

Praktikum 1. Juni 2011

Jörn Loviscach

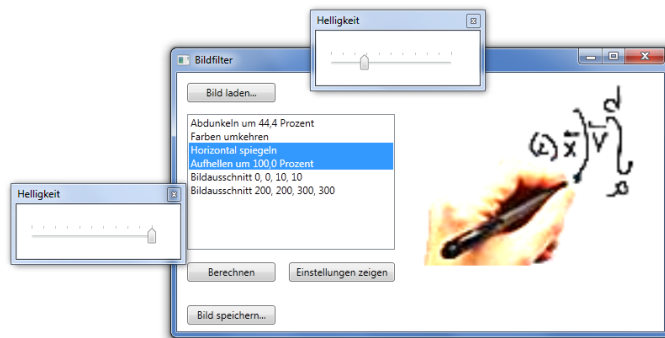
Versionsstand: 31. Mai 2011, 19:46



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Fortsetzung des vorigen Praktikums: ein Programm, das Filter auf ein Bild anwendet.

Beenden Sie die Aufgabe aus dem vorigen Praktikum. Danach soll die vorgegebene Klasse `Info2Praktikum.Bitmap` um Funktionen zum Schreiben und Lesen in Klartext erweitert werden. Außerdem sollen Filter die Möglichkeit haben, Fenster mit Reglern zum Einstellen anzuzeigen.



Bilder als Klartext schreiben und lesen. Erweitern Sie die vorgegebene Klasse `Info2Praktikum.Bitmap` um eine Methode zum Schreiben der Bilddaten und eine Methode zum Lesen der Bilddaten. (Sollte eine dieser beiden Methoden statisch sein?) Mit Hilfe von `System.IO.FileStream`, `System.IO.StreamWriter` und `System.IO.StreamReader` soll hierfür ein Format gemäß diesem Muster verarbeitet werden:

```
110
83
26 26 26 25 ... 23
255 0 0 0 0 ... 0
...
```

In der ersten Zeile steht dabei die Breite des Bilds, in der zweiten Zeile die Höhe; dann folgen Textzeile um Textzeile die Zeilen des Bilds, jeweils die RGB-Werte

aller Pixel der Zeile mit Leerzeichen hintereinander. (Ein solches Format ist zum Beispiel zum Debuggen sinnvoll – oder dazu, Bildfehler mit einem Texteditor sehr präzise zu beheben.)

Im Hauptfenster soll es nun folgende drei Schaltflächen geben:

- Die erste Schaltfläche lädt Bilder in den üblichen Formaten, aber auch diesem neuen Format mit der Extension `.textbitmap`. Mit Hilfe von `System.IO.FileInfo` wird dazu die Extension der ausgewählten Datei bestimmt und im Programm passend verzweigt.
- Die zweite Schaltfläche führt die in der Listbox ausgewählten Filter aus und weist das Ergebnis der Bildfläche im Fenster zu.
- Die dritte Schaltfläche speichert das Ergebnis – wenn schon eines vorhanden ist – im neuen `.textbitmap`-Format.

Einstellungsfenster für die Filter. Ergänzen Sie die `Filter`-Klasse um virtuelle Methoden zum Anzeigen bzw. Schließen von Einstellungsfenstern. Sehen Sie eine vierte Schaltfläche im Hauptfenster vor, die von allen Filtern die Methode zum Anzeigen der Einstellungsfenster aufruft. Wenn das `Closing`-Ereignis des Hauptfensters eintritt, rufen Sie für jedes Filter die Methode zum Schließen des Einstellungsfenster auf.

Für das Filter, das die Helligkeit einstellt, sollen diese beiden Methoden tatsächlich ein Fenster anzeigen bzw. schließen. (Die übrigen Filter könnte man danach auch noch mit Fenstern versorgen – wenn es denn beim jeweiligen Filter überhaupt etwa einzustellen gibt.)

Machen Sie die Helligkeitseinstellung dieses Filters als Property les- und schreibbar. Fügen Sie zum Projekt eine neue WPF-Fenster-Klasse hinzu. Setzen Sie einen Schieberegler mit passendem Minimumwert und Maximumwert in dieses Fenster. Geben Sie dem Konstruktor dieses Fensters einen Verweis auf das Filter mit, damit das Fenster zu Beginn den Helligkeitwert holen kann und beim `ValueChanged`-Ereignis des Schiebereglers den Wert setzen kann. Damit die neue Fenster-Klasse und die Filter-Klassen zusammenspielen, muss man die Filter-Klassen ausnahmsweise `public` machen.

Wenn der Anwender ein Fenster per Mausklick auf das X im Fensterrahmen schließt, lässt sich das Fenster leider nicht mehr mit `Show` wieder öffnen, sondern muss zuerst per Konstruktor neu erzeugt werden. Das erkennt man daran, dass die Property `IsLoaded` dann `false` ist.

Die neuen Helligkeitseinstellungen sollten auch sofort in der Listbox sichtbar werden. Rufen Sie dazu auf geeignete Weise und zu passender Zeit die Methode `Refresh` der `Items`-Sammlung der Listbox auf.

Zusatzaufgabe für Fortgeschrittene. Die `SelectedItem`-Sammlung der Listbox liefert die Einträge in der Reihenfolge, in der man sie angeklickt hat, also nicht immer in der Reihenfolge der Liste. Sorgen Sie dafür, dass die Effekte trotzdem in der Reihenfolge ausgeführt werden, in der sie in der Liste stehen.