特效(非托管)

# 概述:

用户应该能够在两张照片之间添加特效.目前,我们只打算提供两个简单的特效:淡出特效和飞入特效.将来我们可能会扩展特效库.因此,我们需要设计特效功能的接口.

此文档仅包含非托管部分. 参见托管部分接口设计的规格文档.

# 期望的结果:

* 为非托管组件创建和特效库交互的接口.
* 使每个特效不使用WIC和Media Foundation技术实现特效逻辑.

# 不期望的结果:

* 创建 plug-in 模型.
* 调用实际特效库.

# 设计:

在本机部分的某些属性/事件/方法相对简单, 因为它并不需要与UI交互. 考虑到C ++不支持接口, 我们没有使用任何接口, 仅提供了一个抽象基类. 是否要删除接口和仅使用抽象基类有待商榷. 此类被命名为TransitionBase，设计如下.

class TransitionBase

{

public:

int GetTransitionDuration();

void SetTransitionDuration(int value);

BYTE\* GetForegroundFrame();

void SetForegroundFrame(BYTE\* value);

BYTE\* GetBackgroundFrame();

void SetBackgroundFrame(BYTE\* value);

int GetFrameSize();

int GetFrameWidth();

void SetFrameWidth(int value);

int GetFrameHeight();

void SetFrameHeight(int value);

virtual BYTE\* GetOutputFrame(float time) = 0;

virtual void ParseXml(XmlParser\* pParser);

};

## 常用属性

某些托管接口中的属性/事件/方法不需要存在于对应的非托管接口中.相反,非托管基类需要每个视频帧大小的信息.当前,我们在调用组件时硬性定为800\*600,但是将来可能可以改变,所以必须提供接口.每个特效还需要提供一个输出帧.

## 反序列化

在非托管部分,我们只需要自xml反序列化特效. 我们在TransitionBase类提供了常用预制逻辑, 并允许子类重写它.