Planetarium

Simulation SE Projekt

Inhalt

[1. Einleitung 3](#_Toc456929)

[2. Projektmanagement 3](#_Toc456930)

[2.1. Besprechungen 3](#_Toc456931)

[2.2. Merkmale Klassisch & Agil 3](#_Toc456932)

[2.3. Aufgabenverteilung 4](#_Toc456933)

[2.4. Probleme 4](#_Toc456934)

[3. Projektplanung 4](#_Toc456935)

[3.1. Themenabsprache 4](#_Toc456936)

[3.2. Projektziel 4](#_Toc456937)

[3.3. Projektabgrenzung 4](#_Toc456938)

[4. Projektvorbereitung 5](#_Toc456939)

[4.1. Projektumgebung 5](#_Toc456940)

[4.2. Datengrundlage 5](#_Toc456941)

[5. Projektumsetzung 6](#_Toc456942)

[5.1. Oberfläche 6](#_Toc456943)

[5.2. Grunddaten 6](#_Toc456944)

[5.3. Umlaufbahnen 6](#_Toc456945)

[5.4. N-Körper Problem 6](#_Toc456946)

[6. Fazit 7](#_Toc456947)

[7. Quellen 8](#_Toc456948)

# Einleitung

Die nachfolgende Projektdokumentation wurde angefertigt, um das Projekt im Rahmen der Lehrveranstaltung „Software Engineering“ Projekt an der Ostfalia Hochschule für angewandte Wissenschaften im Studiengang Informatik zu veranschaulichen.

Die Hochschule besteht aus vier Standorten (Wolfenbüttel, Wolfsburg, Suderburg und Salzgitter). Gegründet wurde sie im Jahr 1971 und zählt heutzutage ca. 13000 Studenten.

# Projektmanagement

Die Projektherangehensweise wurde zum Anfang des Projektes still festgelegt. Die Gruppe hatte schon mehrere Projekte zusammen bearbeitet und aus diesem Grund wusste jedes Teammitglied, wie der Projektzeitraum ablaufen wird.

Unsere Methode bestand aus einer Vermischung von einem agilen und klassischen Ansatz.

## Besprechungen

Die regelmäßigen Treffen fanden circa alle 1-2 Wochen statt. In diesen Treffen wurden jeweils der aktuelle Stand des Programms, die zukünftigen Aufgaben und Probleme besprochen. Weiterhin wurden Ideen besprochen.

Die meisten Treffen fanden nach dem folgenden Schema statt:

1. Aktuelle Lage
2. Probleme mit den vorherigen Aufgaben
3. Neue Ideen
4. Probleme Allgemein
5. Aufgabenverteilung

Während dieser Besprechungen konnte jedes Mitglied seine Ideen und Vorschläge mit einbringen. Es gab keinen direkten Produkt Manager, sondern nur einen Leiter der versucht hat am Ende alle Meinungen und Ideen zusammenzubringen.

## Merkmale Klassisch & Agil

Wie eingangs erwähnt bestand unsere komplette Projektphase aus einem Mix aus klassischen und agilen Methoden. Die lag daran, dass wir uns für kein direktes Modell entschieden haben, sondern versucht haben für unsere Gruppengröße von fünf Mitgliedern ein geeignetes Modell zu finden.

Unsere agile Arbeitsweise bezieht sich vor allem auf die flexiblen Anforderungen. Unsere Ideen um das Projekt haben sich konstant verändert. Vor allem die genaue Richtung zum Ende hin war am Anfang unklar. Dies konnte sich erst während der Projektphase herausstellen.

Gleichermaßen konnte man den Abstand zwischen den Treffen als einen Sprint sehen. Zu jedem neuen Sprint gab es eine Besprechung und zu jedem abgelaufenen eine Nachbearbeitung.

Die klassische Arbeitsweise bezieht sich vor allem auf das Endprodukt, da der Kunde (Professor Pekrun) nur das Endprodukt erhält. Außerdem wusste jeder ungefähr welche Aufgaben er übernehmen wird während der Entwicklung.

## Aufgabenverteilung

Die Aufgaben wurden vor allem nach Interessen und Wissensstand verteilt. Da in unserem Projekt vor allem die Mathematik und Physik eine große Rolle spielt wurden die jeweiligen Teile auf die entsprechenden Personen aufgeteilt. Weiterhin gab es einen Programmierteil, der für alle fünf relevant war.

Aaron = Physik, Mathematik, Unity Programmierung

Frederik = Physik, Mathematik, Unity Programmierung

Michael = Unity Programmierung, Oberflächenprogrammierung

Jasha = Fehlerabschätzung, Unity Programmierung

Felix = Projektmanagement, Unity Programmierung, Oberflächenprogrammierung

## Probleme

Die Probleme während der Entwicklungsphase wurden untereinander abgesprochen und geklärt. Sollte ein Gruppenmitglied nicht weiterkommen, wurde die Aufgabe in der gesamten Gruppe noch einmal besprochen und versucht zu lösen. Sollte diese Methode nicht funktioniert haben, wurde die Aufgabe weiterverteilt, um einen Ansatz zu versuchen.

# Projektplanung

Da das Projekt im Rahmen einer Lehrveranstaltung erstellt wird, gab es vor ab eine Besprechung, welches Thema ausgewählt wird und auf welcher Basis das Projekt erstellt wird.

Professor Pekrun hatte in der ersten Vorlesung erwähnt, dass das Projekt eine Simulation über ein bestimmtes Themenfeld sein soll. Dies war die einzige Voraussetzung.

## Themenabsprache

Im ersten Treffen zwischen der Gruppe wurden diverse Themenvorschläge ausgetauscht. Am Ende des Meetings gab es die Wahl zwischen einer Ameisenkolonie- und einer Planentenumlaufbahnsimulation. In Rücksprache mit Herr Pekrun hatten wir uns für den zweiten Vorschlag entschieden. Daraus entwickelte sich folgendes Anforderungsprofil für unser Projekt.

## Projektziel

Das Projekt soll dazu dienen die Umlaufbahnen vom Sonnensystem zu simulieren. Um die korrekte Funktionsweise zu überprüfen, sollen die Daten am Ende mit offiziellen Daten verglichen werden. Dieser Vergleich wird in mehreren Testdurchläufen durchgeführt (1 Jahr, 10 Jahre,100 Jahre).

Simuliert werden alle Planeten im Sonnensystem. Gleichzeitig soll es die Möglichkeit geben, verschiedene Objekte wie z.B. ein Asteroid durch die Umlaufbahn zu schicken, um den Einfluss von Fremdkörpern zu simulieren.

Die Daten, die aus der Simulation gewonnen werden, sollen ausgegeben und mit offiziellen Daten verglichen werden.

## Projektabgrenzung

Anfangs war das Projekt dazu gedacht eine Aussage über die Umlaufbahnentwicklung der nächsten 100.000 Jahre zu treffen. Da dies den Projektrahmen aufgrund von Genauigkeit übertreffen würde, entwickeln wir in diesem Schritt ein Prototyp mit einem möglichst kleinen Berechnungsfehler.

# Projektvorbereitung

Wie in der Einleitung erklärt, wurde das Programm für die Veranstaltung „Software Engineering“ entwickelt. Da dies in Gruppenarbeit geschehen ist, wurde ein eigenes Github Repository angelegt, um einen Austausch zu ermöglichen https://github.com/Voxen4/Planetarium.

## Projektumgebung

Das Projekt wurde mit der Game-Engine Unity 3D erstellt. Unity wurde 2005 von Unity Technologies in Amerika entwickelt. Es unterstützt 2D bzw. 3D Anwendungen, sowie Simulationen und bringt diverse Tools zur Berechnung und Darstellung von Objekten mit. Gleichzeitig läuft Unity auf verschiedensten Plattformen.

Für unsere Simulation haben wir die Programmiersprache C# benutzt.

## Datengrundlage

Für unser System ist es essential wichtig Daten zum vergleichen zu haben. Diese Daten bilden die Grundlage unserer Startpositionen bzw. dienen zum Vergleich der Umlaufbahnen.

Um an diese Daten zu gelangen, haben wir im Zuge des Projekts einige Planetarien (Planetarium Berlin), Universitäten (TU BS) und Sternenwarten (Thüringer Landessternwarte) angeschrieben. Die meisten dieser Ansprechpartner haben uns direkt auf die Seite der NASA verwiesen. Das „Horizon Web-Interface“ bietet für verschiedenste Objekte im All Daten an. Diese Daten reichen von 1600-2500. Für unsere Daten haben wir meist folgende Einstellungen gewählt:

|  |  |
| --- | --- |
| **Ephemeris Type** | VECTORS |
| **Target Body** | Jeweiliger Planet |
| **Coordinate Origin** | Sun (body center) |
| **Time Span** | Variabel (1 day) |

Ein Beispieldatensatz hat folgendes Format:



VX X-component of velocity vector (au/day)

VY Y-component of velocity vector (au/day)

VZ Z-component of velocity vector (au/day)

LT One-way down-leg Newtonian light-time (day)

RG Range; distance from coordinate center (au)

RR Range-rate; radial velocity wrt coord. center (au/day)

Zu der Schnittstelle liegt eine Dokumentation vor, welche die jeweiligen Werte beschreibt. Für unser Projekt waren vor allem die X, Y und Z Koordinaten von Relevanz.

# Projektumsetzung

Die Simulation wurde abwechselnd in zwei Schritten umgesetzt. Auf der einen Seite wurde jeweils die Mathematik/Physik hergeleitet. Die andere Seite beinhaltete die Implementierung der Formeln in Unity und damit Berechnung der jeweiligen Daten zu den Planeten.

Das Ziel war es wie Anfangs erwähnt eine Simulation der Umläufe der Planeten zu erschaffen und anschließend eine Fehlerabschätzung durchzuführen. Dazu sollen Objekte durch den Orbit geschossen werden, um den Einfluss dieser Objekte zu simulieren.

## Oberfläche

## Grunddaten

## Umlaufbahnen

## N-Körper Problem

# Fazit

Abschließend bleibt über das Projekt zusagen, dass wir als Gruppe einiges über das Thema Physik im Zusammenhang mit dem Sonnensystem gelernt haben. Weiterhin konnten wir uns alle in das Thema Game Engine in Bezug von Unity einarbeiten. Dies hat den Vorteil, da ein paar Mitglieder der Gruppe den Weg einschlagen möchten, mit solcher Software zu arbeiten.

Auch zu erwähnen ist, dass der Aufwand für einen funktionsfähigen Prototypen sehr hoch angesetzt war, dies aber erst im Nachhinein deutlich wurde. Dies lag vor allem an den vielen verschiedenen Formeln und Fehlern, die sich in den Code dadurch eingeschleust haben.

Wenn wir als Gruppe dieses Thema noch einmal angehen würden, ist uns diesmal bewusst, dass wir andere Software benutzen würden. Die Unity Engine erfüllt ihren Job in angemessenen Rahmen, ist aber nicht auf eine solche Simulation ausgelegt. Alle Daten mussten skaliert und umgerechnet werden. Weiterhin haben wir eine eigene Physik Engine (Runge Kutta Verfahren) geschrieben, da die von Unity genutzte (Euler Verfahren) unseren Fehler deutlich vergrößern würde.

Frei nach dem Prinzip hinterher ist man schlauer, würden wir das Thema Simulation anders angehen.

# Quellen

<https://unity3d.com/de>

https://ssd.jpl.nasa.gov/horizons.cgi