

FTK中的动画机制

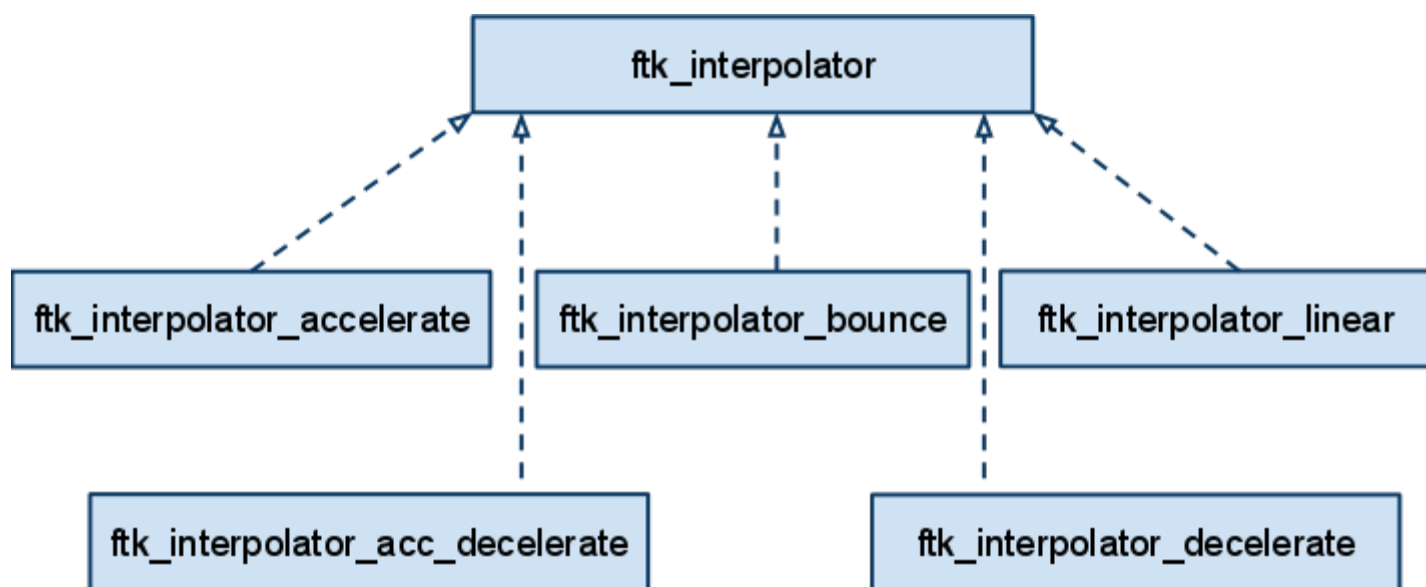
虽然FTK很早就支持了窗口动画，但是作为动画的实现者，我不得不承认FTK的0.6版本之前的动画实现非常蹩脚。主要原因有几点：

1. 动画运动的速度是单一的匀速运动。
2. 动画的帧数不能随着硬件性能而改变。
3. 动画只有窗口的入场(显示)动画，而没有窗口的出场(关闭)动画。
4. 动画机制没有集成到GUI中，必须由使用者创建和调用，使用起来比较麻烦。

前段时间因为工作需要花了不少时间去研究Android的GUI的实现，特意关注了它的动画实现，从中受到不少启发。最近重新实现了FTK的动画机制，很大程度上借鉴了Android的动画原理，当然具体实现上是有很大大差异的。这里简单介绍一下FTK的窗口动画实现：

动画运行时间和动画运行进度之前的关系

如果动画匀速运动，那么时间和位置是线性关系的，比如时间过去50%，那么动画会运行到50%的位置上。但是实现中的物体运动，可能是加速运动，减速运动或皮球一样反弹的，如果所有窗口动画都使用匀速运动，很难达到理想的动画效果。为了解决这个问题，我们引入一个插值器(interpolator)来实现时间和位置之间的变换。



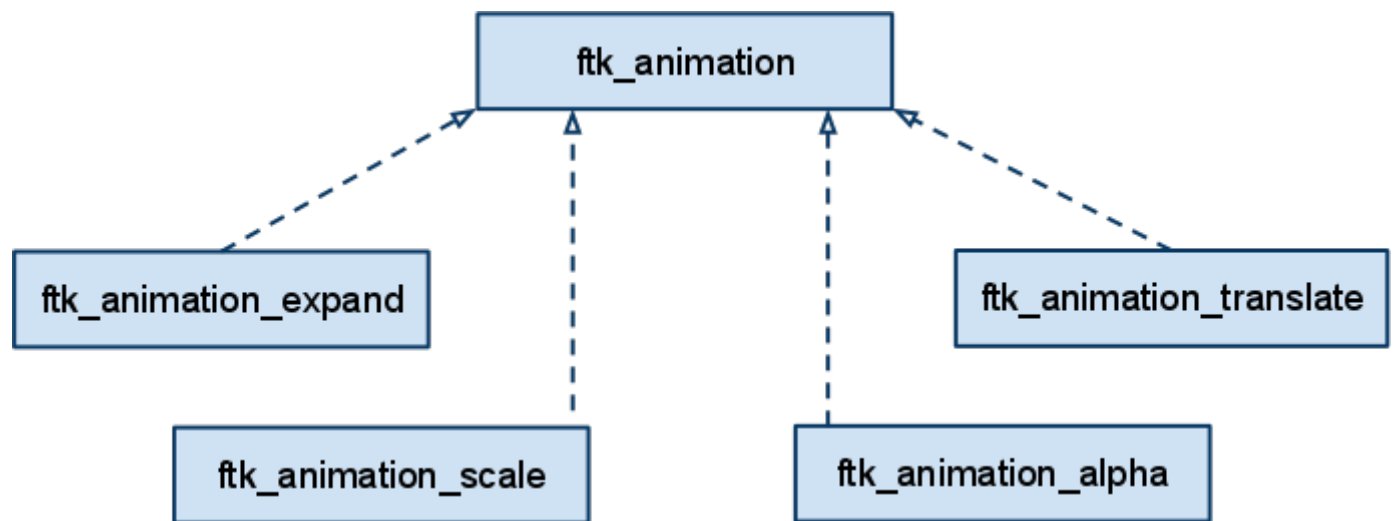
FtkInterpolator接口的主要函数如下：

```
typedef float (*FtkInterpolatorGet)(FtkInterpolator* this, float percent);
```

本函数用于通过时间百分比获取动画百分比。百分比用小数表示，范围在0到1之间。

动画的各种实现

FTK中的动画总体上就是对两个Bitmap进行各种变换，然后显示到屏幕上，常见的动画包括移动，缩放，旋转，扩展和改变透明度等。为了让调用者不关心具体的实现方式和以后的扩展，我们抽象出FtkAnimation接口。



FtkAnimation接口的主要函数如下：

```
typedef Ret (*FtkAnimationStep)(FtkAnimation* this);
```

本函数用于根据当前的进度绘一帧动画，返回值不为RET_OK，则应该结束动画。

```
typedef Ret (*FtkAnimationReset)(FtkAnimation* this, FtkBitmap* old_win, FtkBitmap* new_win,
    FtkRect* old_win_rect, FtkRect* new_win_rect);
```

本函数用于初始化FtkAnimation，因为多个窗口是可以共用一个FtkAnimation对象的，所以每次动画前应该重新初始化。

动画的参数

公共参数：

1. src 表示动画操作的图片，因为窗口动画通常是在窗口切换进行的，我们把新旧两个窗口的图片用于动画变换。可选值有：old_window和new_window，以及两者的组合。
2. interpolator 表示使用何种插值器，可选值有：linear bounce accelerate decelerate

acc_decelerate等。

3. from 表示动画的起始位置，为0到1之间的小数。
4. to 表示动画的结束位置，为0到1之间的小数。
5. duration 动画持续的时间，单位为ms。

注意：在不同的动画中，from和to对应不同的东西，比如alpha动画中对应alpha的值，起始值为from * 0xff，结束值为 to * 0xff。

gtk_animation_expand

1. base: 表示以哪里为基准进行扩展。可选取值有：left right top bottom center
2. alpha 表示是否同时启用alpha效果。
3. push 表示扩展时是否推动另外一个窗口。

gtk_animation_alpha 无特殊参数。

gtk_animation_scale

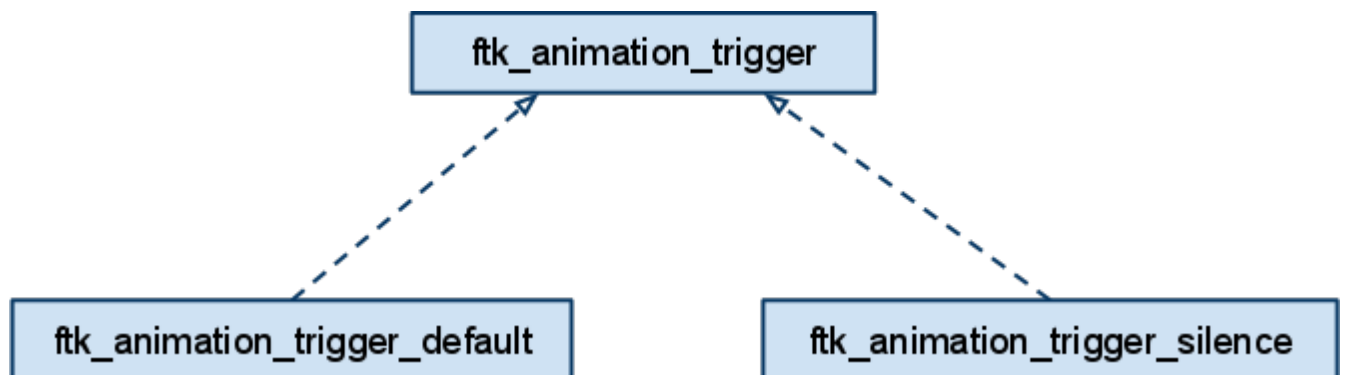
1. alpha 表示是否同时启用alpha效果。

gtk_animation_translate 无特殊参数。

动画的触发

窗口动画主要在窗口显示和隐藏（关闭）时运行的，不同窗口需要不同的动画效果，比如对话框和普通窗口是不一样的。我们先引入一个动画触发器，根据窗口的显示和隐藏（关闭）事件和窗口的类型，触发相应的动画事件。

考虑到低端硬件平台使用动画效果有些困难，所以动画触发器有两种实现，一种启用动画，一种不启用动画。



动画在窗口管理器中触发 (gtk_wnd_manager_default.c) :

```

case FTK_EVT_HIDE:
{
priv->disable_anim++;
ftk_wnd_manager_default_emit_top_wnd_changed(thiz);
priv->disable_anim--;
ftk_wnd_manager_default_do_animation(thiz, event);
ftk_wnd_manager_update(thiz);
return RET_OK;
}
case FTK_EVT_SHOW:
{
priv->disable_anim++;
ftk_wnd_manager_default_emit_top_wnd_changed(thiz);
priv->disable_anim--;
ftk_wnd_manager_default_do_animation(thiz, event);
return RET_OK;
}

```

动画的描述

为了更灵活的描述动画效果，我们选择用XML来描述动画效果，而不是写在代码中。动画首先是与主题关联起来的，不同的主题可以选择是否打开动画，以及使用何种动画效果。

在theme.xml中，我们加了一个新的配置：

```
<animation_trigger name="animations.xml" />
```

如果指定了动画的配置文件名则启用动画，否则不启用动画。

具体的动画描述放在animations.xml中，格式很简单，基本上就是事件+对应的动画描述。动画描述又包括插值器名称和其它参数。参数可以包括一些变量，这些变量在运行时才决定。下面是一个示例：

```

<animations>
<event name="menu_show">
    <expand base="bottom" alpha="enable" src="new_window" interpolator="decelerate"
from="0.1" to="1.0" duration="150"/>
</event>

<event name="menu_hide">
    <expand base="bottom" alpha="enable" src="old_window" interpolator="accelerate"
from="1.0" to="0.1" duration="150"/>
</event>

```

```
<event name="msgbox_show">
    <scale src="new_window" alpha="enable" interpolator="bounce" from="0.5" to="1.0"
duration="200"/>
</event>

<event name="msgbox_hide">
    <scale src="old_window" alpha="enable" interpolator="accelerate" from="1.0" to="0.5"
duration="200"/>
</event>

<event name="dialog_show">
    <scale src="new_window" alpha="enable" interpolator="decelerate" from="0.5" to="1.0"
duration="200"/>
</event>

<event name="dialog_hide">
    <scale src="old_window" alpha="enable" interpolator="accelerate" from="1.0" to="0.5"
duration="200"/>
</event>
<event name="app_window_show">
    <translate src="old_window|new_window" interpolator="decelerate"
dir="h" from="0" to="0.5" duration="300"/>
</event>

<event name="app_window_hide">
    <translate src="new_window|old_window" interpolator="decelerate"
dir="h" from="0.5" to="0" duration="300"/>
</event>

<event name="app_main_window_show">
    <expand src="new_window" interpolator="decelerate"
base="center" alpha="enable" from="0.2" to="1.0" duration="300"/>
</event>

<event name="app_main_window_hide">
    <expand src="old_window" interpolator="accelerate"
base="center" alpha="enable" from="1.0" to="0.5" duration="300"/>
</event>

<event name="misc_window_show">
    <alpha src="new_window" interpolator="linear"
from="0" to="1.0" duration="300"/>
</event>
```

```
<event name="misc_window_hide">
  <alpha src="old_window" interpolator="linear"
    from="1.0" to="0" duration="300"/>
</event>
<event name="combobox_up_show">
  <expand src="new_window" interpolator="decelerate"
    base="bottom" from="0.5" to="1.0" duration="200"/>
</event>

<event name="combobox_up_hide">
  <alpha src="old_window" interpolator="accelerate"
    from="1.0" to="0" duration="200"/>
</event>

<event name="combobox_down_show">
  <expand src="new_window" interpolator="decelerate"
    base="top" from="0.5" to="1.0" duration="200"/>
</event>

<event name="combobox_down_hide">
  <alpha src="old_window" interpolator="accelerate"
    from="1.0" to="0" duration="200"/>
</event>

<event name="applist_v_show">
  <expand src="new_window" interpolator="decelerate"
    base="bottom" push="true" from="0.1" to="1.0" duration="200"/>
</event>
<event name="applist_v_hide">
  <expand src="old_window" interpolator="accelerate"
    base="bottom" from="1.0" to="0.1" duration="200"/>
</event>

<event name="applist_h_show">
  <expand src="new_window" interpolator="decelerate"
    base="left" push="true" from="0.1" to="1.0" duration="200"/>
</event>

<event name="applist_h_hide">
  <expand src="old_window" interpolator="accelerate"
    base="left" from="1.0" to="0.1" duration="200"/>
</event>
```

```
<event name="im_preeditor_show">
    <alpha src="new_window" interpolator="linear"
        from="0" to="1.0" duration="300"/>
</event>

<event name="im_preeditor_hide">
    <alpha src="old_window" interpolator="linear"
        from="1.0" to="0" duration="300"/>
</event>

</animations>
```

窗口特有的动画

```
Ret    ftk_window_set_animation_hint(FtkWidget* thiz, const char* hint);
```

通过ftk_window_set_animation_hint可以为窗口指定动画，动画的描述必须在animations.xml中存在。比如hint为im_preeditor，那么在配置文件中必须有im_preeditor_show和im_preeditor_hide的描述。