Manuál ke zdrojovému kódu aplikace Interlacer

Obsah

1 Úvod

Tento dokument slouží pro snadnější orientaci ve zdrojovém kódu jako celku. Obsahuje stručný popis některých důležitých tříd a popis souvislostí mezi jednotlivými častmi aplikace.

Dále také obsahuje návod na použití prokládacího mechanismu jako samostatné části pro vytvoření výstupního lentikulárního obrázku.

Podrobnější popis jednotlivých tříd, jejich atributů a metod se nachází přímo ve zdrojovém kódu.

2 GfxlibWrapper

GfxlibWrapper je skupina tříd, které se starají o komunikace s knihovnou Magick++. Jsou v projektu v samostatné složce a ke svojí plné funkčnosti potřebují, aby se v adresáři odkud je aplikace spuštěna, nacházel soubor Gfxlib.dll, jehož zdrojové kódy v C++ se nacházejí v odděleném projektu Gfxlib.

2.1 Třída Picture

Picture je třída pro reprezentaci jednotlivých obrázků. Instanci této třídy lze vytvořit dvěma konstruktory:

```
public Picture(int width, int height)
```

- vytvoří obrázek o zadané šířce a výšce

```
public Picture(String filename)
```

- vytvoří instanci třídy a nastaví, z jakého souboru má být obrázek načten, samotné načtení ale provedeno zatím není

Pro načtení obrázku je pak potřeba použít metodu Load, která načte obrázek ze souboru, jehož název byl předán právě tímto konstruktorem. Lze také použít metodu Ping, která pouze načte informace o obrázku (šířku a výšku v px, rozlišení…) ale samotná obrazová data načtena nejsou.

Infromaci, zda byl obrázek již vytvořen, ať už prvním konstruktorem nebo načtením metodou Load (ne Ping) lze zjistit metodou IsCreated.

Po skončení práce s obrázkem je dobré zavolat metodu Destroy, která provede dealokaci paměti. Metoda Destroy se sice volá sama v destruktoru, ale ten je volán garbage collectorem a to se může stát až v případě, kdy na instanci neexistují reference.

K uložení slouží metoda Save, které stačí parametrem předat název nebo celou cestu výsledného souboru. Formát uloženého souboru je rozpoznán z koncovky. Například koncovka .jpg zajistí uložení do formátu JPEG.

2.2 Filtry pro změnu velikosti

Třída FilterType obsahuje 4 statické instance sama sebe. Každná z těchto instancí reprezentuje jeden typ interpolačního filteru, který lze použí pro změnu velikosti obrázku metodou Resize.

Filtery lze použít například takto:

```
Picture picture = new ...;
picture.Resize(500, 400, FilterType.Cubic);
```

Tento kód změní velikost obrázku z původní na 500x400 pixelů s použitím kubické interpolace.

2.3 Jednotky

Výčtový typ Units obsahuje tři prvky, každý z nich reprezentuje jiné jednotky.
In - palce
Cm - centimetry
Mm - milimetry

GfxlibWrapper také obsahuje statickou třídu UnitConverter (v souboru Units.cs), která obsahuje několik metod pro převody jednotek.

3 Prokládání

3.1 Třída PictureContainer

O samotné prokládání se stará třída PictureContainer. Nejdříve je potřeba vytvořit její instanci, třída má pouze jeden konstruktor.

```
public PictureContainer(List<Picture> pictures,
InterlacingData interlacingData, LineData lineData,
ProgressBar progressBar = null)
```

Prvním parametrem je seznam obrázků k proložení v podobě instance třídy List, kde každý obrázek je reprezentován instancí třídy Picture z GfxlibWrapperu.

Druhým parametrem je instance třídy InterlacingData, která obsahuje všechny parametry prokládání, viz níže.

Třetím parametrem je instance třídy LineData, která obsahuje všechny parametry pasovacích značek, viz níže.

Posledním parametrem je instance třídy System.Windows.Forms.ProgressBar, která je pak použita pro zobrazení postupu prokládání. Pokud je progressBar nastaven na null, postup nebude ukázán nikde, ale prokládání bude fungovat.

Před samotným proložením je potřeba zavolat metodu CheckPictures, pokud nebude zavolána, prokládání selže a vyhodí výjimku. Tato metoda zkontroluje velikost všech obrázků, pokud bude u všech stejná, vrátí true a při prokládání budou obrázky proloženy tak, jak jsou. Pokud obrázky nebudou stejně velké, vrátí false a při prokládání pak budou všechny obrázky oříznuty na šířku nejužšího a výšku nejnižšího.

Pro proložení pak stačí zavolat metodu Interlace a poté metodou GetResult získat výsledný obrázek v podobě instance třídy Picture.

Obrázky, které již byly načteny před prokládáním nebo byly vytvořeny prvním konstrutkorem (tedy vždy, když metoda IsCreated vrací true) budou rovnou použity. Obrázky, které byly vytvořeny druhým konstruktorem a před prokládáním nebyly načteny (tedy ty, jejichž metoda IsCreated vrací false) budou načteny při prokládání a po něm rovnou smazány metodou Destroy.

3.2 Třída InterlacingData

Tato třída obsahuje všechny parametry prokládání (šířka, výška a rozlišení výstupního obrázku, informace o použitých jednotkách, hustota čoček lentikulární desky, interpolační filtry pro obě fáze změny velikosti a směr prokládání) a její instance je pro proložení potřeba. Obsahuje pouze atributy a settery a gettery. Jednotky pro rozměry je možné změnit metodou SetUnits, jednotky pro rozlišení a hustotu čoček je možné změnit metodou SetResolutionUnits. Obě metody rovnou přepočítají aktuální hodnoty. Jakákoliv hodnota nastavená kterýmkoliv setterm se předpokládá v jednotkách, které jsou aktuálně nastaveny. Metodou KeepAspectRatio lze vypnout nebo zapnout funkci pro zachování poměru stran. Pokud je zapnuta, pak při změně šířky nebo výšky je automaticky přepočítán i druhý rozměr. Před použitím instance pro proložení je potřeba pomocí setterů nastavit všechny parametry.

3.3 Třída LineData

Tato třída obsahuje všechny parametry pasovacích značek (šířku a barvu čar, barvu pozadí, šířku rámečku s čarami, odsazení, informace na kterých okrajích čáry mají být, informaci zda mají být zarovnány na střed nebo ke kraji) a její instance je pro proložení potřeba. Obsahuje pouze atributy a settery a gettery. Jednotky pro šířku ramečku a odsazení je možné změnit metodou SetUnits, která obě hodnoty rovnou přepočítá. Jakákoliv hodnota nastavená kterýmkoliv setterm se předpokládá v jednotkách, které jsou aktuálně nastaveny. Před použitím instance pro proložení je potřeba pomocí setterů nastavit všechny parametry.

4 Propojení s GUI

Hlavní třídou GUI je třída MainForm. Ta je rozdělena do dvou souborů MainForm.cs a MainFormEvents.cs.

Prokládání se nachází v metodě události na stisknutí tlačítka pro proložení a to včetně vytvoření instance třídy PictureContainer.

Třída MainForm obsahuje instanci třídy ProjectData, která v sobě má instance tříd InterlacingData a LineData, stará se také o načítání a ukládání projektů.

Po jakékoliv změně komponent formuláře jsou rovnou nové hodnoty zaneseny do InterlacingData a LineData v instanci třídy ProjectData. Zároveň je zavolána metoda UpdateAllComponents, která zajistí, že jsou do všech komponent dodány aktuální hodnoty z InterlacingData a LineData.

Třída MainForm obsahuje instanci třídy Settings, která obsahuje informaci o aktuálním nastavení aplikace. Informace o tom, co vše lze nastavit je předávána prostřednictvím instance třídy SettingOptions. Třída SettingOptions obsahuje pouze tři Listy, první obsahuje nastavitelné jazyky, druhý nastavitelné délkové jednotky a třetí nastavitelné jednotky rozlišení. Všechny tyto položky jsou reprezentovány instancemi třídy StringValuePair, která slouží k tomu, že vytvoří dvojici (Jméno, Objekt), kde jméno je typu String a je vraceno metodou toString, Objekt je pak libovolný objekt, který má být takto pojmenován. Tyto dvojice v podobě instancí třídy StringValuePair jsou pak vkládány do ComboBoxů formuláře a při změně jazyka se vytvoří dvojice nové s pojmenováním v daném jazyce a jsou dodány do ComboBoxů. O vy-

tvoření těchto dvojic se stará metoda createSettingOptions, která je vrací v podobě instance třídy SettingOptions. Metoda ApplySettings zařídí promítnutí změn v instanci třídy Settings do formuláře.

Třída MainForm obsahuje instanci třídy PictureInfoData, která se stará o získávání informací o obrázcích kvůli zobrazení v dolním pravém rohu formuláře. Informace o obrázcích si uchovává, aby nebylo nutné stejný obrázek načítat vícekrát. O samotné zobrazení těchto informací se stará metoda setPictureInfo, metoda resetPictureInfo pak nastaví potřebné labely na prázdné řetězce.

Třída MainForm obsahuje instanci třídy PreviewData, která se stará o zobrazování náhledů. Uchovává si načtené obrázky, aby nebylo potřeba načítat stejný obrázek vícekrát. V třídě MainForm je zobrazení náhledu řešeno v metodě setPreview.

5 Lokalizace aplikace

Aplikace je přeložena do dvou jazyků. Do češtiny a angličtiny. Nastavení jazyka je možno provést v aplikaci v menu `nastavení`. Změna jazyka se projeví ihned po stisknutí tlačítka `OK` nebo tlačítka `apply`. Nastavení jazyka se po ukončení aplikace uloží do souboru a při opětovném spuštění si aplikace nastavení jazyka z tohoto souboru načte.

5.1 Přidání textu komponentě

Dříve, než přidáváme jakoukoliv novou komponentu do aplikace nebo upravujeme text již vytvořené komponenty, ujistíme se, že formuláři, do kterého komponentu přidáváme, je nastavena hodnota Language na `default`. To znamená, že formulář je právě nastaven na anglický jazyk a veškeré texty komponent budeme zadávat v anglickém jazyce. Po zadání textu komponentě se její text včetně jejích atributů ihned uloží do resource souboru `MainForm.resx`. Pro překlad textu komponenty do českého jazyka přepneme formuláři nastavení hodnoty Language na `Czech (Czech republic)`, nikoliv pouze `Czech`. Stejným způsobem přidáme text komponentě, nyní v českém jazyce. Text se uloží do resource souboru `MainForm.cs-CZ.resx`.

5.2 Přidání jiného textu

Pokud přidáváme jakýkoliv jiný text (výpis chybové hlášky, tooltip komponenty, položky comboboxů...), musíme dané texty přidat do resource souborů

manuálně. Pro každý jazyk je vytvořen zvláštní resource soubor oddělený od hlavního souboru s texty komponent. Pro češtinu se nazývá `StringRes_CZ.resx` a pro angličtinu `StringRes_EN.resx`. Pokud tedy například přidáváme novou chybovou hlášku, v resource souborech jí přidělíme nějaké jméno (sloupec `Name`), pod kterým k ní budeme přistupovat a do sloupce `Value` napíšeme v odpovídajícím jazyce její text. V kodu budeme k těmto textům přistupovat pomocí statické třídy `Localization`. V této třídě je definovaný ComponentResourceManager nazvaný `resourcesStrings` pomocí něhož budeme přistupovat k těmto resource souborům. Jsou tam definovány ještě dva další ComponentResourceManagery, každý pro správu jednoho formuláře.

Příklad přístupu k textu pomocí jeho jména:

```
Localization.resourcesStrings.GetString("Name")
```