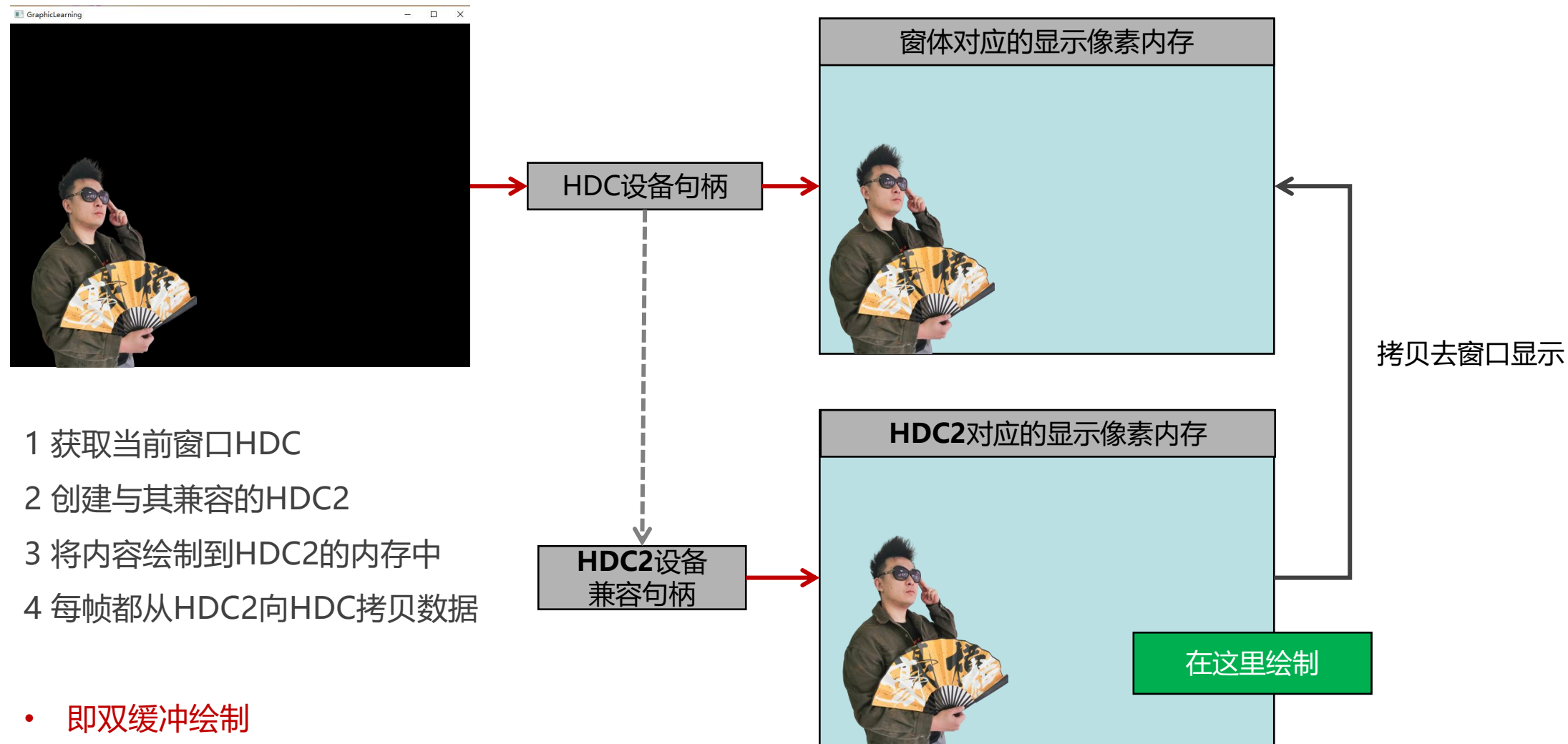
专注3D图形学技术  
教育品牌

## 什么是Windows GDI

- GDI是图形设备接口（**GDI: Graphics Device Interface**）的缩写，是Windows的子系统，它负责在视讯显示器和打印机上显示图形
- HDC（Handle of **Device Context**）：即GDI体系中的**设备描述上下文**；
- ——在Windows操作系统中，**显示资源**属于所有APP公用资源，公用就需要虚拟化成多个虚拟设备
- ——每个窗体运行在其**虚拟显示设备**上，其窗体句柄就对应了一个正在使用的设备DC
- ——每个HDC句柄背后都记录了当前窗体显示对应的内存/尺寸等信息

如果想在窗口上绘制物体，就需要向HDC所指代的设备内存上输出颜色数据即可

## HDC绘图原理图示



## HDC获取与创建

- 可以通过GetDC获取当前窗口对应的HDC虚拟绘图设备；
- 可以通过**CreateCompatibleDC**创建一个与hDC相兼容的hDC2，那么就可以在hDC2上绘制数据，拷贝到hDC进行显示

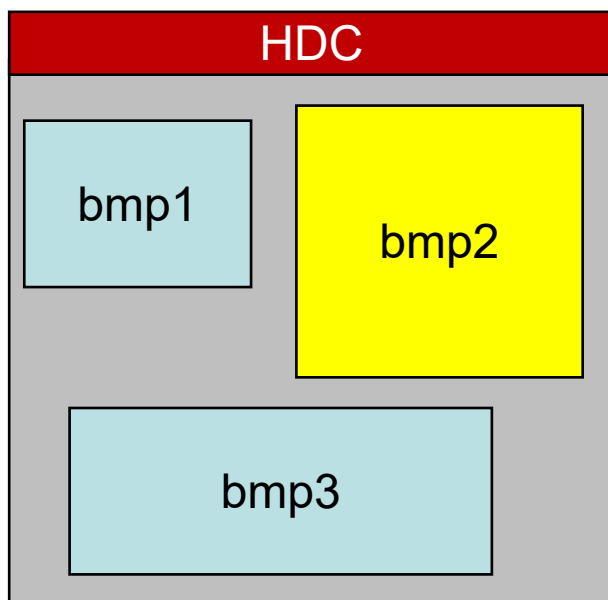
```
#获取当前窗体HDC  
hDC = GetDC(hwnd);  
  
#创建与当前窗体兼容的HDC2（内存格式/分辨率等）  
hDC2= CreateCompatibleDC(hDC );  
  
#把hDC2对应内存的数据拷贝到hDC  
BitBlt(hDC , xdst, ydst, width, height, hDC2, xsrc, ysrc, SRCCOPY);
```

## HDC与绘图

HDC虚拟设备资源，一般情况下指的就是绘图设备内存；我们想向HDC上输出颜色数据，就需要对其进行设置

### 位图

在Windows GDI中，位图对象句柄是**HBITMAP**；每个HDC都可以使用自己的虚拟资源，分配多张位图对象，如下图



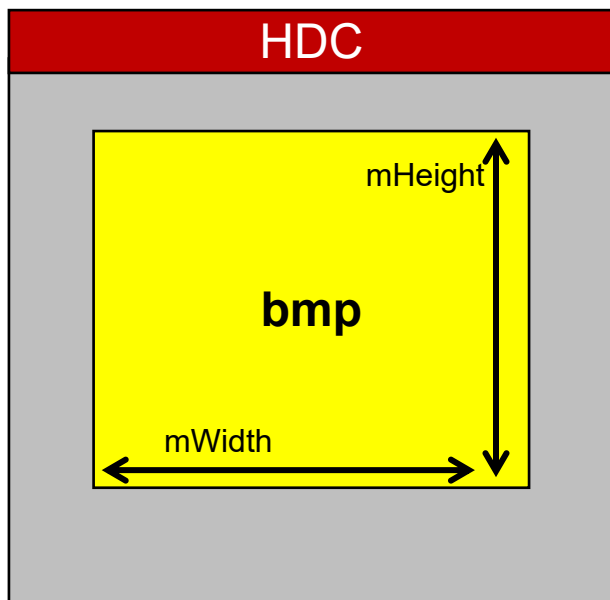
### 操作对象选择

对于每个HDC，都会有一些列的API对其操作（拷出内存/拷入数据等），那么我们可以选择一个位图，作为当前HDC的激活位图，所有对本HDC的数据操作，都是在操作**本激活位图**对应的内存区域

```
#激活bmp2, 作为主位图对象  
SelectObject(hDC, bmp2);
```

## HBITMAP创建

创建位图如下所示，CreateDIBSection会在hDC2上为位图开辟空间，这块内存的指针会给到buffer



```
unsigned char * buffer = nullptr;
```

```
BITMAPINFO bmpInfo{};  
bmpInfo.bmiHeader.biSize = sizeof(BITMAPINFOHEADER);  
bmpInfo.bmiHeader.biWidth = mWidth;  
bmpInfo.bmiHeader.biHeight = mHeight;  
bmpInfo.bmiHeader.biPlanes = 1;  
bmpInfo.bmiHeader.biBitCount = 32;  
bmpInfo.bmiHeader.biCompression = BI_RGB; //实际上存储方式为bgra
```

//创建与hDC2兼容的位图, 在hDC2指代的设备上划拨了一块内存, buffer 指向它

```
HBITMAP bmp = CreateDIBSection(  
    hDC2,  
    &bmpInfo,  
    DIB_RGB_COLORS,  
    (void**)&buffer ,  
    0, 0);
```

//每个HDC虚拟设备都可以分配出来多个位图/画刷等资源，本操作是将bmp作为当前hDC2的操作对象，将来所有对hDC2的拷贝操作都是再拷贝bmp的数据

```
SelectObject(hDC2, bmp);
```

考察前文创建的所有对象关系

