(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104166552 B
(45)授权公告日 2017.10.13

(21)申请号 201410400766.9

(22)申请日 2014.08.15

(65)同一申请的已公布的文献号
申请公布号 CN 104166552 A

(43)申请公布日 2014.11.26

(73)专利权人 成都天奥信息科技有限公司
地址 610000 四川省成都市高新西区新业路天奥科技产业园

(72)发明人 刘宇 卢新平

(74)专利代理机构 成都行之专利代理事务所(普通合伙) 51220
代理人 梁田

(51)Int.Cl.
G06F 9/44(2006.01)

(56)对比文件
CN 103870251 A,2014.06.18,
CN 102508648 A,2012.06.20,
佚名.Android库so文件及skia函数的调用.《http://mobile.51cto.com/android-266606.htm》.2011,第1-5页.
Alex.MFC中的GDI绘图.《http://www.cnblogs.com/weiqubo/archive/2009/12/24/1930029.html》.2009,第1-10页.
GIS_.Flex中ByteArray与BitmapData互相转换实现图片的二进制保存与复原.《http://blog.csdn.net/gis_/article/details/6689013》.2011,第1-3页.
周祖洋等.Windows下的位图在VxWorks中大字显示的实现.《应用科技》.2005,第32卷(第7期),第38-41页.
huazaicola.Vxworks中封装函数库的方法及库文件使用.《Vxworks中封装函数库的方法及库文件使用》.2011,第1-2页.

审查员 吴阳

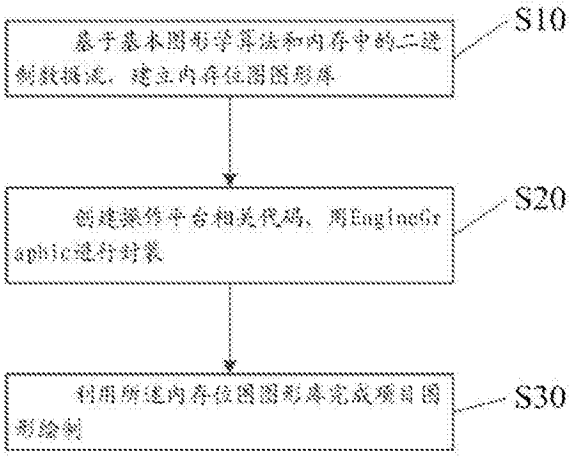
权利要求书1页 说明书6页 附图1页

(54)发明名称

一种可移植内存位图图形库使用方法

(57)摘要

本发明公开了一种可移植内存位图图形库使用方法,所述包括方法包括:首先基于基本图形学算法和内存中的二进制数据流,建立内存位图图形库,然后创建操作平台相关代码,用EngineGraphic进行封装,最后利用所述内存位图图形库完成项目图形绘制,实现了图形绘制效率比一般平台的效率更高,需要图形绘制的大型项中使用这种方法,可使得多达十万甚至百万行代码的项目转移到其他平台只需要修改较少的代码即可的技术效果。



CN 104166552 B

1.一种可移植内存位图图形库使用方法,其特征在于,所述方法包括:

基于基本图形学算法和内存中的二进制数据流,建立内存位图图形库;

创建操作平台相关代码,用EngineGraphic进行封装;

利用所述内存位图图形库完成项目图形绘制;

当在Windows操作平台下时,所述利用所述内存位图图形库完成项目图形绘制具体包括:

使用编译器把图形库源代码编译动态库dll文件;

把编译后的所述dll文件和后缀名h的头文件作为必要的文件放入项目;

在所述项目中使用头文件中的函数;

当在Android操作平台下时,所述利用所述内存位图图形库完成项目图形绘制具体包括:

根据源代码建立JNI相应接口代码;

根据所要运行的android版本,编译android代码,获取所属的后缀名为so动态库;

添加对应的java接口代码;

在项目中使用所述java接口代码;

当在VxWorks操作平台下时,所述利用所述内存位图图形库完成项目图形绘制具体包括:

使用编译器把图形库源代码编译动态库a文件;

把编译后的所述a文件和后缀名h的头文件作为必要的文件放入项目;

在所述项目中使用头文件中的函数;

所述创建操作平台相关代码,用EngineGraphic进行封装具体包括:

在项目中创建类CEngineGraphic;

使用平台图形库中的函数封装平台相关代码到CEngineGraphic中。

2.根据权利要求1所述的方法,其特征在于,所述操作平台包括但不限于:Windows操作平台、Android操作平台、VxWorks操作平台。

3.根据权利要求1所述的方法,其特征在于,所述基本图形学算法包括但不限于:直线算法、画圆算法、填充算法。

4.根据权利要求1所述的方法,其特征在于,所述内存位图图形库的功能包括但不限于:多边形透明拷贝、模式多边形透明拷贝、矩形绘制、三角形绘制、透明拷贝位图、矩形填充、十字叉多边形填充。

一种可移植内存位图图形库使用方法

技术领域

[0001] 本发明涉及计算机图形学研究领域，尤其涉及一种可移植内存位图图形库使用方法。

背景技术

[0002] 图形库在计算机绘图中起着重要的作用，随着可以的进步和发展，图形库的应用日益广泛，如果项目需要用到图形绘制相关的功能，那么就需要用到平台相关图形库，如Windows下的MFC库，WxWidget库、Android平台下的skia库、VxWorks平台下的windml图形库。而当项目需要同时开发几个平台下的，每个平台下都需要维护各个版本的代码，且每个平台下图形库的调用都大不一样，加大了项目的维护及开发。

[0003] 并且在windows下开发复杂度高的且处理大量图形的项目时，windows自带的图形库就显得有点力不从心了，绘制速度跟不上。

[0004] 综上所述，本申请发明人在实现本申请实施例中发明技术方案的过程中，发现上述技术至少存在如下技术问题：

[0005] 在现有技术中，由于当项目需要同时开发几个平台下的，每个平台下都需要维护各个版本的代码，且每个平台下图形库的调用都大不一样，加大了项目的维护及开发，且在windows下开发复杂度高的且处理大量图形的项目时，windows自带的图形库就显得有点力不从心了，绘制速度跟不上，所以，现有的平台下的图形库存在要维护各个版本的代码，且每个平台下图形库的调用都大不一样，加大了项目的维护及开发，图形库绘制速度较慢的技术问题。

发明内容

[0006] 本发明提供了一种可移植内存位图图形库使用方法，解决了现有的平台下的图形库存在要维护各个版本的代码，且每个平台下图形库的调用都大不一样，加大了项目的维护及开发，图形库绘制速度较慢的技术问题，实现了图形绘制效率比一般平台的效率更高，需要图形绘制的大型项中使用这种方法，可使得多达十万甚至百万行代码的项目转移到其他平台只需要修改区区几百行代码即可的技术效果。

[0007] 为解决上述技术问题，本申请实施例提供了一种可移植内存位图图形库使用方法，所述包括方法包括：

[0008] 首先，基于基本图形学算法和内存中的二进制数据流，建立内存位图图形库；

[0009] 然后，创建操作平台相关代码，用EngineGraphic进行封装；

[0010] 然后，利用所述内存位图图形库完成项目图形绘制。

[0011] 其中，所述操作平台包括但不限于：Windows操作平台、Android操作平台、VxWorks操作平台。

[0012] 其中，所述基本图形学算法包括但不限于：直线算法、画圆算法、填充算法。

[0013] 其中，所述内存位图图形库的功能包括但不限于：多边形透明拷贝、模式多边形透

明拷贝、矩形绘制、三角形绘制、透明拷贝位图、矩形填充、十字叉多边形填充。

[0014] 其中，当在Windows操作平台下时，所述利用所述内存位图图形库完成项目图形绘制具体包括：

[0015] 使用编译器把图形库源代码编译动态库dll文件；

[0016] 把编译后的所述dll文件和后缀名h的头文件作为必要的文件放入项目；

[0017] 在所述项目中使用头文件中的函数。

[0018] 其中，当在Android操作平台下时，所述利用所述内存位图图形库完成项目图形绘制具体包括：

[0019] 根据源代码建立JNI相应接口代码；

[0020] 根据所要运行的android版本，编译android代码，获取所属的后缀名为so动态库；

[0021] 添加对应的java接口代码；

[0022] 在项目中使用所述java接口代码。

[0023] 其中，当在VxWorks操作平台下时，所述利用所述内存位图图形库完成项目图形绘制具体包括：

[0024] 使用编译器把图形库源代码编译动态库a文件；

[0025] 把编译后的所述a文件和后缀名h的头文件作为必要的文件放入项目；

[0026] 在所述项目中使用头文件中的函数。

[0027] 其中，所述创建操作平台相关代码，用EngineGraphic进行封装具体包括：

[0028] 在项目中创建类CEngineGraphic；

[0029] 使用平台图形库中的函数封装平台相关代码到CEngineGraphic中。

[0030] 本申请实施例中提供的一个或多个技术方案，至少具有如下技术效果或优点：

[0031] 由于采用了首先基于基本图形学算法和内存中的二进制数据流，建立内存位图图形库，然后创建操作平台相关代码，用EngineGraphic进行封装，最后利用所述内存位图图形库完成项目图形绘制的技术方案，即项目只需要调用内存位图图形库，这样移植的时候就可以不需要修改图形相关代码了，而另外用一个EngineGraphic封装平台相关代码，接口都统一起来，使得内存位图图形库来调用，所以，有效解决了现有的平台下的图形库存在要维护各个版本的代码，且每个平台下图形库的调用都大不一样，加大了项目的维护及开发，图形库绘制速度较慢的技术问题，进而实现了图形绘制效率比一般平台的效率更高，需要图形绘制的大型项中使用这种方法，可使得多达十万甚至百万行代码的项目转移到其他平台只需要修改区区几百行代码即可的技术效果。

附图说明

[0032] 图1是本申请实施例一中可移植内存位图图形库使用方法的流程图；

[0033] 图2是本申请实施例一中可移植内存位图图形库在项目中的位置示意图。

具体实施方式

[0034] 本发明提供了一种可移植内存位图图形库使用方法，解决了现有的平台下的图形库存在要维护各个版本的代码，且每个平台下图形库的调用都大不一样，加大了项目的维护及开发，图形库绘制速度较慢的技术问题，实现了图形绘制效率比一般平台的效率更高，

需要图形绘制的大型项中使用这种方法，可使得多达十万甚至百万行代码的项目转移到其他平台只需要修改较少的代码即可的技术效果。

[0035] 申请实施中的技术方案为解决上述技术问题。总体思路如下：

[0036] 采用了首先基于基本图形学算法和内存中的二进制数据流，建立内存位图图形库，然后创建操作平台相关代码，用EngineGraphic进行封装，最后利用所述内存位图图形库完成项目图形绘制的技术方案，即项目只需要调用内存位图图形库，这样移植的时候就可以不需要修改图形相关代码了，而另外用一个EngineGraphic封装平台相关代码，接口都统一起来，使得内存位图图形库来调用，所以，有效解决了现有的平台下的图形库存在要维护各个版本的代码，且每个平台下图形库的调用都大不一样，加大了项目的维护及开发，图形库绘制速度较慢的技术问题，进而实现了图形绘制效率比一般平台的效率更高，需要图形绘制的大型项中使用这种方法，可使得多达十万甚至百万行代码的项目转移到其他平台只需要修改较少的代码即可的技术效果。

[0037] 为了更好的理解上述技术方案，下面将结合说明书附图以及具体的实施方式对上述技术方案进行详细的说明。

[0038] 实施例一：

[0039] 在实施例一中，提供了一种可移植内存位图图形库使用方法，请参考图1-图2，所述包括方法包括：

[0040] S10，基于基本图形学算法和内存中的二进制数据流，建立内存位图图形库；

[0041] S20，创建操作平台相关代码，用EngineGraphic进行封装；

[0042] S30，利用所述内存位图图形库完成项目图形绘制。

[0043] 其中，在本申请实施例中，所述操作平台包括但不限于：Windows操作平台、Android操作平台、VxWorks操作平台。

[0044] 其中，在本申请实施例中，所述基本图形学算法包括但不限于：直线算法、画圆算法、填充算法。

[0045] 其中，在本申请实施例中，所述内存位图图形库的功能包括但不限于：多边形透明拷贝、模式多边形透明拷贝、矩形绘制、三角形绘制、透明拷贝位图、矩形填充、十字叉多边形填充。

[0046] 其中，在本申请实施例中，当在Windows操作平台下时，所述利用所述内存位图图形库完成项目图形绘制具体包括：

[0047] 使用编译器把图形库源代码编译动态库dll文件；

[0048] 把编译后的所述dll文件和后缀名h的头文件作为必要的文件放入项目；

[0049] 在所述项目中使用头文件中的函数。

[0050] 其中，在本申请实施例中，当在Android操作平台下时，所述利用所述内存位图图形库完成项目图形绘制具体包括：

[0051] 根据源代码建立JNI相应接口代码；

[0052] 根据所要运行的android版本，编译android代码，获取所属的后缀名为so动态库；

[0053] 添加对应的java接口代码；

[0054] 在项目中使用所述java接口代码。

[0055] 其中，在本申请实施例中，当在VxWorks操作平台下时，所述利用所述内存位图图

形库完成项目图形绘制具体包括：

[0056] 使用编译器把图形库源代码编译动态库a文件；

[0057] 把编译后的所述a文件和后缀名h的头文件作为必要的文件放入项目；

[0058] 在所述项目中使用头文件中的函数。

[0059] 其中，在本申请实施例中，所述创建操作平台相关代码，用EngineGraphic进行封装具体包括：

[0060] 在项目中创建类CEngineGraphic；

[0061] 使用平台图形库中的函数封装平台相关代码到CEngineGraphic中。

[0062] 其中，在实际应用中，基于计算机图形学中的内容，里面包括了直线算法、画圆算法，填充算法等，有了这些基本算法后，即可衍生出其他许多图形算法包括虚线绘制、透明拷贝等诸多算法，以此来构成一个图形库。

[0063] 上述描述的诸多图形算法实际上操作的都是二进制数据流，因此效率非常高效，可以把二进制数据流看成一个width*height的屏幕。一个点占四个字节(A、R、G、B)，因此一张width*height的屏幕可以抽象看成是字节数为4*width*height的二进制数据流。至此所描述的都是平台无关的，只是通过抽象二进制数据流为一张位图，通过计算机图形学中的算法来操作位图以此达到图形渲染的效果。

[0064] 而通过绘制后的内存位图使用每个平台的图形库与实际屏幕相关联。几乎每个平台都有一个类似的函数SetBitmapBits，参数是字节流及字节数。如：

[0065] MFC中的CBitmap类有函数SetBitmapBits(unsigned char *pByte,int nBytes)；

[0066] Android中的源代码中的skia库中可以获得位图的地址，通过memcpy即可达到内存位图拷贝的目的:memcpy(SkBitmap->getPixels(), pByte, nByte)；

[0067] VxWorks中windml库的有函数uglBitmapWrite。

[0068] 请参考图2，图2 内存位图在项目中的位置，如图2所示：项目只需要调用内存位图图形库，这样移植的时候就可以不需要修改图形相关代码了。而另外用一个EngineGraphic封装平台相关代码，接口都统一起来，使得内存位图图形库来调用。

[0069] 因此使用了这种方法过后，图形绘制效率比一般平台的效率更高。需要图形绘制的大型项中使用这种方法，可使得多达十万甚至百万行代码的项目转移到其他平台只需要修改区区几百行代码即可。

[0070] 其中，在实际应用中，应用本申请实施例中的可移植内存位图图形库与传统的操作平台图形库效率比较，其中，与Android平台下效率比较，如表1所示：

表1

[0071]

	调用 Java 层 300×200 次画线	SurfaceView 中直接调用 JNI 层 300×200 次画线函数	View 中直接在 JNI 层画 300×200 条线	SurfaceView 中用内存位图库直接在 JNI 层画 300×200 次画线	View 中用内存位图库在 JNI 层 300×200 次画线
第一次绘制时间 毫秒	624	115	74	51	49
第二次绘制时间 毫秒	648	114	71	48	49
第三次绘制时间 毫秒	676	118	70	48	45
平均时间 毫秒	648.6666667	115.6666667	71.66666667	49.33333333	47.66666667

[0072] 从表1中可以看到使用了内存位图库在JNI层提高了一倍左右的渲染效率。

[0073] 在windows下MFC平台的效率比较，比较绘制直线以及圆的绘制，如表2所示：

[0074]

表2

	直接使用 MFC 图形库的直线函数，绘制直线 10000 次	使用内存位图图形绘制直线 10000 次	直接使用 MFC 图形库绘制圆 10000 次	使用内存位图图形绘制圆 10000 次
第一次绘制时间 毫秒	453	79	484	234
第二次绘制时间 毫秒	453	94	500	213
第三次绘制时间 毫秒	453	94	500	234
平均时间 毫秒	453	89	494.6666667	229

[0075] 其中，从表2中可以看到直线绘制的效率是5倍左右。而圆的绘制效率也是1倍。

[0076] 上述本申请实施例中的技术方案，至少具有如下的技术效果或优点：

[0077] 由于采用了首先基于基本图形学算法和内存中的二进制数据流，建立内存位图图形库，然后创建操作平台相关代码，用EngineGraphic进行封装，最后利用所述内存位图图形库完成项目图形绘制的技术方案，即项目只需要调用内存位图图形库，这样移植的时候就可以不需要修改图形相关代码了，而另外用一个EngineGraphic封装平台相关代码，接口都统一起来，使得内存位图图形库来调用，所以，有效解决了现有的平台下的图形库存在要维护各个版本的代码，且每个平台下图形库的调用都大不一样，加大了项目的维护及开发，图形库绘制速度较慢的技术问题，进而实现了图形绘制效率比一般平台的效率更高，需要图形绘制的大型项中使用这种方法，可使得多达十万甚至百万行代码的项目转移到其他平台只需要修改较少的代码即可的技术效果。

[0078] 尽管已描述了本发明的优选实施例，但本领域内的技术人员一旦得知了基本创造性概念，则可对这些实施例作出另外的变更和修改。所以，所附权利要求意欲解释为包括优

选实施例以及落入本发明范围的所有变更和修改。

[0079] 显然，本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样，倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内，则本发明也意图包含这些改动和变型在内。

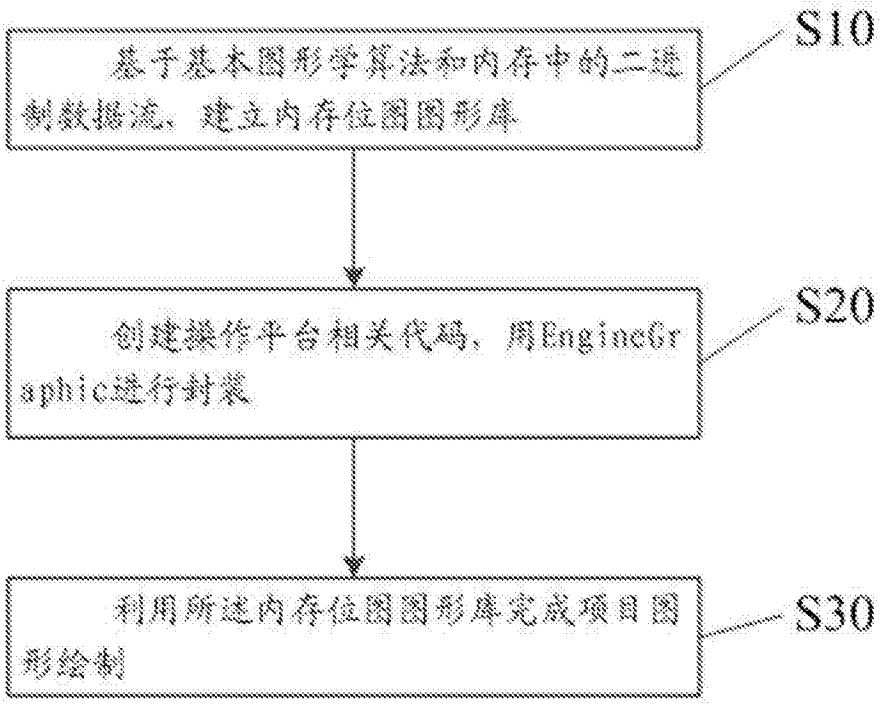

基于基本图形学算法和内存中的二进制数据流，建立内存位图图形库
S10
创建操作平台相关代码，用EngineGraphic进行封装
S20
利用所述内存位图图形库完成项目图形绘制
S30

图1


EngineGraphic
封装平台相关函数
SetBitmapBits
内存位图图形库
需要用到图形绘制的项目

图2