

Abschlussprojekt des Praktikums

Die nachfolgende Aufgabenstellung ist das Abschlussprojekt des Praktikums. Es weist einen größeren Umfang als die bisherigen Praktikumsübungen auf. Es wird daher in Übung 8 (Aufgaben 1 und 2) und Übung 9 (Aufgabe 3) bearbeitet.

In diesem Projekt sollen Sie zeigen, dass Sie eine Klasse selber entwerfen und vorhandene Klassen professionell einsetzen können.

Aufgabenstellung

Es soll eine Klasse `AVI_Diashow` erstellt werden, mit deren Hilfe aus BMP-Bildern eine Diashow als AVI-Film erzeugt werden kann. Aus jedem Bild wird ein Clip mit vorgebbbarer Länge erzeugt. Die Clips werden dann zu einem AVI-Film zusammengebaut.

Die Klasse `AVI_Diashow` besitzt folgende öffentlichen Komponenten und Methoden:

`enum Ueberblendung { Soft, Schieben }:` Dies ist ein Datentyp, der einen besonderen Übergang zwischen zwei Clips im AVI-Film spezifiziert. Der Wert `Soft` soll ein weiches Überblenden zwischen Clips vorgeben, der Wert `Schieben` bewirkt, dass das neue Bild das alte Bild nach rechts wegschiebt.

`AVI_Diashow(const string& AviName, int Breite, int Hoehe):` Beim Konstruktor wird der Name der zu erzeugenden AVI-Datei sowie die Breite und Höhe der Bilder im AVI-Film angegeben.

`int Clip_hinzufuegen(const string& name, int Laenge):` Mit dieser Methode wird ein Eintrag im `AVI_Diashow`-Objekt gespeichert, der einen Clip der Show beschreibt. Der Parameter `name` gibt den Namen einer BMP-Bilddatei an. Der Wert `Laenge` spezifiziert die Länge (Anzahl von Bildern) des zu erzeugenden Clips. Der `int`-Rückgabewert ist ein Index des Clips, unter dem er in den folgenden Methoden `Ueberblendung_hinzufuegen` und `Grafik_hinzufuegen` angesprochen werden kann.

`bool Ueberblendung_hinzufuegen(int Clip, Ueberblendung ue, int Laenge):` Diese Methode spezifiziert den Übergang zwischen den beiden Clips mit Index `Clip` und `Clip+1`. Die Art der Überblendung wird durch den Wert `ue` vom Aufzählungstyp `Ueberblendung` spezifiziert. Wird `Soft` angegeben, wird der erste Clip sanft in den zweiten Clip überführt. Wird `Schieben` angegeben, schiebt der neue Clip den alten Clip nach rechts aus dem Bild heraus. Die Anzahl der Bilder im Clip, welche zur Überblendung verwendet werden, ist durch den Parameter `Laenge` bestimmt. Die Art der Überblendung wird im `AVI_Diashow`-Objekt gespeichert.

Der Rückgabewert gibt an, ob die Information dem Clip hinzugefügt werden konnte.

`bool Grafik_hinzufuegen(int Clip, const GrafikElement& g):` Mit dieser Methode können dem Clip `GrafikElement`-Objekte hinzugefügt werden, die später beim Erzeugen des AVI-Films über den Clip gezeichnet werden. Die Methode kann für einen Clip mehrfach aufgerufen werden, alle Grafik-Objekte müssen gespeichert und später gezeichnet werden.

Der Rückgabewert gibt an, ob die Information dem Clip hinzugefügt werden konnte.

`int Laenge_des_Films() const:` Mit dieser Methode kann die Länge des zu erstellenden AVI-Films abgefragt werden. Dabei muss berücksichtigt werden, dass sich

die Clips bei Überblendungen überlappen. Somit können nicht einfach die Längen der Clips addiert werden.

`void Film_erstellen()` const: Nachdem alle gewünschten Clips und Überblendungen spezifiziert sind, kann mit dieser Methode der AVI-Film erstellt werden. Wenn für den Übergang zwischen zwei Clips eine Überblendung spezifiziert ist, muss diese angewendet werden. Ansonsten wird hart vom ersten auf den zweiten Clip umgeschaltet.

Erst in dieser Methode sollen die BMP-Bilddateien geöffnet und zu realen Clips zusammengesetzt werden. Alle vorherigen Methoden sollen nur die Informationen über die Clips und Überblendungen speichern.

Aufgabe 1: Testversion der Klasse `AVI_Diashow` (4 Punkte)

Erstellen Sie ein Konzept, wie die Informationen der Clips sowie der zugehörigen Grafiken und Überblendungen in Ihrer Klasse `AVI_Diashow` abgelegt werden. Verwenden Sie wo sinnvoll immer Container der Standardbibliothek.

Arbeiten Sie Ihr Konzept so aus, dass Sie es Interessenten anschaulich vorstellen können.

Aufgabe 2: Testversion der Klasse `AVI_Diashow` (6 Punkte)

Erstellen Sie eine erste Version der Klasse, in der alle Methoden außer `Film_erstellen` in ihrer endgültigen Form enthalten sind. Testen Sie die Klasse mit dem Programm „`Test_AVI_Diashow.cxx`“.

Die Methode `Film_erstellen` erzeugt noch keinen Film sondern gibt Informationen auf der Konsole aus, wie der Film zusammengesetzt ist.

Überprüfen Sie in dieser Methode die Konsistenz der Beschreibung. Z.B. müssen die Längen der Clips und der Überblendungen zueinander passen. Ist ein Clip zu kurz für die zugehörige Überblendung, soll eine Warnmeldung ausgegeben und der Clip auf die notwendige Länge erweitert werden.

Aufgabe 3: Klasse `AVI_Diashow` zur Erzeugung von AVI-Filmen aus Bildern (10 Punkte)

Vervollständigen Sie die Klasse `AVI_Diashow`, so dass sie die beschriebene Funktionalität erhält und damit ein AVI-Film erzeugt werden kann.

Teilen Sie den Programmcode sinnvoll auf mehrere Programm- und Header-Dateien auf und erstellen Sie ein zugehöriges Makefile.

Zur Vereinfachung können Sie annehmen, dass alle BMP-Bilder die gleiche Größe aufweisen wie der zu erstellende AVI-Film. (Freiwillige Erweiterung: Skalierung der BMP-Bilder auf die Bildgröße des AVI-Films.)

Testen Sie die Klasse mit dem Programm „`Test_AVI_Diashow.cxx`“. Dieses erzeugt mittels Ihrer Klasse aus den 6 Bildern „`Schwarz.bmp`“, „`Bild1.bmp`“ bis „`Bild4.bmp`“ und „`Weiss.bmp`“ den Film „`Test.bmp`“.

Hinweis: Zum Lesen von BMP-Bildern dient die beigestellte Klasse `BmpRead`.

Hinweis: Wenn Sie Objekte, deren Typ eine Unterklasse von `GrafikElement` ist, in einem Container der Standardbibliothek ablegen wollen, sollten Sie die in der Vorlesung vorgestellte Klasse `GrafikHuelle` verwenden.

Hinweis: Manche Windows-XP Rechner werden extrem langsam, wenn große AVI-Filme im aktuellen Verzeichnis liegen. Grund ist, dass der Explorer eine Vorschau des AVI-Films erzeugen will. Damit verbraucht er nahezu die gesamte Rechenzeit. Eine Abhilfe ist auf der Web-Seite <http://www.windows-tweaks.info/html/cpu-auslastung.html> unter Punkt 4 beschrieben.