

目 录

1 概述	1
1.1 架构	1
1.2 文件组成	2
1.3 系统特点说明	3
2 功能说明	5
2.1 图形用户界面	5
2.2 动画组件	6
2.2.1 动画效果	6
2.2.2 运动曲线	6
2.3 界面编辑工具	7
3 示例工程	8
3.1 Demo1	8
3.2 Demo2	8
3.3 Demo3	9
3.4 Demo4	10
3.5 Demo5	10
4 API 参考	12
4.1 addAnimationData	12
5 API 参考	12
5.1 KDrawable::enableShow	12
5.2 KDrawable::SetBound	13
5.3 KDrawable::GetBound	13
5.4 KDrawable::SetPaint	14
5.5 KDrawable::GetPaint	14
5.6 KDrawable::SetClip	15
5.7 KDrawable::GetClip	15
5.8 KDrawable::SetClipEnable	15
5.9 KDrawable::getType	16

5.10 KColorDrawable::KColorDrawable	17
5.11 KColorDrawable::setColor	17
5.12 KColorDrawable::getColor	17
5.13 KImageDrawable::KImageDrawable	18
5.14 KImageDrawable::KImageDrawable(const kn_string& strFile)	18
5.15 KImageDrawable::KImageDrawable(IRESurface* pSurface)	19
5.16 KImageDrawable::CreateFromFile	19
5.17 KTextDrawable::KTextDrawable	20
5.18 KTextDrawable::SetText	21
5.19 KTextDrawable::GetText	21
5.20 KTextDrawable::setColor	21
5.21 KTextDrawable::getColor	22
5.22 KTextDrawable::SetFontSize	22
5.23 KTextDrawable::GetFontSize	23
5.24 KTextDrawable::setFontName	23
5.25 KTextDrawable::GetFontName	24
5.26 KTextDrawable::setTextAlign	24
5.27 KTextDrawable::getTextAlign	24
5.28 KTextDrawable::setBold	25
5.29 KTextDrawable::isBold	25
5.30 KTextDrawable::setOutBoundDot	26
5.31 KShapeDrawable::KShapeDrawable	27
5.32 KShapeDrawable::SetShape	28
5.33 KSVGDrawable::KSVGDrawable	28
5.34 KSVGDrawable::KSVGDrawable(kn_string svgFilePath)	29
5.35 KSVGDrawable::KSVGDrawable(kn_string svgFilePath, RERect& rect)	29
5.36 KSVGDrawable::addLine	30
5.37 KSVGDrawable::addPolygon	31
5.38 KSVGDrawable::addPath	31

5.39 KSVGDrawable::addSVGShape	33
5.40 KView::Release	33
5.41 KView::Create	34
5.42 KView::Create	34
5.43 KView::RefreshSurface	35
5.44 KView::enableCache	35
5.45 KView::GetScreen	36
5.46 KView::SetScreen	36
5.47 KView::getType	36
5.48 KView::addDrawable	37
5.49 KView::enableMessage	37
5.50 KView::IsShown	38
5.51 KView::SetShow	38
5.52 KView::SetOpacity	38
5.53 KView::GetOpacity	39
5.54 KView::GetPosition	39
5.55 KView::GetPosition	40
5.56 KView::GetScreen	40
5.57 KView::GetScreenPosition	41
5.58 KView::SetPosition	41
5.59 KView::Invalidate	41
5.60 KView::InvalidateView	42
5.61 KView::InvalidateForParent	42
5.62 KView::clearInvalidRect	43
5.63 KView::GetRect	43
5.64 KView::SetRect	43
5.65 KView::GetBoundRect	44
5.66 KView::SetBoundRect	44
5.67 KView::Translate	45

5.68 KView::Scale.....	45
5.69 KView::Rotate	46
5.70 KView::Skew.....	46
5.71 KView::SetTranslate	47
5.72 KView::SetScale	47
5.73 KView::SetRotate.....	48
5.74 KView::SetSkew	48
5.75 KView::OnMove	49
5.76 KView::OnDown.....	49
5.77 KView::OnUp	50
5.78 KView::OnDClick	50
5.79 KView::OnRDown	50
5.80 KView::OnRUp.....	51
5.81 KView::OnRDClick.....	51
5.82 KView::onDownDirect	52
5.83 KView::onMoveDirect.....	52
5.84 KView::onUpDirect.....	53
5.85 KView::onWheelDirect	53
5.86 KView::OnWheel.....	54
5.87 KView::OnUserMsg.....	54
5.88 KView::IsMousePicked	55
5.89 KView::getRotateAngle	55
5.90 KView::getRotatePoint	55
5.91 KStaticView::SetFontSize.....	56
5.92 KStaticView::setTextAlign.....	56
5.93 KStaticView::SetTextBound.....	57
5.94 KStaticView::GetTextBound	57
6 数据结构.....	58
6.1 KDrawable_PTR.....	58

6.2 KColorDrawable_PTR.....	58
6.3 KImageDrawable_PTR	58
6.4 KTextDrawable_PTR	59
6.5 KShapeDrawable_PTR.....	59
6.6 VEC_DRAWABLE	59
6.7 LoopType	59
6.8 KVIEWTYPE.....	60
6.9 SkPoint.....	60
6.10 Align	61
6.11 ViewState.....	61

自然人机界面开发平台用户手册

1 概述

光庭自然人机界面开发平台是针对实现自然人机交互界面（**Natural User Interface**）开发的下一代 HMI 引擎，一个供开发人员开发高级自然图形界面应用的中间件开发包。

人机界面技术经历了数个发展阶段，从最初的命令型界面(CLI)，到图形化用户界面（GUI）。伴随着计算机处理能力的增强、体积的减小和成本的降低，人机界面技术有了更多的设备形态（智能手机、平板、车载系统等），具备了除鼠标和键盘之外更多的输入方式（触摸屏、语音识别、动作识别等），而随之也出现了自然用户界面 NUI（**Natural User Interface**）。

NUI 代表着更直观、更具操作性、更人性化的自然人机交互方式，具备一些基本特点：以内容为中心的界面设计、动态的界面元素构造出的无缝交互体验、更自然的人机交互方式（如比鼠标更直观的多点触摸屏）



示例：使用 NUI 引擎开发的字体浏览程序

基于 GUI 框架的传统 HMI 引擎并不能满足 NUI 界面的设计需求，光庭自主开发自然人机界面开发平台，拥有高效的界面渲染效率及快速的消息响应机制，可以便捷实现 NUI 方式的界面交互

1.1 架构

NUI 引擎由如下几个核心组件构成

组件	说明
KScreen	屏幕组件，UI 的显示层，一个应用只能创建一个，所有可见元素都是在 Screen 上显示
Drawable	Drawable 类代表屏幕可绘制的对象，图片，文字，矢量图形等都属于 Drawable，通过不同的派生类定义这些对象。View 最终的呈现，都是依靠 drawable 对象呈现
View	各种可见元素，View 有各种派生类，用于各种行为和事件的处理及输出，对于 NUI 开发，针对特定的设计需要开发特定的 view
ViewGroup	View 的容器，可以容纳多个 View，进行组合，也可容纳 ViewGroup 本身
NUIAnimation	动画类：对于 NUI 来说，界面是动态构成的，相对于 GUI，有更多的动态效果呈现。动画类能够组织界面的 View，通过关键帧及指定物理运动曲线的方式使界面 View 运动起来
CAnimationThread	动画线程管理类：动画类需要定时触发渲染来实现动画绘制，NUI 引擎中这个过程是通过多线程完成的，而 CAnimationThread 就是用于绑定动画的动画线程类

1.2 文件组成

自然人机界面开发平台提供开发文档，静态链接库，开发包 SDK 头文件及使用示例，如下表：

静态 Lib 库	HMIEngine.lib	界面引擎库
	RenderingEngine.lib	渲染引擎库
	TINYXML.lib	xml 解析库
开发文档	自然人机界面开发平台使用说明.doc API 手册	
示例工程	HMIDemo HMIDemoMove HMIMessageDemo HMIPageDemo	
头文件清单	AnimationThread.h	DrawLooper.h

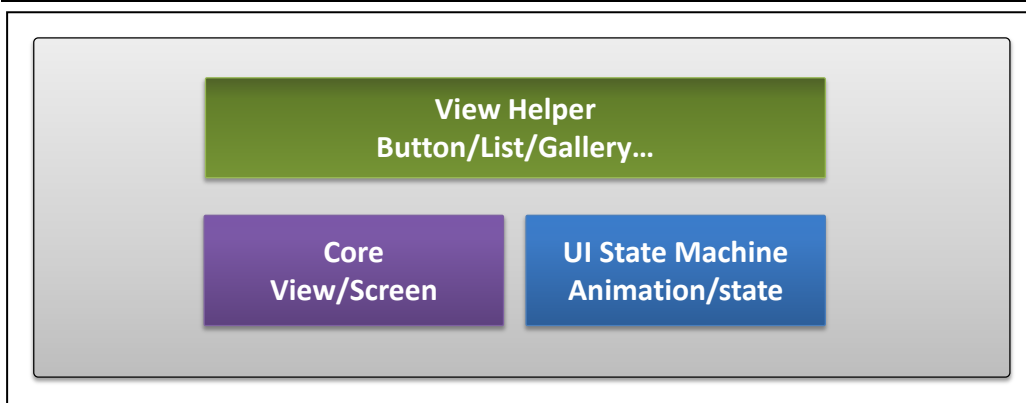
	DataSync.h KButtonView.h KCheckboxView.h KColorDrawable.h KDrawable.h KEasingCurve.h KGalleryView.h KImageDrawable.h KLayer.h KListView.h KMoveableView.h KProcessView.h KScreen.h KScrollView.h KShapeDrawable.h KSvgAnalyzing.h KTextDrawable.h KTextView.h KTimer.h KTransition.h KView.h KViewGroup.h	ImageCoder.h MaskFilter.h NewRenderingEngine.h PathEffect.h Rasterizer.h REBitmap.h RECanvas.h REColor.h REEffectHelper.h REMatrix.h REPaint.h REPath.h REPoint.h RERect.h RERegion.h REScalar.h RETypeface.h Shader.h Xfermode.h mainforwindows.h message.h NUIAnimation.h

1.3 系统特点说明

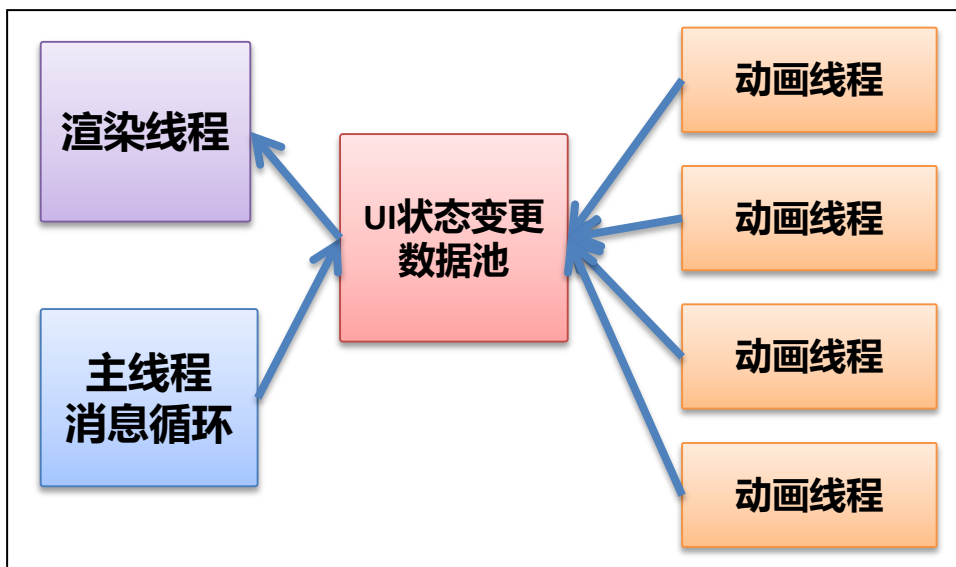
本系统采用创新性的异步渲染架构和动画驱动架构，结合高效的图形渲染引擎，在复杂动画呈现的动态界面下，能够快速响应用户消息，应用程序开发人员可以使用本平台构造自然流畅的用户体验。

1. 易于实现 NUI 界面，能够很好的支持界面动态效果
2. 界面定义脚本化，工具化，可以提高界面开发和维护的效率
3. 支持多个平台，减少软件移植工作量
4. 支持多种输入设备及手势识别

使用 NUI 引擎进行应用开发，主要是在 NUI 架构下，使用各类 View 的控件实现特定的需求，必要时可以派生特定的 View 类进行定制化功能。同时开发中多利用 NUI 擅长的动态机制实现能够提升用户体验的界面交互设计。



异步的动态渲染架构



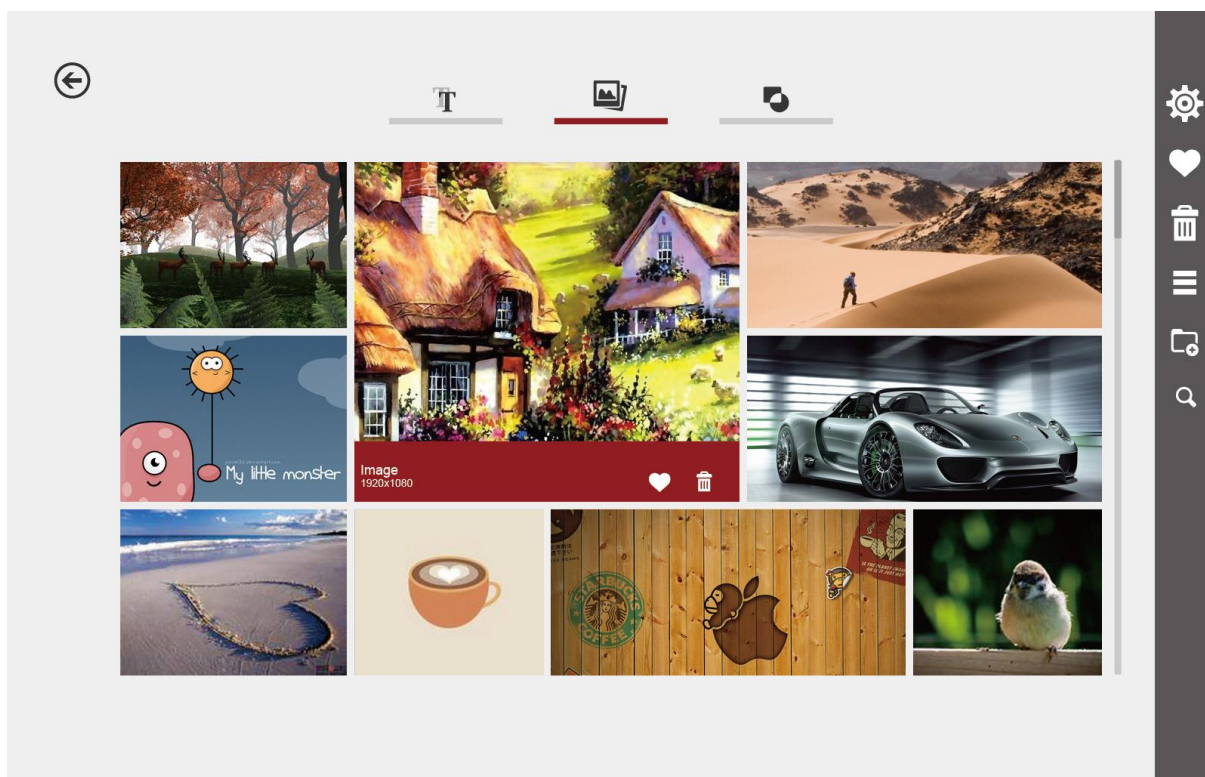
2 功能说明

2.1 图形用户界面

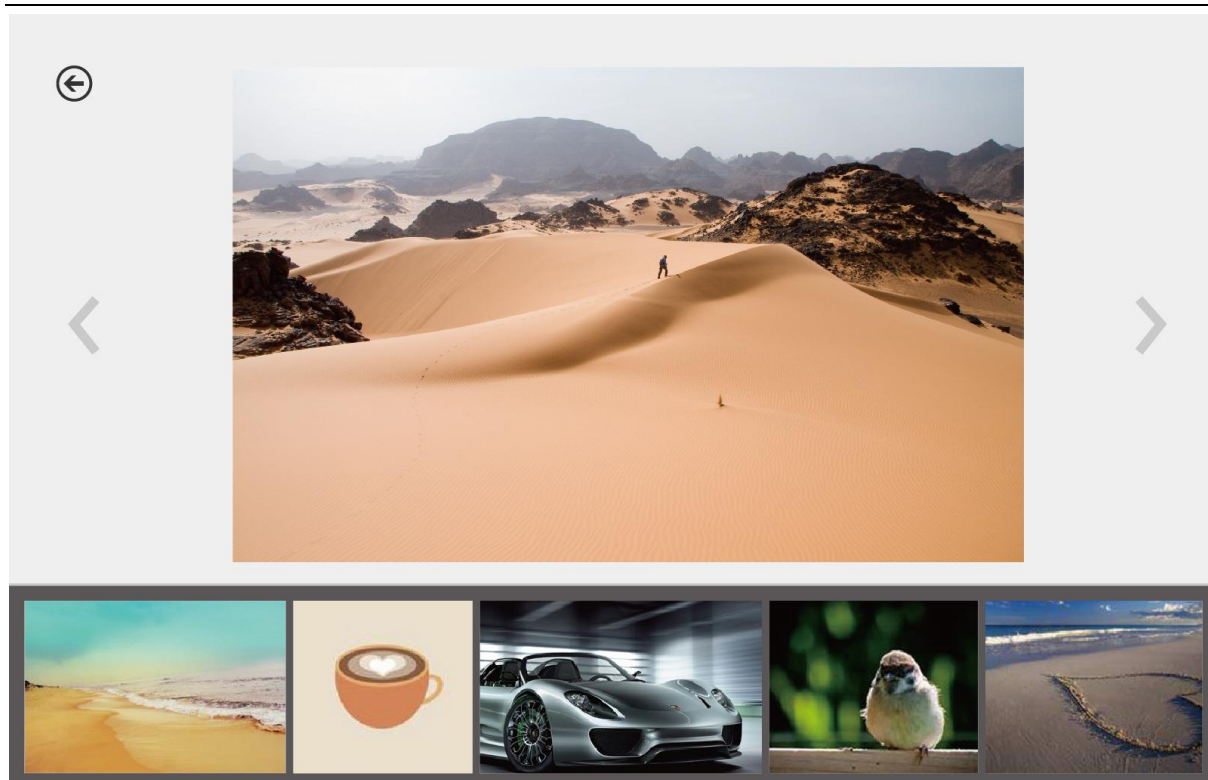
NUI 引擎提供实现图形用户界面的全套控件，同时支持鼠标和触摸屏手势操作。原生控件支持如下表，用户还可基于原生控件架构开发扩展控件。

控件	说明
按钮	可指定四种显示状态，支持背景，图标，及名称的设定
文本框	显示文本信息
进度条	显示进度信息，通过前景和背景图设置
滑动条	可指定背景和前景滑块的图片资源，实现滑块移动的滑动条效果
列表框	列表框支持手势操作和滑动动画，并可容纳组合控件
Gallery 容器	网格形容器，可容纳不同大小的控件，按照网格对其
矢量图像容器	支持 svg 矢量图导入
可拖动控件	支持交互操作的可拖动界面控件

使用 NUI 图形界面引擎，配合良好的界面设计，可以实现很美观的程序效果，如下图使用 NUI 开发库开发的图片浏览程序。



使用 NUI 引擎实现的图片浏览器



使用 NUI 引擎实现的图片浏览器

2.2 动画组件

2.2.1 动画效果

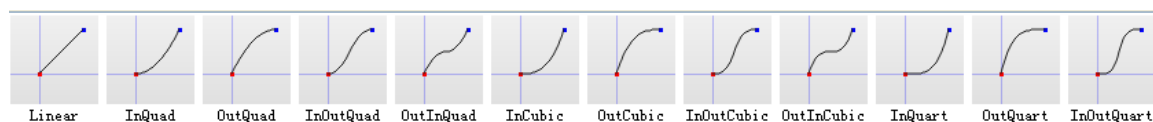
NUI 的最大特点是界面元素会有丰富的动画效果，NUI 引擎支持如下特性动画，可以很方便实现动画效果。

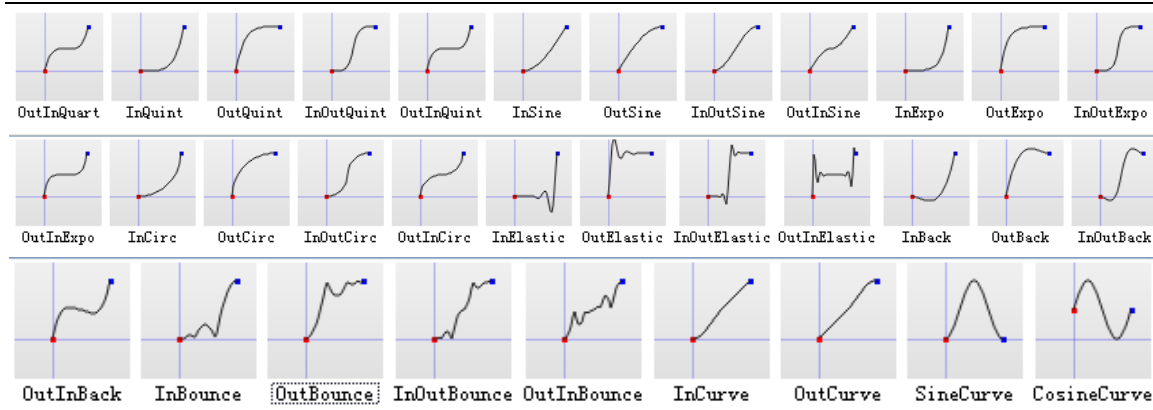
支持的动画效果	说明
移动动画	通过指定起点，终点位置信息，实现界面元素位移动画
透明度动画	实现界面元素渐变透明动画
旋转动画	实现界面元素旋转动画
裁剪区动画	实现界面元素局部区域遮罩动画
缩放动画	实现界面元素尺寸大小变化的缩放动画

2.2.2 运动曲线

要构造自然的动画运动，在运动节奏上需要符合自然的运动体验，有一定加速和减速的过程。有时还需要有抖动感，NUI 引擎可以直接生成这种效果，仅仅在构造动画时使用运动曲线就能完成。

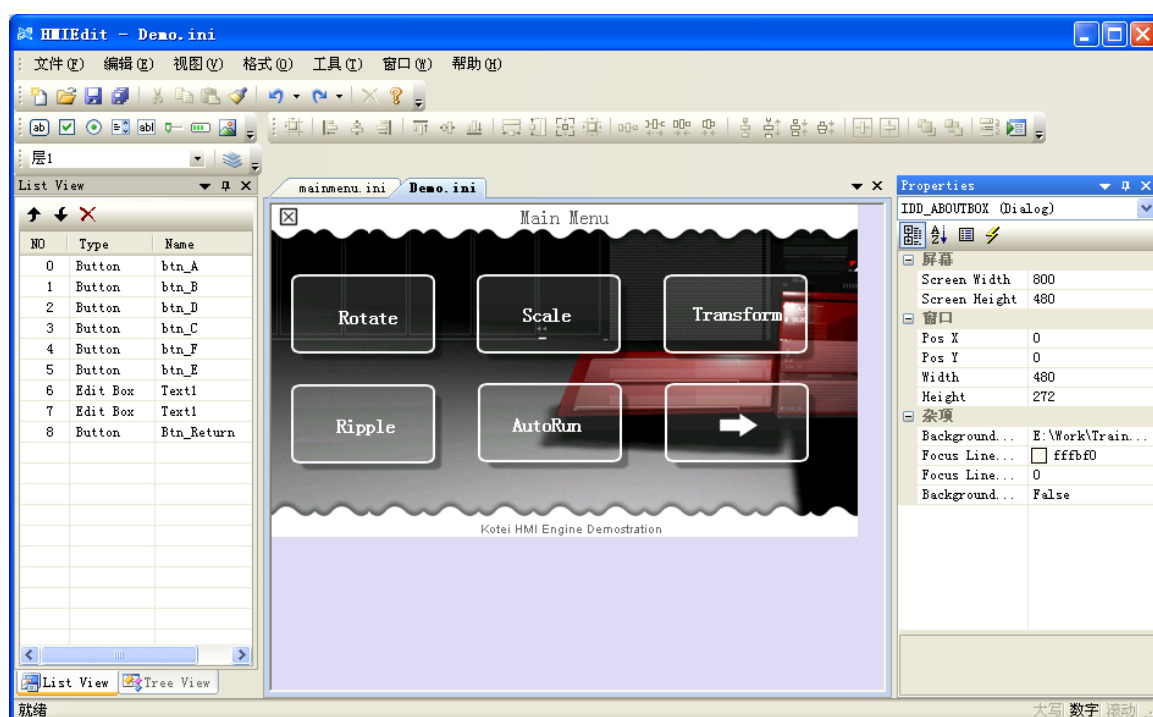
NUI 引擎内置了 45 种运动曲线，可满足大部分动画运动的需要。





2.3 界面编辑工具

NUI 引擎提供界面的设计编辑工具，设计的界面可以直接存储为接口文件供程序运行时加载使用

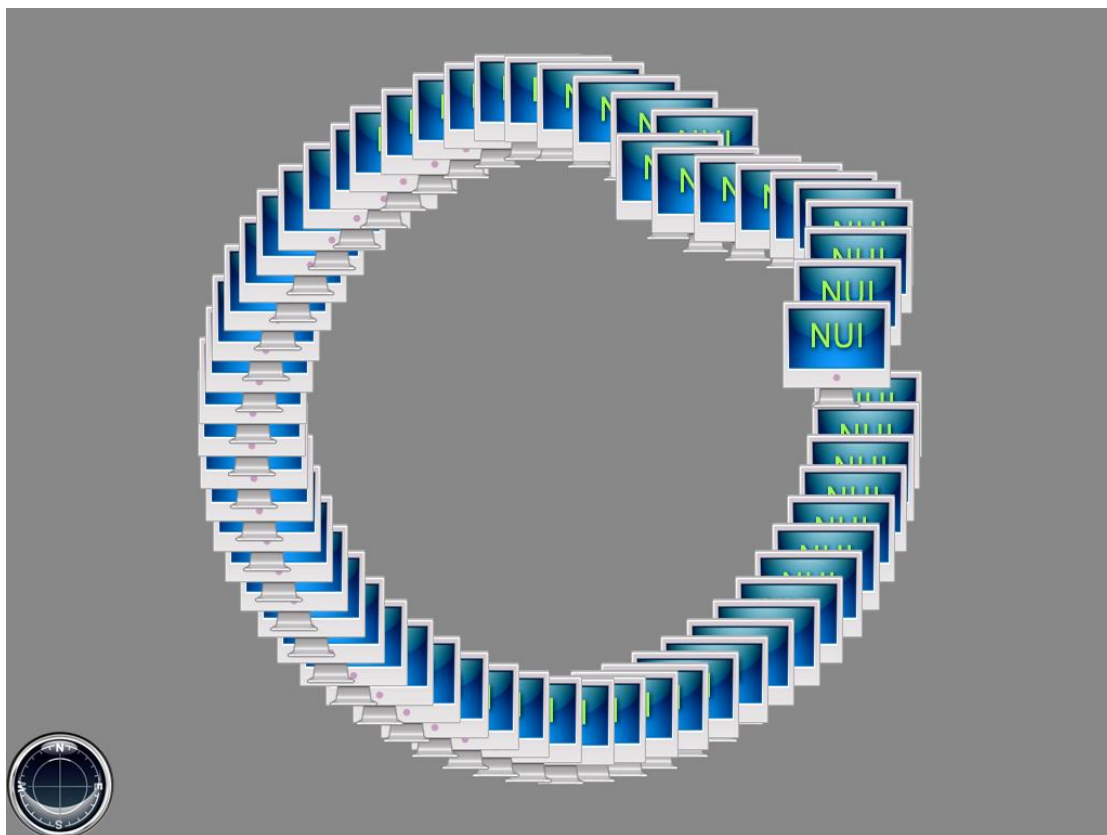


利用该工具，用户可以直接编辑显示的界面元素，而不必使用代码控制界面布局

3 示例工程

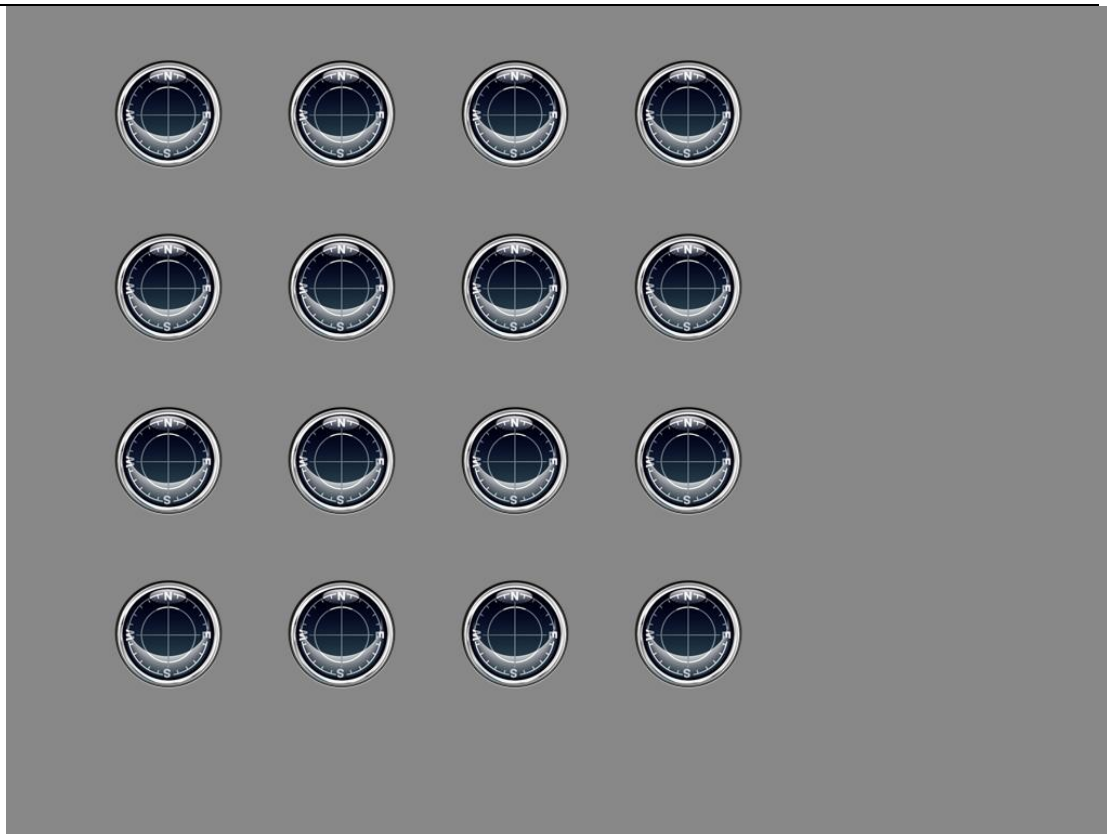
系统提供 5 个示例工程用以演示 NUI 引擎的效果及使用方法

3.1 Demo1



演示动画机制及交互响应速度

3.2 Demo2



演示多个同步动画处理及多点触摸效果

只能在 Win7 以上系统运行，请在 Acer 的 windows 平板上进行多点触控测试

3.3 Demo3



演示按钮事件的动态定义效果

3.4 Demo4

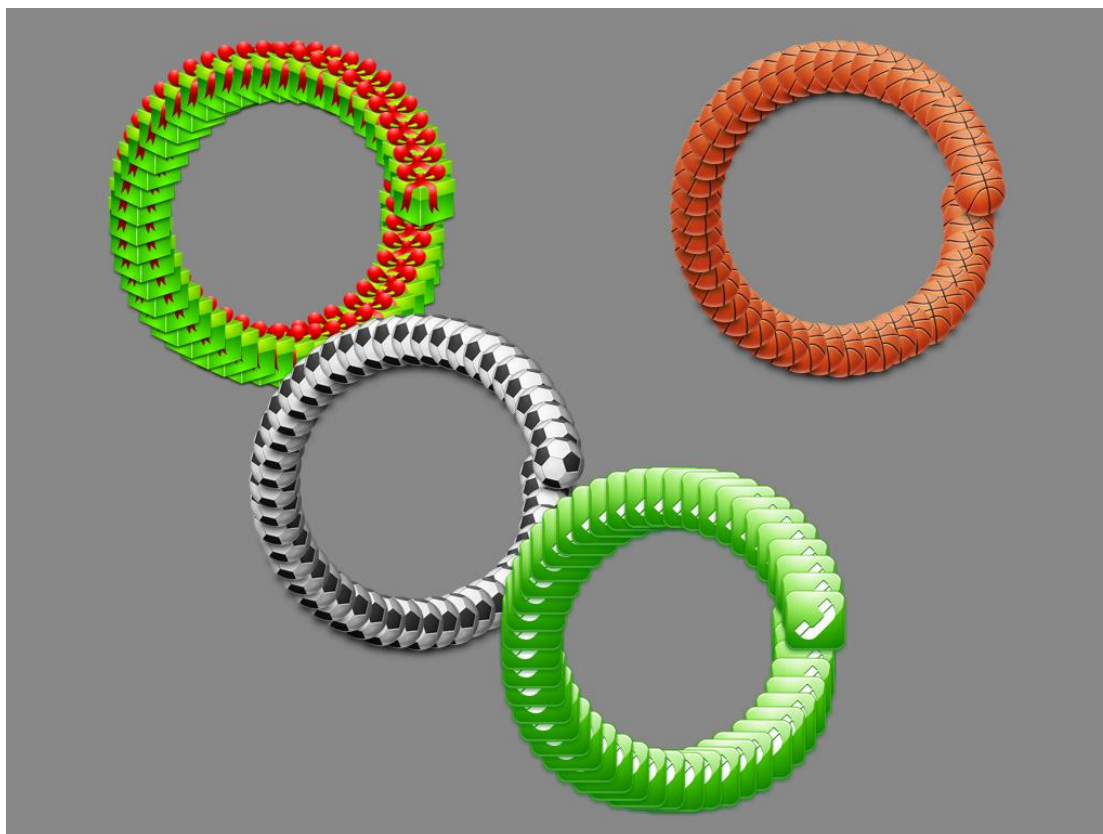
My Car Diary

February 2013

MON	TUE	WED	THU	FRI	SAT	SUN
28 🚗 180km	29 🚗 180km	30 🚗 180km	31 🚗 180km	1 🚗 180km	2 🚗 180km	3 🚗 180km
4 🚗 180km	5 🚗 180km	6 🚗 180km	7 🚗 180km	8 🚗 180km	9 🚗 180km	10 🚗 180km
11 🚗 180km	12 🚗 180km	13 🚗 180km	14 🚗 180km	15 🚗 180km	16 🚗 180km	17 🚗 180km
18 🚗 180km	19 🚗 180km	20 🚗 180km	21 🚗 180km	22 🚗 180km	23 🚗 180km	24 🚗 180km
25 🚗 180km	26 🚗 180km	27 🚗 180km	28 🚗 180km	1 🚗 180km	2 🚗 180km	3 🚗 180km

一个汽车行驶记录日历应用的效果演示

3.5 Demo5



演示多个窗口的切换效果

演示多个动画窗口同屏交互和叠加显示效果

（使用左右方向键在三个测试窗口间切换，同时鼠标可以同界面元素交互）

4 API 参考

4.1 addAnimationData

名称	CNUIAAnimation::addAnimationData			
内容	kn_bool addAnimationData(CPropertyAnimationData* p);			
功能说明	加入动画数据			
参数说明	名称	数据类型	输入输出	详细说明
	p	CPropertyAnimationData	IN	添加的动画数据指针
返回值	True, 添加动画数据成功 ; False, 添加动画数据失败			
使用注意事项	动画播放过程中, 包括 pause 都是不能加入的			
线程安全	是			
示例代码	<pre> CPropertyClip * pPropertyClip = new CPropertyClip (pltemView, rectTarget); CNUIAAnimation* pNUIAAnimation = new CNUIAAnimation (2000, 0, 0); CPropertyAnimationData* pAnimationData = new CPropertyAnimationData(pPropertyClip); pNUIAAnimation->addAnimationData(pAnimationData); </pre>			

5 API 参考

5.1 KDrawable::enableShow

名称	KDrawable::enableShow			
内容	void KDrawable::enableShow(kn_bool b);			
功能说明	设置 KDrawable 是否显示			
参数说明	名称	数据类型	输入输出	详细说明
	b	kn_bool	IN	是否显示
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	<pre> KTextDrawable_PTR drawable = KTextDrawable_PTR (new </pre>			

	<pre>KTextDrawable(_T("text")) ; drawable ->enableShow(true);</pre>
--	---

5.2 KDrawable::SetBound

名称	KDrawable::SetBound			
内容	void KDrawable::SetBound(RERect rect);			
功能说明	设置 KDrawable 的绘制范围			
参数说明	名称	数据类型	输入输出	详细说明
	rect	RERect	IN	绘制的矩形范围 矩形的坐标是相对 View 的坐标
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	<pre>KTextDrawable_PTR drawable = KTextDrawable_PTR (new KTextDrawable(_T("text"))); drawable ->SetBound(RERect::MakeXYWH(5, 5, 100, 100));</pre>			

5.3 KDrawable::GetBound

名称	KDrawable:: GetBound			
内容	RERect KDrawable::GetBound();			
功能说明	获取 KDrawable 的绘制范围			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	返回 KDrawable 的绘制范围			
使用注意事项	无			
线程安全	是			

示例代码	<code>RERect rect = m_drawable_cycle->GetBound();</code>
------	---

5.4 KDrawable::SetPaint

名称	KDrawable::SetPaint			
内容	<code>void KDrawable::SetPaint(REPaint* pPaint);</code>			
功能说明	设置 KDrawable 的 REPaint			
参数说明	名称	数据类型	输入输出	详细说明
	pPaint	REPaint*	IN	Drawable 的 相关的绘制属性设置
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	<pre>KShapeDrawable_PTR drawable = KShapeDrawable_PTR(new KShapeDrawable(pShape)); drawable->SetPaint(pPaint);</pre>			

5.5 KDrawable::GetPaint

名称	KDrawable::GetPaint			
内容	<code>REPaint* KDrawable::GetPaint()</code>			
功能说明	获取 KDrawable 的 REPaint			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	返回 KDrawable 的 REPaint			
使用注意事项	无			
线程安全	是			
示例代码	<pre>KTextDrawable_PTR drawable = KTextDrawable_PTR (new KTextDrawable(_T("text"));</pre>			

	REPaint* paint = drawable ->GetPaint();
--	---

5.6 KDrawable::SetClip

名称	KDrawable::SetClip			
内容	void KDrawable::SetClip(RERect rect)			
功能说明	设置 KDrawable 的裁剪区域			
参数说明	名称	数据类型	输入输出	详细说明
	rect	RERect	IN	裁剪区域的矩形，该矩形的坐标相对于整个 screen 坐标
返回值	Void			
使用注意事项	无			
线程安全	是			
示例代码	drawable->SetClip(rect);			

5.7 KDrawable::GetClip

名称	KDrawable::GetClip			
内容	RERect KDrawable::GetClip ()			
功能说明	获取 KDrawable 的裁剪区域			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	返回 RERect KDrawable 的裁剪区域			
使用注意事项	该矩形的坐标是相对于整个 screen 坐标			
线程安全	是			
示例代码	RERect rect = drawable->GetClip();			

5.8 KDrawable::SetClipEnable

名称	KDrawable::SetClipEnable
----	--------------------------

内容	void KDrawable::SetClipEnable(kn_bool b)			
功能说明	设置 KDrawable 的裁剪区域是否可用			
参数说明	名称	数据类型	输入输出	详细说明
	b	kn_bool	IN	裁剪区域是否可用
返回值	void			
使用注意事项	默认裁剪区域不可用 SetClip 时裁剪区域自动开启			
线程安全	是			
示例代码	drawable-> SetClipEnable (true);			

5.9 KDrawable::getType

名称	KDrawable::getType			
内容	KDrawable::KDRAWABLETYPE KDrawable::getType()			
功能说明	获取 KDrawable 的类型			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	KDRAWABLE_BASE = 0, // KDrawable KDRAWABLE_COLOR = 1, // KColorDrawable KDRAWABLE_TEXT = 2, // KTextDrawable KDRAWABLE_IMAGE = 3, // KImageDrawable KDRAWABLE_SHAPE = 4 // KShapeDrawable			
使用注意事项	无			
线程安全	是			
示例代码	<pre> If(drawable-> getType() == KDrawable::KDRAWABLE_TEXT) { } </pre>			

5.10 KColorDrawable::KColorDrawable

名称	KColorDrawable::KColorDrawable			
内容	KColorDrawable::KColorDrawable(REColor color)			
功能说明	KColorDrawable 的构造函数,创建一个可以绘制的颜色对象			
参数说明	名称	数据类型	输入输出	详细说明
	color	REColor	IN	颜色值 32 位, ARGB
返回值	KcolorDrawable 对象			
使用注意事项	建议使用智能指针形式保存 Drawable 指针			
线程安全	是			
示例代码	<pre>KColorDrawable_PTR m_bk_drawable = KColorDrawable_PTR(new KColorDrawable(RE_ColorBLACK));</pre>			

5.11 KColorDrawable::setColor

名称	KColorDrawable::setColor			
内容	void KColorDrawable::setColor(REColor color)			
功能说明	设置 KcolorDrawable 的颜色			
参数说明	名称	数据类型	输入输出	详细说明
	color	REColor	IN	颜色值 32 位, ARGB
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	<pre>KColorDrawable_PTR m_bk_drawable = KColorDrawable_PTR(new KColorDrawable(RE_ColorBLACK)); m_bk_drawable->setColor(RE_ColorWHITE);</pre>			

5.12 KColorDrawable::getColor

名称	KColorDrawable::getColor
----	--------------------------

内容	REColor KColorDrawable::getColor()			
功能说明	获取 KColorDrawable 的颜色			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	返回 32 位颜色值			
使用注意事项	无			
线程安全	是			
示例代码	<pre>KColorDrawable_PTR m_bk_drawable = KColorDrawable_PTR(new KColorDrawable(RE_ColorBLACK)); REColor = m_bk_drawable->getColor();</pre>			

5.13 KImageDrawable::KImageDrawable

名称	KImageDrawable::KImageDrawable			
内容	KImageDrawable::KImageDrawable()			
功能说明	KImageDrawable 的构造函数，用于创建可绘制的 Image			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	返回 KImageDrawable 对象			
使用注意事项	建议使用智能指针			
线程安全	是			
示例代码	<pre>KImageDrawable_PTR imagedrawable = KImageDrawable_PTR(new KImageDrawable ());</pre>			

5.14 KImageDrawable::KImageDrawable(const kn_string& strFile)

名称	KImageDrawable::KImageDrawable(const kn_string& strFile)			
内容	KImageDrawable::KImageDrawable(const kn_string& strFile)			

功能说明	通过图片文件创建 KImageDrawable			
参数说明	名称	数据类型	输入输出	详细说明
	strFile	kn_string&	IN	图片文件的文件名
返回值	返回 KImageDrawable 对象			
使用注意事项	建议使用智能指针			
线程安全	是			
示例代码	<pre>KImageDrawable_PTR imagedrawable = KImageDrawable_PTR(new KImageDrawable (_T("1.png")));</pre>			

5.15 KImageDrawable::KImageDrawable(IRESurface* pSurface)

名称	KImageDrawable::KImageDrawable(IRESurface* pSurface)			
内容	KImageDrawable::KImageDrawable(IRESurface* pSurface)			
功能说明	从内存中的 Surface 创建 KImageDrawable			
参数说明	名称	数据类型	输入输出	详细说明
	pSurface	IRESurface*	IN	内存中的 Surface
返回值	返回 KImageDrawable 对象			
使用注意事项	建议使用智能指针			
线程安全	是			
示例代码	<pre>IRESurface* pSurface = RESurfaceFactory::CreateRESurface(); KImageDrawable_PTR imagedrawable = KImageDrawable_PTR(new KImageDrawable (pSurface));</pre>			

5.16 KImageDrawable::CreateFromFile

名称	KImageDrawable::CreateFromFile
内容	kn_bool KImageDrawable::CreateFromFile(const kn_string& strFile)

功能说明	从图片文件创建 KImageDrawable			
参数说明	名称	数据类型	输入输出	详细说明
	strFile	kn_string&	IN	图片文件的文件名
返回值	返回 true 成功， false 失败			
使用注意事项	无			
线程安全	是			
示例代码	kn_bool bOK = m_bk_drawable->CreateFromFile(path);			
示例代码	KImageDrawable_PTR daBkg = KImageDrawable_PTR(new KImageDrawable(strBkgFile)); daBkg->SetFillType(KImageDrawable::FILL_ALL);			

5.17 KTextDrawable::KTextDrawable

名称	KTextDrawable::KTextDrawable			
内容	KTextDrawable::KTextDrawable(const kn_char* szText, REColor color = RE_ColorBLACK, kn_int iFontSize = 20, REPaint::Align align = REPaint::kLeft_Align);			
功能说明	KTextDrawable 的构造函数			
参数说明	名称	数据类型	输入输出	详细说明
	szText	const kn_char*	IN	文本字符串
	color	REColor	IN	文本颜色，默认黑色
	iFontSize	kn_int	IN	文字大小，默认 20
	align	REPaint::Align	IN	对齐方式，默认左对齐
返回值	KTextDrawable 对象			
使用注意事项	无			
线程安全	是			
示例代码	KTextDrawable_PTR text_drawable = KTextDrawable_PTR (new			

	<code>KTextDrawable(_T("EE"), RE_ColorBLACK, 16, REPaint::kLeft_Align));</code>
--	--

5.18 KTextDrawable::SetText

名称	KImageDrawable::SetText			
内容	<code>void KTextDrawable::SetText(const kn_char* szText)</code>			
功能说明	设置 KTextDrawable 的文本内容			
参数说明	名称	数据类型	输入输出	详细说明
	szText	const kn_char*	IN	设置的文本内容
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	<code>text_drawable-> SetText(_T("text"));</code>			

5.19 KTextDrawable::GetText

名称	KTextDrawable::GetText			
内容	<code>const kn_string& KTextDrawable::GetText()</code>			
功能说明	获取 KTextDrawable 的文本内容			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	返回文本内容			
使用注意事项	无			
线程安全	是			
示例代码	<code>kn_string str = text_drawable-> GetText();</code>			

5.20 KTextDrawable::setColor

名称	KTextDrawable::setColor
内容	<code>void KTextDrawable::setColor(REColor color)</code>

功能说明	设置文本颜色			
参数说明	名称	数据类型	输入输出	详细说明
	color	REColor	IN	文本颜色 32 位值
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	text_drawable-> setColor(RE_ColorBLACK);			

5.21 KTextDrawable::getColor

名称	KTextDrawable::getColor			
内容	REColor KTextDrawable::getColor()			
功能说明	获取文本颜色			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	REColor			
使用注意事项	无			
线程安全	是			
示例代码	REColor color = text_drawable-> getColor();			

5.22 KTextDrawable::SetFontSize

名称	KTextDrawable::SetFontSize			
内容	void KTextDrawable::SetFontSize(kn_int iFontSize)			
功能说明	设置文字大小			
参数说明	名称	数据类型	输入输出	详细说明
	iFontSize	kn_int	IN	字体大小
返回值	void			

使用注意事项	无
线程安全	是
示例代码	<code>text_drawable-> SetFontSize (10);</code>

5.23 KTextDrawable::GetFontSize

名称	KTextDrawable::GetFontSize			
内容	<code>kn_int KTextDrawable::GetFontSize()</code>			
功能说明	获取文字大小			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	<code>kn_int</code> 文字大小			
使用注意事项	无			
线程安全	是			
示例代码	<code>kn_int iFontSize -= text_drawable-> GetFontSize();</code>			

5.24 KTextDrawable::setFontName

名称	KTextDrawable::setFontName			
内容	<code>void KTextDrawable::setFontName(const char* fontName)</code>			
功能说明	获取文字大小			
参数说明	名称	数据类型	输入输出	详细说明
	<code>fontName</code>	<code>const char*</code>	IN	系统安装的字体名称
返回值	<code>void</code>			
使用注意事项	无			
线程安全	是			
示例代码	<code>m_text_drawable->setFontName("Microsoft YaHei");</code>			

5.25 KTextDrawable::GetFontName

名称	KTextDrawable::GetFontName			
内容	const char* KTextDrawable::GetFontName() const			
功能说明	获取字体名称			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	const char* pszFontName = m_text_drawable-> GetFontName ();			

5.26 KTextDrawable::setTextAlign

名称	KTextDrawable::setTextAlign			
内容	void KTextDrawable::setTextAlign(REPaint::Align align)			
功能说明	设置文字对齐方式			
参数说明	名称	数据类型	输入输出	详细说明
	align	REPaint::Align	IN	kLeft_Align, // 左对齐 kCenter_Align, // 中间对齐 kRight_Align, // 右对齐
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	m_text_drawable-> setTextAlign (REPaint::kCenter_Align);			

5.27 KTextDrawable::getTextAlign

名称	KTextDrawable::getTextAlign			
内容	REPaint::Align KTextDrawable::getTextAlign()			

功能说明	获取文字对齐方式			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	REPaint::Align 对齐方式:			
使用注意事项	无			
线程安全	是			
示例代码	REPaint::Align align = m_text_drawable->getTextAlign ();			

5.28 KTextDrawable::setBold

名称	KTextDrawable::setBold			
内容	void KTextDrawable::setBold(kn_bool b)			
功能说明	设置文字是否为粗体			
参数说明	名称	数据类型	输入输出	详细说明
	b	kn_bool	IN	是否为粗体
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	m_text_drawable-> setBold (true);			

5.29 KTextDrawable::isBold

名称	KTextDrawable::isBold			
内容	kn_bool KTextDrawable::isBold()			
功能说明	获取文字是否为粗体			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	kn_bool 文字是否为粗体			

使用注意事项	无
线程安全	是
示例代码	kn_bool b = m_text_drawable->isBold ();

5.30 KTextDrawable::setOutBoundDot

名称	KTextDrawable::setOutBoundDot			
内容	void KTextDrawable::setOutBoundDot()			
功能说明	设置文字超过 Drawable 区域时添加省略号			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	无			
使用注意事项	<p>该函数处理文字超长的问題。该接口调用后，会根据 Text 文本的长度和 Drawable 范围修改 Text 的内容。</p> <p>所以当 Text 内容被重新设置，或者 Drawable 的范围被重新设置之后，或者字体被重新设置后，需要重新调用该接口。</p>			
线程安全	是			
示例代码	<pre> KTextDrawable_PTR pTextDrawable = KTextDrawable_PTR (new KTextDrawable(strMessage.c_str(), RE_ColorWHITE, 30, REPaint::kCenter_Align)); pTextDrawable->SetBound(RERect::MakeXYWH(0, 0, m_rect.width() , m_rect.height())); pTextDrawable ->setFontName("Calibri"); // 第一次设置文字超长处理 pTextDrawable -> setOutBoundDot(); </pre>			

	<pre>// ... pTextDrawable ->setFontSize("26"); // 文本字体改变后，需重新设置文字超长处理 pTextDrawable -> setOutBoundDot();</pre>
--	--

5.31 KShapeDrawable::KShapeDrawable

名称	KShapeDrawable::KShapeDrawable			
内容	KShapeDrawable::KShapeDrawable(KShape* pShape)			
功能说明	KShapeDrawable 的构造函数			
参数说明	名称	数据类型	输入输出	详细说明
	pShape	Kshape*	IN	形状信息,现有的形状包括 KRectShape KOvalShape KRoundRectShape KCircleShape KLineShape KPolygonShape KPathShape
返回值	KShapeDrawable 对象			
使用注意事项	建议使用智能指针			
线程安全	是			
示例代码	<pre>KRectShape* pRectShape = new KRectShape(RERect::MakeXYWH(0, 0, 400, 50)); KShapeDrawable_PTR s_bkg_image = KShapeDrawable_PTR(new KShapeDrawable(pRectShape));</pre>			

5.32 KShapeDrawable::SetShape

名称	KShapeDrawable:: SetShape			
内容	void KShapeDrawable::SetShape(KShape* pShape)			
功能说明	设置形状			
参数说明	名称	数据类型	输入输出	详细说明
	pShape	Kshape*	IN	形状信息,现有的形状包括 KRectShape KOvalShape KRoundRectShape KCircleShape KLineShape KPolygonShape KPathShape
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	<pre>KRoundRectShape* pRRectShape = new KRoundRectShape (RRect::MakeXYWH(0, 0, iWidth, iHeight), 2, 2); m_drawable_slider->SetShape(pRRectShape);</pre>			

5.33 KSVGDrawable::KSVGDrawable

名称	KSVGDrawable::KSVGDrawable			
内容	KSVGDrawable::KSVGDrawable()			
功能说明	KSVGDrawable 构造函数			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	KSVGDrawable 对象			

使用注意事项	建议使用智能指针
线程安全	是
示例代码	KSVGDrawable_PTR svgdrawable = KSVGDrawable_PTR(new KSVGDrawable);

5.34 KSVGDrawable::KSVGDrawable(kn_string svgFilePath)

名称	KSVGDrawable::KSVGDrawable(kn_string svgFilePath)			
内容	KSVGDrawable::KSVGDrawable(kn_string svgFilePath)			
功能说明	KSVGDrawable 构造函数，通过 SVG 文件创建			
参数说明	名称	数据类型	输入输出	详细说明
	kn_string	svgFilePath	IN	SVG 文件路径
返回值	KSVGDrawable 对象			
使用注意事项	建议使用智能指针			
线程安全	是			
示例代码	KSVGDrawable_PTR svgdrawable = KSVGDrawable_PTR(new KSVGDrawable(_T("1.svg")));			

5.35 KSVGDrawable::KSVGDrawable(kn_string svgFilePath, RERect& rect)

名称	KSVGDrawable::KSVGDrawable(kn_string svgFilePath, RERect& rect)			
内容	KSVGDrawable::KSVGDrawable(kn_string svgFilePath, RERect& rect)			
功能说明	KSVGDrawable 构造函数，通过 SVG 文件和包围盒创建			
参数说明	名称	数据类型	输入输出	详细说明
	svgFilePath	kn_string	IN	SVG 文件路径
	rect	RERect&	IN	包围盒
返回值	KSVGDrawable 对象			
使用注意事项	建议使用智能指针			
线程安全	是			

示例代码	<pre>KSVGDrawable_PTR svgdrawable = KSVGDrawable_PTR(new KSVGDrawable(_T("1.svg"), RERect::MakeXYWH(0, 0, iWidth, iHeight)));</pre>
------	---

5.36 KSVGDrawable::addLine

名称	KSVGDrawable::addLine			
内容	void KSVGDrawable::addLine(REPoint p1, REPoint p2, REPaint* pPaint)			
功能说明	添加线段			
参数说明	名称	数据类型	输入输出	详细说明
	p1	REPoint	IN	线段起点
	p2	REPoint	IN	线段终点
	pPaint	REPaint*	IN	线段的式样
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	<pre>REPoint points[] = { {17.981, 31.401},{17.981, 114.08}, {56.815, 31.401},{56.815, 114.08}, }; int iArraysize = sizeof(points)/sizeof(points[0]); REPaint paintLine; paintLine.setColor(0xffff0000); paintLine.setStrokeWidth(5); paintLine.setStrokeMiter(10); paintLine.setAntiAlias(true); for(int i = 0; i < iArraysize; i+=2) { svgdrawable->addLine(points[i], points[i+1], &paintLine); }</pre>			

5.37 KSVGDrawable::addPolygon

名称	KSVGDrawable::addPolygon			
内容	void KSVGDrawable::addPolygon(REPoint* pPoints, int count, kn_bool bClose, REPaint* pPaint)			
功能说明	添加多边形 / 或折线			
参数说明	名称	数据类型	输入输出	详细说明
	pPoints	REPoint	IN	顶点坐标
	count	int	IN	顶点个数
	bClose	kn_bool	IN	是否闭合
	pPaint	REPaint*	IN	多边形或折线式样
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	<pre>REPoint poly[3] = {{37.636,0}, {15.5,31.496}, {59.364,31.496}}; REPaint paintPoly; paintPoly.setColor(0xffc0000); paintPoly.setAntiAlias(true); svgdrawable->addPolygon(poly, 3, true, &paintPoly);</pre>			

5.38 KSVGDrawable::addPath

名称	KSVGDrawable::addPath			
内容	void KSVGDrawable::addPath(REPath& path, REPaint* pPaint)			
功能说明	添加多边形 / 或折线			
参数说明	名称	数据类型	输入输出	详细说明
	path	REPath&	IN	Path, 各种 shape 的集合
	pPaint	REPaint*	IN	Path 的式样
返回值	void			

使用注意事项	无
线程安全	是
示例代码	<pre>REPoint pointStart = {3, 157.051}; REPoint pointControl1 = {36.071, 122.013}; REPoint pointControl2 = {46.307, 49.572}; REPoint pointEnd = {46.307, 49.572}; // 三次贝塞尔 REPath path; path.moveTo(pointStart); path.cubicTo(pointControl1, pointControl2, pointEnd); path.lineTo(30.953, 60.99); path.lineTo(52.141, 5.872); path.rLineTo(20.938, 56.299); path.lineTo(57.724, 49.966); path.rCubicTo(0, 0, 1.969, 82.678, 10.63, 116.534); REPaint paintPath; paintPath.setAntiAlias(true); paintPath.setStyle(REPaint::Style::kStroke_Style); paintPath.setStrokeWidth(3); paintPath.setStrokeMiter(10); paintPath.setColor(0xff90606); KSVGDrawable_PTR svgdrawable2 = KSVGDrawable_PTR(new KSVGDrawable); svgdrawable2->SetBound(ERect::MakeXYWH(150, 0, 150, 200));</pre>

	svgdrawable2->addPath(path, &paintPath);
--	--

5.39 KSVGDrawable::addSVGShape

名称	KSVGDrawable::addSVGShape			
内容	Void KSVGDrawable::addSVGShape(KShape* pShape, REPaint* pFillPaint, REPaint* pStorkPaint)			
功能说明	添加 SVG 形状 Polygon, Path, RoundRect, Circle, Ellipse			
参数说明	名称	数据类型	输入输出	详细说明
	pShape	KShape*	IN	SVG 形状 Polygon, Path, RoundRect, Circle, Ellipse...
	pFillPaint	REPaint*	IN	填充式样
	pStorkPaint	REPaint*	IN	边线式样
返回值	void			
使用注意事项	无			
线程安全	是			
示例代码	<pre>((KOvalShape*)m_pShapeList.at(i))->InitPaint(); REPaint fillPaint = ((KOvalShape*)m_pShapeList.at(i))->m_fillPaint; REPaint strokePaint = ((KOvalShape*)m_pShapeList.at(i))->m_strokePaint; this-> addSVGShape ((KOvalShape*)m_pShapeList.at(i),&fillPaint,&strokePaint);</pre>			

5.40 KView::Release

名称	KView::Release			
内容	virtual void Release();			
功能说明	销毁 view 列表			
参数说明	名称	数据类型	输入输出	详细说明
返回值	无			

使用注意事项	view 还需要使用时，不能调用。
线程安全	是
示例代码	无

5.41 KView::Create

名称	KView:: Create			
内容	virtual kn_bool Create(kn_int iX, kn_int iY, kn_int iWidth, kn_int iHeight);			
功能说明	创建 view 区域			
参数说明	名称	数据类型	输入输出	详细说明
	iX	kn_int	IN	设置 view 区域左上角 x 坐标
	iY	kn_int	IN	设置 view 区域左上角 y 坐标
	iWidth	kn_int	IN	设置 view 区域的宽
	iHeight	kn_int	IN	设置 view 区域的高
返回值	True, view 的创建区域成功 ; False, 创建 view 的区域失败			
使用注意事项	x 坐标和 y 坐标是相对 view 上级的			
线程安全	是			
示例代码	无			

5.42 KView::Create

名称	KView::Create			
内容	virtual kn_bool Create(Rrect& rect)			
功能说明	创建 view 的区域			
参数说明	名称	数据类型	输入输出	详细说明
	rect	Rrect	IN	添加矩形区域
返回值	True, 创建 view 的区域成功 ; False, 创建 view 的区域失败			
使用注意事项	无			
线程安全	是			

示例代码	无
------	---

5.43 KView::RefreshSurface

名称	KView::RefreshSurface			
内容	void RefreshSurface (IRESurface* pDstSurface, kn_int x, kn_int y);			
功能说明	将 view 绘制到指定的 IRESurface 上的指定位置			
参数说明	名称	数据类型	输入输出	详细说明
	pDstSurface	IRESurface	IN	目标 IRESurface 的指针
	x	kn_int	IN	指定 view 左上角绘制到画布上的 x 坐标
	y	kn_int	IN	指定 view 左上角绘制到画布上的 y 坐标
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.44 KView::enableCache

名称	KView::enableCache			
内容	void enableCache(kn_bool b);			
功能说明	控制是否刷新隐藏的缓冲区域			
参数说明	名称	数据类型	输入输出	详细说明
	b	kn_bool	IN	是否刷新隐藏的缓冲区域，True 允许刷新，False 不允许刷新
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.45 KView::GetScreen

名称	KView::GetScreen			
内容	KScreen* GetScreen ();			
功能说明	获取屏幕			
参数说明	名称	数据类型	输入输出	详细说明
返回值	获取该 view 的屏幕指针或者其上一级 view 的屏幕指针			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.46 KView::SetScreen

名称	KView::SetScreen			
内容	void SetScreen (KScreen* p);			
功能说明	设置屏幕			
参数说明	名称	数据类型	输入输出	详细说明
	p	KScreen	IN	设置屏幕指针
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.47 KView::getType

名称	KView::getType			
内容	KVIEWTYPE getType ();			
功能说明	获取 view 的类型			
参数说明	名称	数据类型	输入输出	详细说明
返回值	返回该 view 的类型，详见参见 3.2KVIEWTYPE 说明			

使用注意事项	无
线程安全	是
示例代码	无

5.48 KView::addDrawable

名称	KView::addDrawable			
内容	virtual void addDrawable(KDrawable_PTR p);			
功能说明	给 view 添加一个可绘制对象			
参数说明	名称	数据类型	输入输出	详细说明
	p	KDrawable_PTR	IN	添加一个可绘制对象。
返回值	无			
使用注意事项	注意顺序，后添加的可绘制对象会覆盖前面的。			
线程安全	是			
示例代码	<pre>KImageDrawable_PTR m_imgBkg = KImageDrawable_PTR(new KImageDrawable(strPath + _T("txtRouteDetail.png"))); m_imgBkg->SetBound(RERect::MakeXYWH(114, 5, 169, 103)); addDrawable(m_imgBkg);</pre>			

5.49 KView::enableMessage

名称	KView::enableMessage			
内容	void enableMessage(kn_bool);			
功能说明	控制 view 是否能响应消息			
参数说明	名称	数据类型	输入输出	详细说明
	b	kn_bool	IN	控制 view 响应消息 True 可以响应，false 不能响应
返回值	无			
使用注意事项	无			

线程安全	是
示例代码	无

5.50 KView::IsShown

名称	KView::IsShown			
内容	Virtual kn_bool IsShown();			
功能说明	查看 view 是否可见			
参数说明	名称	数据类型	输入输出	详细说明
返回值	返回 False, view 不可见, 返回 True, view 可见			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.51 KView::SetShow

名称	KView::SetShow			
内容	virtual void SetShow(kn_bool bShow);			
功能说明	控制 view 是否可见			
参数说明	名称	数据类型	输入输出	详细说明
	bShow	kn_bool	IN	设置为 True, view 可见, 设置为 False, view 不可见
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.52 KView::SetOpacity

名称	KView::SetOpacity			
内容	virtual void SetOpacity(kn_byte value);			

功能说明	设置 view 的透明度			
参数说明	名称	数据类型	输入输出	详细说明
	value	kn_int	IN	设置 view 的透明度。
返回值	无			
使用注意事项	设置透明度参数取值范围为 0~255，数值越小越透明。			
线程安全	是			
示例代码	m_pScaleInfo->SetOpacity(255);			

5.53 KView::GetOpacity

名称	KView::GetOpacity			
内容	virtual kn_int GetOpacity();			
功能说明	获取 view 的透明度			
参数说明	名称	数据类型	输入输出	详细说明
返回值	返回 view 的透明度。			
使用注意事项	无			
线程安全	是			
示例代码	kn_int value = m_pScaleInfo->GetOpacity();			

5.54 KView::GetPosition

名称	KView::GetPosition			
内容	virtual void GetPosition(kn_int& iX, kn_int& iY);			
功能说明	获取 view 左上角的 x 坐标和 y 坐标			
参数说明	名称	数据类型	输入输出	详细说明
	iX	kn_int	OUT	返回 view 左上角在相对于其上一级的 x 坐标
	iY	kn_int	OUT	返回 view 左上角在相对于其上一级的 y 坐标
返回值	无			

使用注意事项	无
线程安全	是
示例代码	<pre>kn_int x,y; m_p_view->GetPosition(x, y);</pre>

5.55 KView::GetPosition

名称	KView::GetPosition			
内容	virtual REPoint GetPosition();			
功能说明	获取 view 左上角坐标点			
参数说明	名称	数据类型	输入输出	详细说明
返回值	REpoint, 返回 view 左上角坐标点, 详见 3.3SkPoint			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.56 KView::GetScreen

名称	KView::GetScreenXY			
内容	virtual void GetScreenXY(kn_int &x, kn_int &y);			
功能说明	获取屏幕左上角 x 坐标和 y 坐标			
参数说明	名称	数据类型	输入输出	详细说明
	x	kn_int	OUT	获取屏幕左上角 x 坐标
	y	kn_int	OUT	获取屏幕左上角 y 坐标
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.57 KView::GetScreenPosition

名称	KView::SetScreenPosition			
内容	virtual void SetScreenPosition(kn_int iX, kn_int iY);			
功能说明	设置屏幕左上角 x 坐标和 y 坐标			
参数说明	名称	数据类型	输入输出	详细说明
	iX	kn_int	IN	设置屏幕左上角的 x 坐标
	iY	kn_int	IN	设置屏幕左上角的 y 坐标
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码				

5.58 KView::SetPosition

名称	KView::SetPosition			
内容	virtual void SetPosition(kn_int iX, kn_int iY);			
功能说明	设置 view 的位置			
参数说明	名称	数据类型	输入输出	详细说明
	iX	kn_int	IN	设置 view 左上角 x 坐标
	iY	kn_int	IN	设置 view 做上角 y 坐标
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码				

5.59 KView::Invalidate

名称	KView::Invalidate			
内容	virtual void Invalidate(RERect r);			
功能说明	设置局部刷新区域			
参数说明	名称	数据类型	输入输出	详细说明

	r	RERect	IN	设置指定矩形区域为局部刷新区域
返回值	无			
使用注意事项	设置的矩形是相对于 view 本身的的矩形区域			
线程安全	是			
示例代码	无			

5.60 KView::InvalidateView

名称	KView::InvalidateView			
内容	virtual void InvalidateView(kn_bool bRefresh=TRUE);			
功能说明	控制 view 自身是否立即要求重绘			
参数说明	名称	数据类型	输入输出	详细说明
	bRefresh	kn_bool	IN	传入 True 立即刷新 view 本身，传入 False 不刷新 view 本身，默认参数为 true。
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.61 KView::InvalidateForParent

名称	KView::InvalidateForParent			
内容	virtual void InvalidateForParent(RERect& r);			
功能说明	设置一个相对上一级的矩形区域为局部刷新区域			
参数说明	名称	数据类型	输入输出	详细说明
	r	RERect	IN	设置矩形区域为局部刷新区域
返回值	无			
使用注意事项	这个矩形区域的位置是相对 view 上一级的			

线程安全	是
示例代码	无

5.62 KView::clearInvalidRect

名称	KView::clearInvalidRect			
内容	virtual void clearInvalidRect();			
功能说明	删除局部刷新区域设置			
参数说明	名称	数据类型	输入输出	详细说明
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.63 KView::GetRect

名称	KView::GetRect			
内容	virtual RRect GetRect();			
功能说明	获取 view 在上一级中的矩形区域			
参数说明	名称	数据类型	输入输出	详细说明
返回值	返回 view 相对上一级的矩形区域			
使用注意事项	无			
线程安全	是			
示例代码	RRect POIstRect = m_pPOIAgentListView->GetRect();			

5.64 KView::SetRect

名称	KView::SetRect			
内容	virtual void SetRect(const RRect& rect);			
功能说明	设置 view 的矩形区域			

参数说明	名称	数据类型	输入输出	详细说明
	rect	RERect	IN	设置 view 相对上一级的矩形区域
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	<pre> RERect iconRect= RERect::MakeXYWH(btnRect.centerX() - 12.5, btnRect.centerY() - 6, 25, 12); m_pPOIAgentIconView->SetRect(iconRect); </pre>			

5.65 KView::GetBoundRect

名称	KView::GetBoundRect			
内容	virtual RERect GetBoundRect();			
功能说明	获取 view 变换后的包围盒			
参数说明	名称	数据类型	输入输出	详细说明
返回值	返回 view 变换后的矩形包围盒			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.66 KView::SetBoundRect

名称	KView::SetBoundRect			
内容	virtual void SetBoundRect(const RERect& rect);			
功能说明	设置 view 的包围盒			
参数说明	名称	数据类型	输入输出	详细说明
	rect	RERect	IN	设置 view 的包围盒矩形区域
返回值	无			

使用注意事项	无
线程安全	是
示例代码	<pre>RERect rect = RERect::MakeXYWH(0,0,100,100); m_p_view->SetBoundRect(rect);</pre>

5.67 KView::Translate

名称	KView::Translate			
内容	virtual void Translate(REScale dx, REScale dy);			
功能说明	view 平移			
参数说明	名称	数据类型	输入输出	详细说明
	dx	REScale	IN	view 左上角需要移动到的 x 坐标
	dy	REScale	IN	view 左上角需要移动到的 y 坐标
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.68 KView::Scale

名称	KView::Scale			
内容	virtual void Scale(REScale sx, REScale sy, REScale px = 0, REScale py = 0);			
功能说明	指定参照点对 view 进行缩放			
参数说明	名称	数据类型	输入输出	详细说明
	sx	REScale	IN	x 轴方向缩放比例
	sy	REScale	IN	y 轴方向缩放比例
	px	REScale	IN	缩放参照点的 x 坐标，默认参数为 0
	py	REScale	IN	缩放参照点的 y 坐标，默认参数为 0
返回值	无			

使用注意事项	无
线程安全	是
示例代码	无

5.69 KView::Rotate

名称	KView::Rotate			
内容	virtual void Rotate(kn_float degrees, REScalar px = 0, REScalar py = 0);			
功能说明	指定某个点为中心旋转 view			
参数说明	名称	数据类型	输入输出	详细说明
	degrees	kn_float	IN	旋转角度
	px	REScalar	IN	旋转中心点的 x 坐标
	py	REScalar	IN	旋转中心点的 x 坐标
返回值	无			
使用注意事项	第一个参数是旋转多少度，正数是顺时针，负数是逆时针			
线程安全	是			
示例代码	无			

5.70 KView::Skew

名称	KView:: Skew			
内容	virtual void Skew(kn_float sx, kn_float sy, REScalar px = 0, REScalar py = 0);			
功能说明	设置 view 倾斜			
参数说明	名称	数据类型	输入输出	详细说明
	sx	kn_float	IN	x 轴方向倾斜比例
	sy	kn_float	IN	y 轴方向倾斜比例
	px	REScalar	IN	倾斜参照点的 x 坐标
	py	REScalar	IN	倾斜参照点的 y 坐标
返回值	无			
使用注意事项	无			

线程安全	是
示例代码	无

5.71 KView::SetTranslate

名称	KView::SetTranslate			
内容	virtual void SetTranslate(REScalar dx, REScalar dy);			
功能说明	平移 view			
参数说明	名称	数据类型	输入输出	详细说明
	dx	REScalar	IN	view 左上角需要平移到的 x 坐标
	dy	REScalar	IN	view 左上角需要平移到的 y 坐标
返回值	无			
使用注意事项	该平移方法是非累加的			
线程安全	是			
示例代码	无			

5.72 KView::SetScale

名称	KView::SetScale			
内容	virtual void SetScale(REScalar sx, REScalar sy, REScalar px = 0, REScalar py = 0);			
功能说明	指定参照点对 view 进行缩放			
参数说明	名称	数据类型	输入输出	详细说明
	sx	REScalar	IN	x 轴方向缩放比例
	sy	REScalar	IN	y 轴方向缩放比例
	px	REScalar	IN	缩放参照点的 x 坐标，默认参数为 0
	py	REScalar	IN	缩放参照点的 y 坐标，默认参数为 0
返回值	无			
使用注意事项	该缩放方法是非累加的			

线程安全	是
示例代码	无

5.73 KView::SetRotate

名称	KView::SetRotate			
内容	virtual void SetRotate(kn_float degrees, REScalar px = 0, REScalar py = 0);			
功能说明	指定一个点作为中心旋转 view			
参数说明	名称	数据类型	输入输出	详细说明
	degrees	kn_float	IN	旋转角度
	px	REScalar	IN	旋转中心点的 x 坐标
	py	REScalar	IN	旋转中心点的 x 坐标
返回值	无			
使用注意事项	该旋转方法是非累加的			
线程安全	是			
示例代码	无			

5.74 KView::SetSkew

名称	KView::SetSkew			
内容	virtual void SetSkew(kn_float sx, kn_float sy, REScalar px = 0, REScalar py = 0);			
功能说明	指定参照点设置 view 倾斜			
参数说明	名称	数据类型	输入输出	详细说明
	sx	kn_float	IN	x 轴方向倾斜比例
	sy	kn_float	IN	y 轴方向倾斜比例
	px	REScalar	IN	倾斜参照点的 x 坐标
	py	REScalar	IN	倾斜参照点的 y 坐标
返回值	无			
使用注意事项	该倾斜方法是非累加的			
线程安全	是			

示例代码	
------	--

5.75 KView::OnMove

名称	KView::OnMove			
内容	virtual void OnMove(kn_int iScreenX, kn_int iScreenY);			
功能说明	子类实现鼠标在 view 上移动时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	iScreenX	kn_int	IN	鼠标移动时的 x 坐标（相对父 Group 的 x 坐标）
	iScreenY	kn_int	IN	鼠标移动时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.76 KView::OnDown

名称	KView::OnDown			
内容	virtual void OnDown(kn_int iScreenX, kn_int iScreenY);			
功能说明	子类实现鼠标在 view 上左键下压时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	iScreenX	kn_int	IN	鼠标左键下压时的 x 坐标（相对父 Group 的 x 坐标）
	iScreenY	kn_int	IN	鼠标左键下压时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.77 KView::OnUp

名称	KView::OnUp			
内容	virtual void OnUp(kn_int iScreenX, kn_int iScreenY);			
功能说明	子类实现鼠标在 view 上左键弹起时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	iScreenX	kn_int	IN	鼠标左键弹起时的 x 坐标（相对父 Group 的 x 坐标）
	iScreenY	kn_int	IN	鼠标左键弹起时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.78 KView::OnDClick

名称	KView::OnDClick			
内容	virtual void OnDClick(kn_int iScreenX, kn_int iScreenY);			
功能说明	子类实现鼠标在 view 上左键双击时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	iScreenX	kn_int	IN	鼠标左键双击时的 x 坐标（相对父 Group 的 x 坐标）
	iScreenY	kn_int	IN	鼠标左键双击时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.79 KView::OnRDown

名称	KView::OnRDown			
内容	virtual void OnRDown(kn_int iScreenX, kn_int iScreenY);			
功能说明	子类实现鼠标在 view 上右键下压时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	iScreenX	kn_int	IN	鼠标右键下压时的 x 坐标（相对父 Group 的 x 坐标）
	iScreenY	kn_int	IN	鼠标右键下压时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.80 KView::OnRUp

名称	KView::OnRUp			
内容	virtual void OnRUp(kn_int iScreenX, kn_int iScreenY);			
功能说明	子类实现鼠标在 view 上右键弹起时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	iScreenX	kn_int	IN	鼠标右键弹起时的 x 坐标（相对父 Group 的 x 坐标）
	iScreenY	kn_int	IN	鼠标右键弹起时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.81 KView::OnRDClick

名称	KView:: OnRDClick
----	-------------------

内容	virtual void OnRDClick(kn_int iScreenX, kn_int iScreenY);			
功能说明	子类实现鼠标在 view 上右键双击时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	iScreenX	kn_int	IN	鼠标右键双击时的 x 坐标（相对父 Group 的 x 坐标）
	iScreenY	kn_int	IN	鼠标右键双击时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.82 KView::onDownDirect

名称	KView:: onDownDirect			
内容	virtual void onDownDirect(kn_int iScreenX, kn_int iScreenY);			
功能说明	子类实现鼠标在 view 上左键下压时的功能处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	iScreenX	kn_int	IN	鼠标左键下压时的 x 坐标（相对父 Group 的 x 坐标）
	iScreenY	kn_int	IN	鼠标左键下压时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	DoMessage 时直接调用，不经信号槽传递，供子类控件内部使用			
线程安全	是			
示例代码	无			

5.83 KView::onMoveDirect

名称	KView::onMoveDirect			
内容	virtual void onMoveDirect(kn_int iScreenX, kn_int			

	<code>iScreenY);</code>			
功能说明	子类实现鼠标在 view 上移动时的功能处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	<code>iScreenX</code>	<code>kn_int</code>	IN	鼠标移动时的 x 坐标（相对父 Group 的 x 坐标）
	<code>iScreenY</code>	<code>kn_int</code>	IN	鼠标左移动时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	DoMessage 时直接调用，不经信号槽传递，供子类控件内部使用			
线程安全	是			
示例代码	无			

5.84 KView::onUpDirect

名称	<code>KView::onUpDirect</code>			
内容	<code>virtual void onUpDirect(kn_int iScreenX, kn_int iScreenY);</code>			
功能说明	子类实现鼠标在 view 上左键弹起时的功能处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	<code>iScreenX</code>	<code>kn_int</code>	IN	鼠标左键弹起时的 x 坐标（相对父 Group 的 x 坐标）
	<code>iScreenY</code>	<code>kn_int</code>	IN	鼠标左键弹起时的 y 坐标（相对父 Group 的 y 坐标）
返回值	无			
使用注意事项	DoMessage 时直接调用，不经信号槽传递，供子类控件内部使用			
线程安全	是			
示例代码	无			

5.85 KView::onWheelDirect

名称	<code>KView::onWheelDirect</code>			
内容	<code>virtual void onWheelDirect(kn_int iScreenX, kn_int iScreenY);</code>			

功能说明	子类实现鼠标在 view 上鼠标中键操作时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	pMsg	KMessageMouseWheel	IN	鼠标中键消息数据指针
返回值	无			
使用注意事项	DoMessage 时直接调用，不经信号槽传递，供子类控件内部使用			
线程安全	是			
示例代码	无			

5.86 KView::OnWheel

名称	KView::OnWheel			
内容	virtual void OnWheel(KMessageMouseWheel* pMsg);			
功能说明	子类实现鼠标在 view 上鼠标中键操作时的处理方法			
参数说明	名称	数据类型	输入输出	详细说明
	pMsg	KMessageMouseWheel	IN	鼠标中键消息数据指针
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.87 KView::OnUserMsg

名称	KView::OnUserMsg			
内容	virtual kn_bool OnUserMsg(KMessage* pMsg);			
功能说明	子类实现用户自定义消息的处理			
参数说明	名称	数据类型	输入输出	详细说明
	pMsg	KMessage	IN	传入用户自定义消息的数据指针
返回值	FALSE 不处理，TRUE 处理			

使用注意事项	无
线程安全	是
示例代码	无

5.88 KView::IsMousePicked

名称	KView::IsMousePicked			
内容	kn_bool IsMousePicked();			
功能说明	查看鼠标是否下压			
参数说明	名称	数据类型	输入输出	详细说明
返回值	true 鼠标是下压状态，false 鼠标不是下压状态			
使用注意事项	无			
线程安全	是			
示例代码				

5.89 KView::getRotateAngle

名称	KView:: getRotateAngle			
内容	REScalar getRotateAngle();			
功能说明	获取 view 的旋转角度			
参数说明	名称	数据类型	输入输出	详细说明
返回值	返回 view 的旋转角度			
使用注意事项	正数是顺时针，负数是逆时针			
线程安全	是			
示例代码				

5.90 KView::getRotatePoint

名称	KView::getRotatePoint			
内容	void getRotatePoint(REScalar& x, REScalar& y);			

功能说明	获取 view 旋转的参照点坐标			
参数说明	名称	数据类型	输入输出	详细说明
	x	REScalar	OUT	view 旋转参照点的 x 坐标
	y	REScalar	OUT	view 旋转参照点的 y 坐标
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.91 KStaticView::SetFontSize

名称	KStaticView:: SetFontSize			
内容	void SetFontSize(kn_int iFontSize);			
功能说明	设置显示的字符串字体大小			
参数说明	名称	数据类型	输入输出	详细说明
	iFontSize	kn_int	IN	字体大小
返回值	无			
使用注意事项	无			
线程安全	是			
示例代码	无			

5.92 KStaticView::setTextAlign

名称	KStaticView:: setTextAlign			
内容	void setTextAlign(REPaint::Align align);			
功能说明	设置字符串对齐方式			
参数说明	名称	数据类型	输入输出	详细说明
	align	REPaint::Align	IN	对齐方式，参见 3.10
返回值	无			

使用注意事项	无
线程安全	是
示例代码	<pre> KStaticView* pViewDate = new KStaticView; pViewDate->Create(0,0,100,100); pViewDate->SetText(szDate); pViewDate->SetFontSize(34); pViewDate->setTextAlign(REPaint::kRight_Align); </pre>

5.93 KStaticView::SetTextBound

名称	KStaticView:: SetTextBound			
内容	void SetTextBound(RERect rect);			
功能说明	设置字符串在 KStaticView 内的位置和大小			
参数说明	名称	数据类型	输入输出	详细说明
	rect	RERect	IN	字符串占据的矩形
返回值	无			
使用注意事项	矩形位置为相对 KStaticView 的位置			
线程安全	是			
示例代码	无			

5.94 KStaticView::GetTextBound

名称	KStaticView:: GetTextBound			
内容	RERect GetTextBound();			
功能说明	获取字符串在 KStaticView 内的位置和大小			
参数说明	名称	数据类型	输入输出	详细说明
	无			
返回值	RERect: 字符串在 KStaticView 内占据的矩形			
使用注意事项	矩形位置为相对 KStaticView 的位置			
线程安全	是			

示例代码	无
------	---

6 数据结构

6.1 KDrawable_PTR

结构名或类名	KDrawable_PTR		
说明	typedef boost::shared_ptr<KDrawable> KDrawable_PTR; KDrawable 的智能指针形式		
数据成员	名称	数据类型	详细说明
	略		
使用注意事项	无		

6.2 KColorDrawable_PTR

结构名或类名	KColorDrawable_PTR		
说明	typedef boost::shared_ptr<KColorDrawable> KColorDrawable_PTR; KColorDrawable 的智能指针形式		
数据成员	名称	数据类型	详细说明
	略		
使用注意事项	无		

6.3 KImageDrawable_PTR

结构名或类名	KImageDrawable_PTR		
说明	typedef boost::shared_ptr<KImageDrawable> KImageDrawable_PTR; KImageDrawable 的智能指针形式		
数据成员	名称	数据类型	详细说明
	略		
使用注意事项	无		

6.4 KTextDrawable_PTR

结构名或类名	KTextDrawable_PTR		
说明	typedef boost::shared_ptr<KTextDrawable> KTextDrawable_PTR; KTextDrawable 的智能指针形式		
数据成员	名称	数据类型	详细说明
	略		
使用注意事项	无		

6.5 KShapeDrawable_PTR

结构名或类名	KShapeDrawable_PTR		
说明	typedef boost::shared_ptr<KShapeDrawable> KShapeDrawable_PTR; KshapeDrawable 的智能指针形式		
数据成员	名称	数据类型	详细说明
	略		
使用注意事项	无		

6.6 VEC_DRAWABLE

结构名或类名	VEC_DRAWABLE		
说明	typedef vector<KDrawable_PTR> VEC_DRAWABLE; KDrawable 的智能指针列表		
数据成员	名称	数据类型	详细说明
	略		
使用注意事项	无		

6.7 LoopType

结构名或类名	LoopType
说明	动画循环类别枚举

数据成员	名称	数据类型	详细说明
	LOOP_TIME		反复循环
	LOOP_PINGPONG		从头至尾，再从尾至头反复循环
使用注意事项	无		

6.8 KVIEWTYPE

结构名或类名	KVIEWTYPE		
说明	View 类型枚举		
数据成员	名称	数据类型	详细说明
	KVIEW_BASE		
	KVIEW_WINDOW		
	KVIEW_BUTTON		
	KVIEW_TEXT		
	KVIEW_LIST		
	KVIEW_CHECKBOX		
	KVIEW_RADIOBOX		
	KVIEW_SLIDER		
	KVIEW_PROGRESS		
	KVIEW_IMAGECTRL		
	KVIEW_MOVEABLE		
	KVIEW_STATIC		
	KVIEW_RADIOBOXGROUP		
	KVIEW_GALLERY		
	KVIEW_GALLERY_ITEM		
	KVIEW_LIST_ITEM		
使用注意事项	无		

6.9 SkPoint

结构名或类名	SkPoint
说明	点信息

数据成员	名称	数据类型	详细说明
	fX	kn_int	
	fY	Kn_int	
使用注意事项	无		

6.10 Align

结构名或类名	Align		
说明	对齐方式枚举		
数据成员	名称	数据类型	详细说明
	kLeft_Align		左对齐
	kCenter_Align		居中对齐
	kRight_Align		右对齐
	kAlignCount		对齐方式总数
使用注意事项	无		

6.11 ViewState

结构名或类名	ViewState		
说明	聚焦状态枚举		
数据成员	名称	数据类型	详细说明
	BS_NORMAL		正常状态
	BS_FOCUS		聚焦状态
	BS_PRESSED		按下状态
	BS_DISABLED		失效状态
使用注意事项	无		