



Bachelorpraktikum AI3

Aufgabe 2: Einlesen einer Bilddatei in den Arbeitsspeicher

Ziel dieser Aufgabe ist das Einlesen einer Bilddatei in den Arbeitsspeicher.

Rahmenbedingungen

- a) Die Bilddatei muss im TGA-Format gelesen werden können.
- b) Das Resultat des Einlesevorgangs soll eine Instanz einer ebenfalls zu entwickelnden Bild-Klasse sein. In dieser Klasse sollen RGB(A)-Daten mit einem Byte pro Farbkanal und Pixel vorgehalten werden. Der Ladevorgang soll durch statische Methoden innerhalb der Bild-Klasse umgesetzt werden.
- c) Beim Schreiben der Bild-Klasse ist ein C++-übliches Vorgehen zu nutzen, insbesondere das RAII-Pattern (Resource acquisition is initialization) zum Vermeiden ungültiger Klassenzustände. In anderen Worten, eine Konstruktion eines Bildes soll bereits vollständige Parameter erfordern. Fehlerzustände müssen im Konstruktor erkannt und ungültige Klasseninstanzen durch Ausnahmebehandlung vermieden werden. Die Bildklasse soll Speicher intern automatisch verwalten und bei Zerstörung eines Bildobjekts automatisch freigeben.
- d) Korrupte oder nicht unterstützte Bilddaten sind zu erkennen und über geeignete Mechanismen (Exceptions) zu propagieren. Speicherlecks sind zu vermeiden, ebenfalls sind offene Dateien korrekt zu schliessen.
- e) Redundanzen im Code sind durch geschickte Funktionsaufteilung zu vermeiden. Alle Variablen, Methoden, und Klassen sind entsprechend dem Anwendungs-Kontext sinnvoll zu benennen. Dokumentation und Formatierung analog zur ersten Aufgabe.
- f) Wie bei der ersten Aufgabe ist beim Implementieren auf korrektes C++, auf vollständige und sinnvolle Dokumentation, sowie auf saubere Formatierung und Strukturierung zu achten.

Einlesen einer TGA-Datei

Das Einlesen einer TGA-Datei ist von Hand zu implementieren, es darf keine Bibliothek genutzt werden. Falls Tutorials aus dem Netz verwendet werden, ist der Code dennoch selbst neu zu schreiben. Copy und Paste ist nicht zulässig, und fällt spätestens bei Detailfragen im Testat auf.

Eine Beschreibung des TGA-Formates findet sich unter:
http://de.wikipedia.org/wiki/Targa_Image_File



Es müssen nicht alle möglichen TGA-Dateien und TGA-Eigenschaften unterstützt werden, sondern nur die folgenden Wertebelegungen:

Länge Bild-ID = 0
Farbpalettentyp, Palettenbeginn, Palettenlänge = 0
Bildtyp = 2
X, Y-Nullpunkt = 0
Bits pro Bildpunkt = 24 oder 32

Nach dem Einlesen des Bilddaten-Blocks mit Pixelfarben kann der Einlesevorgang abgebrochen werden. Weitere Metainformationen sind nicht zu lesen.

Zum Testen Ihrer Implementierung können Sie die Kreaturenbilder (32 Bits pro Bildpunkt) und Landschaftsbilder (24 Bits) aus dem eLearning-Kurs nutzen.

Google-Stichwörter

ifstream binary, auto_pointer, vector, tga tutorial, RAI