

Fehlersuche (Debugging) in Computergrafik-Anwendungen

Problem: Das Debuggen von interaktiven Anwendungen ist manchmal schwieriger als beispielsweise von Konsolenanwendungen. Hat man etwa in einer Computergrafikanwendung ein Dreiecksnetz mit `100` Vertices, dann muss man in jedem Frame `100*3=300` `float`-Koordinaten betrachten. Es ist alles andere als trivial, darin einen oder gar mehrere Fehler zu erkennen.

Daher bietet es sich an, wo möglich visuell zu debuggen: Man versucht, die Fehler im 3D-Fenster zu identifizieren. Das ist aber nur möglich, wenn man dort auch etwas erkennen kann. Damit das sichergestellt ist, haben sich unter anderem die folgenden Strategien als erfolgversprechend erwiesen:

- **Aktueller Stand immer sichtbar:** Man stellt bei der Entwicklung immer sicher, dass der aktuelle Stand sichtbar ist. Sobald man eine Änderung einbaut, die die Szene *unverständlich* macht (z.B. Objekte sind unsichtbar, Vertex-Positionen explodieren, ...) macht man die Änderung rückgängig. Man konserviert also einen Zustand der Anwendung, der visuell verständlich ist. Fügt man dann in kleinen Schritten weitere Funktionalität ein, dann wird erst dann zum nächsten Feature übergegangen, wenn das vorherige Feature funktioniert und sichtbar ist.
- **Grafisch Debuggen:** Der Vorteil einer grafischen Anwendung ist, dass das Ergebnis permanent sichtbar ist. Man kann also in der Darstellung Debugging-Information kodieren. Beispielsweise kann man einzelne Dreiecke einfärben oder ihre Größe anpassen, sodass in der Darstellung klar ist, um welches Dreieck es geht. Durch zusätzliche grafische Primitive (beispielsweise Linien, Kugeln, ...) kann man einzelne Teile hervorheben oder Verbindungen zwischen Teilen der Szene darstellen.
- **Minimalbeispiele:** Dies ist eigentlich keine Besonderheit beim Debuggen grafischer Anwendungen, aber auch dort von großer Bedeutung. Man sollte zunächst mit Datensätzen arbeiten, die möglichst einfach sind. Kann man beispielsweise ein Feature oder einen Algorithmus mit einem einzigen Dreieck testen, dann sollte man ein Dreiecksnetz erstellen und verwenden, das auch wirklich nur aus einem Dreieck besteht.