Martin Krosche, Oliver Pajonk

Teilprojekt Achterbahneditor

Aufgabenbeschreibung für das Softwareentwicklungspraktikum am Institut für Wissenschaftliches Rechnen

Institut für Wissenschaftliches Rechnen
Carl-Friedrich-Gauss-Fakultät
Technische Universität Braunschweig
Version: 1.0.3 für das Sommersemester 2009
Revision: 2284
Erstellt am 30. März 2009

Braunschweig, 2009

Inhaltsverzeichnis

Inhaltsverzeichnis i

1 Einleitung 1

2 Über das Dokument 1

3 Aufgabenstellung 1

4 Beschreibung des Editors 1
4.1 Übersicht 3

5 Technische Produktumgebung 3

6 Arbeitsumgebung 4

7 Einsatz 4

8 Qualitätsanforderungen 4

9 Dokumentation 4

10 Ergänzungen 5

Literatur 5

1 Einleitung

Das Institut für Wissenschaftliches Rechnen stellt zwei gekoppelte Projektthemen für das Softwareentwicklungspraktikum (SEP) 2009. Zum einen soll ein CAE-Simulator[1] entwickelt werden, der eine Achterbahn vereinfacht, aber mathematisch korrekt simuliert und dreidimensional visualisiert (Teilprojekt „Achterbahnsimulator“). Zum anderen soll ein CAE-Editor zur komfortablen, ingenieursmäßigen Konstruktion von Achterbahnkurven implementiert werden (Teilprojekt „Achterbahneditor“).

Dieses Dokument enthält die Aufgabenbeschreibung und Anforderungsspezifikation für den *Achterbahneditor*. Die Grundlagen der mathematischen Modellierung finden sich in [1]. Zudem ist es unerlässlich, dass sich beide Gruppen zusätzlich mit der jeweils anderen Aufgabenbeschreibung vertraut machen, um das Gesamtprojekt zu verstehen und auf ein gemeinsames Ziel hinarbeiten zu können.

Zusätzliche Informationen zur Organisation finden sich auf `http://www.wire.tu-bs.de` und der entsprechenden Informationsseite des Insituts für Software Systems Engineering (SSE), `http://www.sse-tubs.de/teaching/ss09/sep`.

2 Über das Dokument

Dieses Dokument enthält viele, im Normalfall blau gekennzeichnete, *Hyperlinks*. Benutzen Sie zuerst diese, um bei Fragen zusätzliche Hilfen oder einfach weiterführende und vertiefende Informationen zu erhalten.

3 Aufgabenstellung

Die Aufgabe dieses Teilprojektes ist der softwareingenieursmäßige Entwurf und die Implementierung eines visuellen, interaktiven CAE-Konstruktionswerkzeugs für Achterbahnen.

Das Konstruktionswerkzeug soll über 2D-Ansichten und -Editoren verfügen (siehe Abbildung 1) und eine Möglichkeit zum vereinfachten Aufruf des Visualisierungs- und Simulationsprogramms des anderen Teilprojekts verfügen.

4 Beschreibung des Editors

Eine 2D Ansicht soll die Grundfläche der zu konstruierenden Bahn aus der Vogelperspektive zeigen. Die Ausdehnung der rechteckigen Grundfläche (Länge und Breite) soll vom Benutzer eingestellt werden können. Ein Raster sowie ein Koordinatenanzeiger für den Mauszeiger geben dem Benutzer Hilfestellung bei der Konstruktion. Einzelne Bahnstützstellen sollen interaktiv in diesem Fenster an die gewünschte Position gesetzt werden können. Da allerdings durch das interaktive Setzen der Stützstellen nur zwei der typischerweise sechs Parameter gegeben werden, müssen die fehlenden Parameter durch den Benutzer *in geeigneter, einfacher Art und Weise* eingegeben werden können. Zu diesen Parametern zählen z.B. die Höhe einer Stützstelle und der Vektor für die Gierachse (siehe [2]). Die Parameter sollen mit geeigneten Standardwerten versehen werden, welche vom Benutzer bei Bedarf modifiziert werden können.

Zudem soll es Bahnbausteine geben, die vom Nutzer angewählt und im Editorfenster interaktiv in ähnlicher Weise wie einzelne Stützstellen gesetzt werden können. Ein Baustein besteht aus einer Reihe von Stützstellen, die das jeweilige Bahnelement vollständig beschreiben. Solche Bausteine sind z.B. 90 Grad Anstiege oder auch Inversionen wie Loopings oder Korkenzieher. Möchte der Benutzer seinen gesetzten Baustein in den Parametern verändern, soll er auch diese Möglichkeit bekommen. Die Auswahl der Bahnkurvenelemente sowie Parameteränderungen erfolgen über geeignete grafische Elemente wie Menüs, Symbolleisten oder Schaltflächen.

[1] CAE = *Computer Aided Engineering*

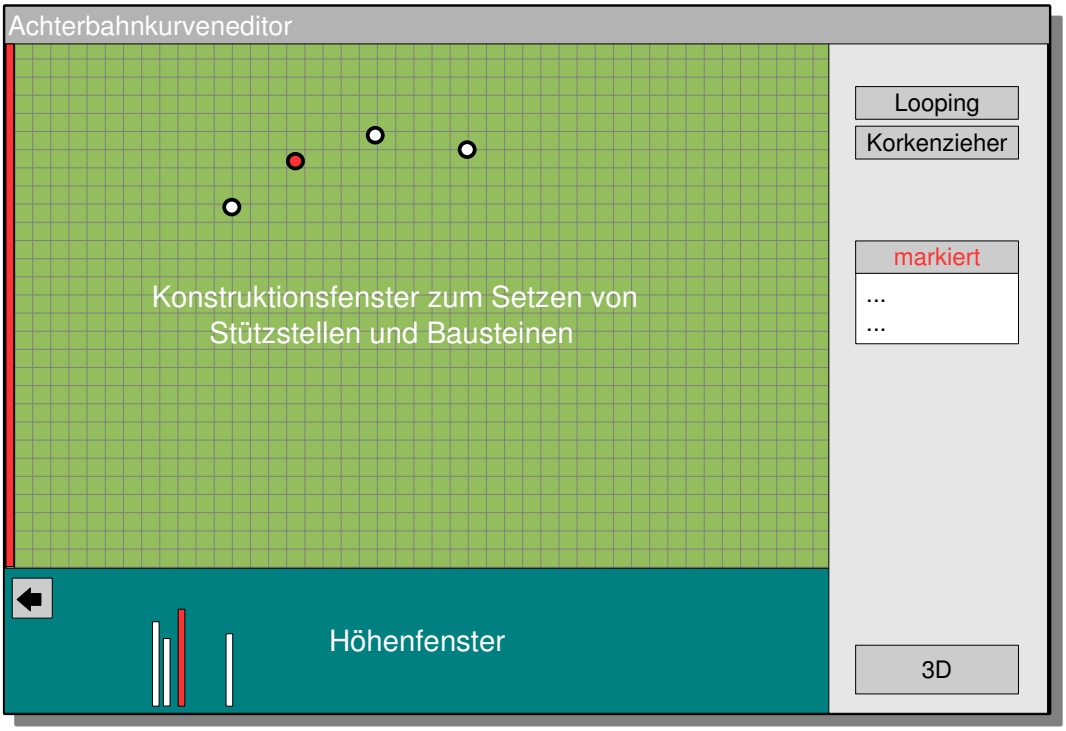

Abbildung 1: *Unvollständige Skizze für einen Achterbahneditor*

In einer weiteren 2D Ansicht sollen seitliche Ansichten (Profile) der aktuell konstruierten Bahn angezeigt werden. Dadurch erhält der Benutzer Einsicht in die Höhenverhältnisse der gesetzten Stützstellen. Eine im Konstruktionsfenster markierte Stützstelle soll auch in diesem Fenster hervorgehoben werden. Zudem soll der Benutzer alle vier Seiten der Grundfläche nacheinander betrachten können. Welche Seite derzeit angezeigt wird soll auch im Konstruktionsfenster durch eine Markierung klar ersichtlich sein.

Über einen Schalter kann der Benutzer sich nach getaner Arbeit seine Konstruktion dreidimensional anschauen. Dies Erfolgt durch Starten des Visualisieres vom Teilprojekt Achterbahnsimulator. In diesem soll dann die Möglichkeit gegeben sein über die Maus zu rotieren und zu translieren, um sich den Bahnverlauf genau anschauen zu können und eventuelle Konstruktionsprobleme zu erkennen. Hierfür ist keine dynamische Simulation der Bahn erforderlich.

Die Konstruktionsdaten einer Bahn sollen vollständig in eine XML-Datei übertragen werden (näheres hierzu in [1]). Desweiteren soll eine solche Datei auch wieder eingelesen werden können, um eine schon bestehende Konstruktion anpassen zu können.

Mindestens zwei real existierende Achterbahnen sollen mit dem Editor möglichst naturgetreu nachgebaut werden. Zudem soll noch mindestens eine ausgedachte Bahn konstruiert werden.

Es ergeben sich Schnittmengen mit dem Teilprojekt Achterbahnsimulator hinsichtlich...

- ...der XML-Behandlung: das Dateiformat muss abgestimmt werden. Hier bietet sich eine Spezifikation und automatisierte Validierung mit Hilfe eines *XML Schemas* an.
- ...der Visualisierung: die Koeffizienten für die Splines müssen von beiden Projekten konsistent und korrekt berechnet werden; die Darstellung des Bahnverlaufs muss konsistent erfolgen (es sollen CAE-Werkzeuge sein!).

Die dynamische Simulation und 3D-Visualisierung wird durch das Teilprojekt Achterbahnsimulator bearbeitet, die Konstruktion durch dieses Teilprojekt. Die XML-Behandlung ist die wichtig-

ste Schnittstelle zwischen beiden Projekten und muss daher sorgfältig und von beiden gemeinsam bearbeitet werden.

4.1 Übersicht

In der folgenden Zusammenfassung wird der zu entwickelnde Editor noch einmal übersichtlich dargestellt.

- 2D Konstruktionsfenster
  - Visualisierung der Konstruktion in Vogelperspektive
  - Raster und Koordinatenanzeige
  - interaktives Setzen von Stützstellen
  - interaktives Setzen von Bausteinen:
    - 90 Grad Anstieg
    - Looping
    - Korkenzieher
    - Immelmann
    - Boomerang
    - *Sonstige*
  - Markierung des derzeit angewählten Kurvenelements
- Menü
  - siehe 2D Konstruktionsfenster (Stützstelle, Bausteine)
  - Möglichkeit zum Start der 3D-Visualisierung und der 3D-Simulation
- Höhenfenster
  - Visualisierung der Stützstellenhöhen
  - Wechsel zwischen den einzelnen Seiten mit Markierung im 2D-Konstruktionsfenster
  - Markierung des derzeit angewählten Kurvenelements
- Einlesen von Konstruktionsdaten einer Achterbahnkurve aus einer XML-Datei
- Auslagern von Konstruktionsdaten einer Achterbahnkurve in eine XML-Datei
- Konstruktion von mindestens zwei real existierenden Bahnen und einer ausgedachten Bahn

5 Technische Produktumgebung

Im Folgenden ist die technische Produktumgebung gegeben. Es stehen im wesentlichen zwei grundsätzlich unterschiedliche Varianten zur Auswahl.

Programmiersprache Zur Wahl stehen *Java* oder *ISO/IEC C++*. Aufgrund der Erfahrungsgemäß kürzeren Entwicklungszeit wird Java empfohlen. Die Referenz für Java ist das *Sun JDK 6*, die Referenz für C++ ist die *GNU Compiler Collection 4.3*. Für C++ ist zusätzlich Kompatibilität zum *Intel C++ Compiler 11.0* wünschenswert (sprich: optional).

Betriebssystem Eine aktuelle 32bit Linux Distribution (z.B. *OpenSuSE*, *Debian*, *Ubuntu*,...) mit *Kernel 2.6*. Wünschenswert ist 32- *und* 64-Bit Kompatibiliät.

Hardware Aktueller, *x86* oder *Intel 64* / *AMD64*-basierter Standard-PC (z.B. *Intel Core 2*)

Testwerkzeuge *JUnit* oder *CppUnit*

Tools/Bibliotheken Hier ist eine Analyse notwendig

6 Arbeitsumgebung

Die Entwicklungs-, Dokumentations und Projektmanagementwerkzeuge, welche benutzt werden sollen, sind die folgenden:

Dokumentenerstellung *OASIS Open Document Format*, *Microsoft Word* oder *LaTeX 2ε*, wobei die Empfehlung aufgrund der weitaus besseren Teamarbeitsmöglichkeiten in Verbindung mit Subversion klar bei LaTeX 2ε liegt. Die Abgabe der Dokumente erfolgt im *PDF Format*.

Entwicklungsumgebung *Eclipse* oder *KDevelop* (je nach Geschmack und Programmiersprache)

Versionsverwaltung *Subversion* (ausschließlich, siehe auch Abschnitt 10)

Build-Werkzeug *Apache Ant*/*Apache Maven* oder *GNU Make*

Projektverwaltung *Trac*, eventuell mit zusätzlichen *Plugins*

7 Einsatz

Die zu entwickelnde Software soll in vollem Umfang auf dem Betriebssystem Linux (Details siehe Abschnitt 5) lauffähig sein und mit dem Editorwerkzeug des anderen Projektes problemlos zusammenarbeiten.

8 Qualitätsanforderungen

Eine benutzerfreundliche, klar strukturierte Bedienung des Editors ist Voraussetzung. Insbesondere auf Exaktheit und Effizienz (in der Benutzung) wird hier Wert gelegt, da es sich um ein Designwerkzeug für Ingenieure handelt mit dem sie möglichst produktiv Achterbahnen konstruieren wollen. Eine möglichst einfache Installation mit ausführlicher Beschreibung ist wichtig.

Der Editor soll ausreichend stabil laufen. Datenverlust und somit Verlust von Arbeit ist aber in jedem Fall zu vermeiden. Besondere Optimierungen bezüglich der Laufzeiteffizienz sind nicht vorgesehen. Das Produkt muss allerdings auf „normalen“ Rechnern in vollem Umfang lauffähig und benutzbar sein.

Das Programmdesign soll bedingt für eine spätere Erweiterung ausgelegt sein. Die Programmstruktur soll mit wenig Zeitaufwand für Außenstehende verständlich sein. Die Arbeit im (Projekt-)Team soll durch die Architektur der Software ermöglicht und unterstützt werden. Nötige Kommentare sollen nach Javadoc/Doxygen Vorgaben erfolgen, um eine möglichst automatische und somit konsistente Dokumentation zu erzeugen. Programmcode und Kommentare sollen, wie in Industrie und Forschung üblich, in Englisch verfasst werden.

Zum Testen der Klassen und Module sollen Unit-Tests, unter der Verwendung einer geeigneten Unit-Test-Bibliothek, herangezogen werden. Bei der Wahl der zu verwendenen Tools/Bibliotheken soll darauf geachtet werden, dass diese mit hoher Wahrscheinlichkeit auch in Zukunft von den Herstellern gepflegt werden.

9 Dokumentation

Zugehörige schriftliche Ausarbeitungen können in deutsch oder englisch verfasst werden. Hierzu wird ausdrücklich LaTeX2e empfohlen, es sind aber auch andere Formate möglich (siehe Abschnitt 6). Vom SSE werden Vorlagen für alle Dokumente bereitgestellt, welche auch zu benutzen sind.

10 Ergänzungen

- Der Einsatz von Subversion als Versionsverwaltungswerkzeug ist *bindend* und muss in der folgenden Weise erfolgen: alle primären (d.h. nicht automatisiert erstellbaren) Artefakte sind in dem bereitgestellten Repository zu verwalten. Dies schließt die folgenden Dinge ein (ist aber nicht beschränkt auf diese): Programmcode; Dokumente bzw. Quelltexte für Dokumente; Grafiken; Prototypen; Diskussionspaper; Präsentationen. Automatisiert erstellte Artefakte dürfen nur aus besonderem Grund ins Repository eingespielt werden. Dies schließt die folgenden Dinge ein (ist aber nicht beschränkt auf diese): Ausführbare Programme und Bibliotheken, welche aus dem Quelltext des Projekts erstellt werden; Temporäre Dateien und Zwischendateien; generierte PDFs. Ein besonderer Grund *für* das Einspielen von sekundären Artefakten ist z.B. das Ablegen des PDFs einer Abgabeversion eines Dokuments.
- Der Inhalt und die zeitliche Entwicklung des Subversion Repositories werden mit zur Bewertung des SEPs herangezogen.
- Die in diesem SEP entstehende Software und Dokumentation soll unter GPL oder LGPL Lizenz (je nach Programmierumgebung) gestellt werden um für spätere Präsentationen zur Verfügung zu stehen.

Schließlich möchten wir anregen, dass Sie, falls Sie Spaß daran haben und die Zeit reicht, selbstständig die Aufgabenstellung erweitern – hierzu ein paar Vorschläge:

- Berechnung der Streckenlänge
- Abschätzen der Fahrzeit
- Hinzufügen von weiteren typischen Features eines CAD-Werkzeugs – dafür ist die Evaluation eines solchen nötig

Literatur

[1] Kayser-Herold, Oliver, Andreas Keese, Markus Krosche, Martin Krosche und Oliver Pajonk: *Achterbahn-Editor/-Simulator - Dokumentation für das Softwareentwicklungspraktikum am Institut für Wissenschaftliches Rechnen*, März 2009.

[2] Kayser-Herold, Oliver, Andreas Keese, Markus Krosche, Martin Krosche und Oliver Pajonk: *Teilprojekt Achterbahnsimulator -Aufgabenbeschreibung für das Softwareentwicklungspraktikum am Institut für Wissenschaftliches Rechnen*, März 2009.