

# Projekt „Gottorfium“

## Simulation und Visualisierung eines Planetensystems und des Sternenhimmels

Sebastian Bechtold, Nils Weiher

Universität Heidelberg, Interdisziplinäres Zentrum für  
wissenschaftliches Rechnen (IWR)

Ein Softwarepraktikum für Fortgeschrittene im Wintersemester 2007/2008

# Gliederung

1. Ein Überblick: Was macht unser Programm?
2. Ephemeridenrechnung: Positionsbestimmung von Objekten im Planetensystem
3. Räumliche Orientierung von Objekten im Planetensystem
4. Objekte ausserhalb des Planetensystems: Sterne und Sternbilder
5. Implementierung
6. Abschluss

Ein Überblick: Was macht unser Programm?

# Ein Überblick: Was macht unser Programm?

- **Visuelle Simulation eines Planetensystems und des Sternenhimmels**
- Ziel: Realitätsnahe Darstellung des Himmels abhängig von **Standort** und **Zeitpunkt**
- Grundsätzlich: **Zwei völlig unterschiedliche Klassen** von darzustellenden Objekten: Objekte **innerhalb** des Planetensystems (Planeten, Monde, Asteroiden etc.) und **ausserhalb** (Sterne, Nebel, Galaxien etc.)
- Konsequenz für die Programmierung: Völlig getrennte, unterschiedliche Darstellungsmethoden für beide Objektklassen (dazu später mehr)

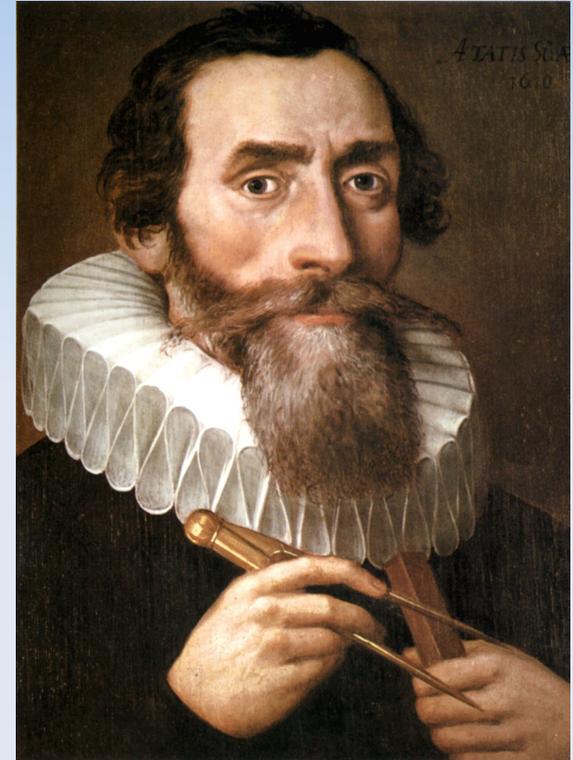
# Zwei grundverschiedene Klassen darzustellender Objekte

- Objekte im Planetensystem:
  - Relativ geringe Entfernungen: Wenige Lichtminuten – max. einige Lichtstunden
  - Deutlich wahrnehmbare Bewegungen: I.d.R. Elliptische, periodische Umlaufbahnen
  - Nicht selbstleuchtend, werden vom Zentralstern beleuchtet
- Objekte ausserhalb des Planetensystems:
  - Gigantische Entfernungen: Min. 5 Lichtjahre (Alpha Centauri) – max. viele Millionen Lichtjahre (andere Galaxien)
  - Im Verhältnis zur Entfernung nahezu keine Bewegung
  - Selbstleuchtend

# Ephemeridenrechnung: Positionsbestimmung von Objekten im Planetensystem

# Die Keplerschen Gesetze (I)

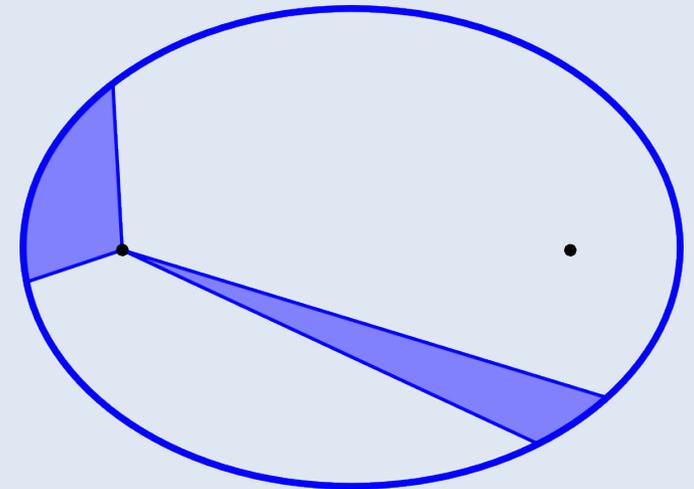
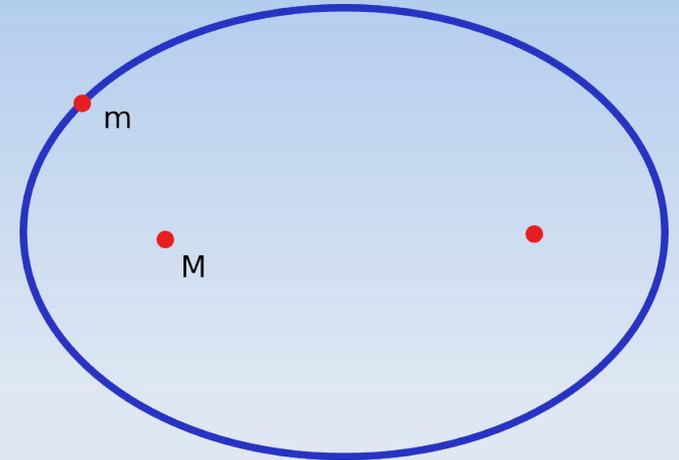
- Die drei Keplerschen Gesetze beschreiben **grundlegende Gesetzmäßigkeiten von Planetenbahnen**
- Entdeckt von **Johannes Kepler** 1609 (erstes und zweites) bzw. 1618 (drittes)



Johannes Kepler  
(1571-1630),  
Bildquelle: Wikipedia

# Die Keplerschen Gesetze (II)

1. Die Umlaufbahn eines Objekts ist eine Ellipse und das Schwerezentrum des Systems liegt in einem ihrer Brennpunkte
2. In gleichen Zeiten überstreicht der Fahrstrahl (Objekt – Schwerezentrum) gleiche Flächen.
3. Die Quadrate der Umlaufzeiten zweier Objekte entsprechen den Kuben ihrer großen Halbachsen:



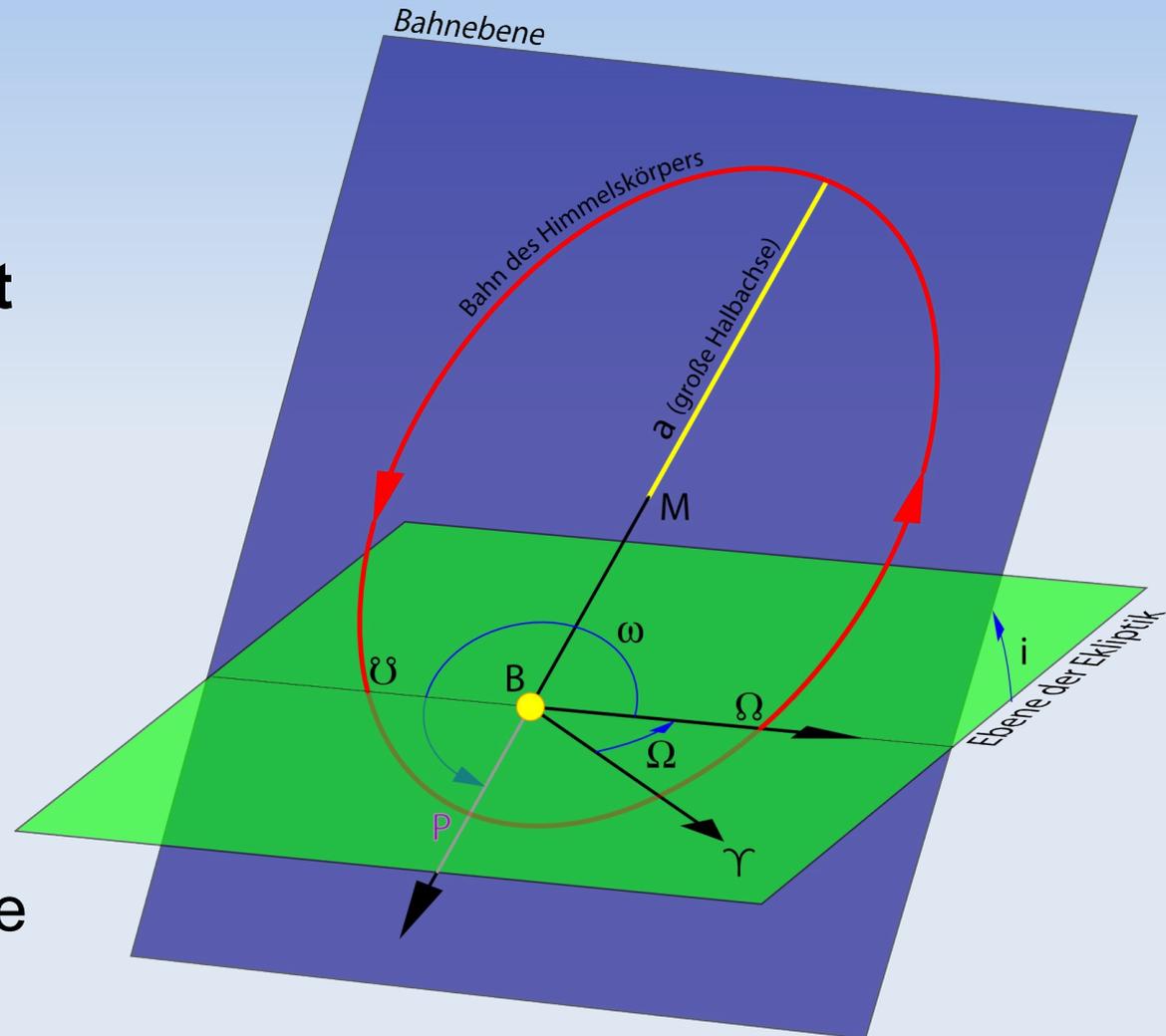
$$(T_1 / T_2)^2 = (a_1 / a_2)^3$$

# Bahnelemente

- Variablen zur Beschreibung einer konkreten Umlaufbahn mit Hilfe der Keplerschen Gesetze
- Zahlreiche Bahnelemente definiert, allerdings viele redundant / voneinander abhängig
- 6 unabhängige Bahnelemente erforderlich für vollständige, eindeutige Bahnbeschreibung
- Positionsbestimmung für Objekt erfolgt durch Eingabe der Bahnelemente in Formeln, die aus den Keplerschen Gesetzen abgeleitet sind (hier nicht behandelt)

# 6 typische Bahnelemente zur Positionsberechnung

- **Große Halbachse** der Bahnellipse
- **Numerische Exzentrizität** der Bahnellipse
- **Mittlere ekliptikale Longitude** des Objekts
- **Longitude des aufsteigenden Knotens**
- **Longitude der Periapsis**
- **Inklination** der Bahnebene gegen die Ekliptikebene



Quelle der Zeichnung: Wikipedia, modifiziert

# Bahnberechnung: Von der Theorie zur Praxis

**Wichtig:** Alle Werte **ändern sich mit der Zeit** und beziehen sich deshalb immer auf einen gegebenen **festen Zeitpunkt**.

**Im folgenden Abschnitt:**

- Wie erhält man die Bahnelemente für einen gewünschten Zeitpunkt?
- Welche anderen Methoden zur Positionsberechnung gibt es?
- Welche Methoden setzen wir in der Simulation für welche Objekte ein?

# Positionsberechnung für die Planeten

- Wir verwenden: *Variations séculaires des orbites planétaires 87 (VSOP87)*
- Von P. Bretagnon und G. Francou, Paris 1987
- Eine der genauesten Methoden zur Positionsberechnung für die acht großen Planeten
- Verfügbar in verschiedenen Versionen, welche Positionsangaben in unterschiedlichen Formaten liefern
- Wir verwenden Version, welche die Bahnelemente für gewünschten Zeitpunkt zurückliefert und aktualisieren so die Bahnelemente eines Objekts 1x pro Umlauf. Darauf basierend Positionsberechnung nach Kepler für jeden einzelnen Frame.

# Positionsberechnung für den Erdmond

- Wir haben für den Mond leider kein Verfahren zur Berechnung der Bahnelemente für gegebene Zeitpunkte (d.h. Ähnlich VSOP87) gefunden
- Häufig genutztes Verfahren *ELP2000* liefert lediglich direkt die Position des Mondes zurück
- Gezwungenermaßen verwenden wir auch *ELP2000*. Wir hätten aber lieber ein Verfahren, das Bahnelemente liefert, da diese für verschiedene Visualisierungen und Berechnungen sehr nützlich sind.

# Positionsberechnung für Monde anderer Planeten

- Prinzipiell: Sehr komplex wegen gravitativer Wechselwirkungen mit dem umkreisten Planeten (und anderen). Für die Monde jedes Planeten sind andere, spezielle Algorithmen erforderlich.
- Bisher haben wir keinen Mond ausser dem der Erde implementiert. Ephemeridenrechnungsverfahren für die Monde aller großen Planeten sind aber in freier Software wie *Stellarium* oder *Celestia* enthalten und könnten übernommen werden.

# Positionsberechnung für Asteroiden, Kometen und Raumfahrzeuge

- Oft starke und häufige Änderungen der Bahnelemente wegen geringer Masse (leichte gravitative Beeinflussbarkeit), z.B. durch
  - Gravitation von Sonne, Planeten und Monden
  - Sonnenwind
  - Aktive Kursänderungen (bei Raumfahrzeugen)
  - Reibung mit Atmosphäre (bei Satelliten in niedrigen Umlaufbahnen)
- Längerfristige Vorausberechnung von Bahnelementen bzw. Positionen daher praktisch nicht möglich
- Begrenzte Lösung: Online-Update der Bahnelemente von Bezugsquelle im Internet (implementiert z.B. in *KStars*)

# Räumliche Orientierung von Objekten im Planetensystem

# Räumliche Orientierung von Objekten im Planetensystem

- Zur vollständigen Beschreibung der räumlichen Lage eines Objekts gehört neben dem Ort auch die Orientierung im Raum.
- Für von uns modellierte Himmelskörper bedeutet dies:
  - Orientierung aller Achsen zu einem Referenzzeitpunkt
  - Zeitliche Veränderung der Achsenorientierungen

# Räumliche Orientierung konkret

- Richtung der **Rotationsachse** zu Referenzepoche J2000
- Richtungsvektor (Mittelpunkt –  $0^\circ$ Lon. /  $0^\circ$ Lat.) zu Referenzepoche J2000 (**Sternzeit** zu J2000)
- Zeitabhängige Änderung des Richtungsvektors der Rotationsachse (**Präzession**)
- Zeitabhängige Änderung des Richtungsvektores (Mittelpunkt –  $0^\circ$ Lon. /  $0^\circ$ Lat.) (**Rotationsgeschwindigkeit**)
- Quelle: *Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements 2006*

# Objekte ausserhalb des Planetensystems: Sterne und Sternbilder

# Die "Himmelskugel"

- Sterne sind **außerordentlich** weit vom Sonnensystem entfernt, ihre Entfernung wird daher **nicht** simuliert
- Alle Sterne sind auf einer Himmelskugel in gleichem Abstand zur Kamera platziert (Radius im Grunde egal)
- Himmelskugel hat stets die Kamera als Mittelpunkt, um die Illusion von großer (unendlicher) Entfernung zu erzeugen ("Skybox"-Konzept)
- Im OpenGL-Objektraum sind daher die Sterne i.d.R. Näher an der Kamera als die Objekte im Sonnensystem. Korrekte Überdeckung wird durch passende Wahl der Zeichenreihenfolge erreicht.

# Sternkataloge

- Laden von Sternkatalogen
  - modular konzipiert
  - Lade-Klassen für verschiedene Katalog-Formate
  - können gleichzeitig im Programm existieren
- Derzeit nur Klasse für den **Yale Bright Star Catalogue** (YBSC)
  - ~4000 von insgesamt ~6000 mit bloßem Auge sichtbaren Sterne

# Modellierte Sterneigenschaften

- Sternort (Position auf der Himmelkugel)
- Scheinbare Helligkeit (Magnitude) in einigen festen Abtufungen (noch keine "stetige Funktion implementiert)

# Konstellationen und Sternbilder

- Laden von Konstellation, wie bei Sternen, d.h. Unterstützung verschiedener Dateiformate
- momentane Implementierung
  - eigenes XML-Format
  - enthält Linien, sowie Namen der Konstellationen
  - Intern keine datenstrukturelle Verbindung von Sternen und Konstellationen
  - Konstellationslinien können auch ganz ohne geladene Sternenkataloge eingeblendet werden

# Sonstiges zur Implementierung

# Begrenzte Genauigkeit durch nicht berücksichtigte Faktoren...

- **... bezüglich Berechnung der Beobachterposition:**
  - Planet ist keine perfekte Kugel (Abplattung durch Rotationsfliehkraft, Gebirge etc.)
  - Zeitliche Veränderung der Rotationsachsenorientierung nicht zu 100% berücksichtigt (Präzession ja, Nutation nein(?))
  - ... etc.
- **... bezüglich Berechnung der Position von Objekten:**
  - Verzögerungen durch begrenzte Lichtgeschwindigkeit
  - Lichtablenkung durch große Massen (Raumkrümmung)
  - Lichtablenkung/-streuung durch Atmosphäre
  - ... etc.

# Programmarchitektur (I)

- Komplette implementiert in C++, massiv objektorientiert
- Grundsätzlich: Code für physikalische Simulation und Visualisierung getrennt (anders als z.B. bei Stellarium)
  - Erlaubt Implementierung unterschiedlicher Visualisierungsmethoden (z.B. Himmelskartenerzeugung als PNG, o.ä.)
  - Bisher nur interaktive OpenGL-Darstellung realisiert
- Modulare Architektur ermöglicht einfache Erweiterung und Austausch von Komponenten

# Programmarchitektur (II)

- Integration der Gottorf-Globus-Texturen "minimalinvasiv"
- Keine Änderungen an Basiskomponenten, nur Ableitungen ohnehin vorhandener Klassen
- Code aus dem alten Gottorf-Globus-Programm im Wesentlichen unverändert übernommen

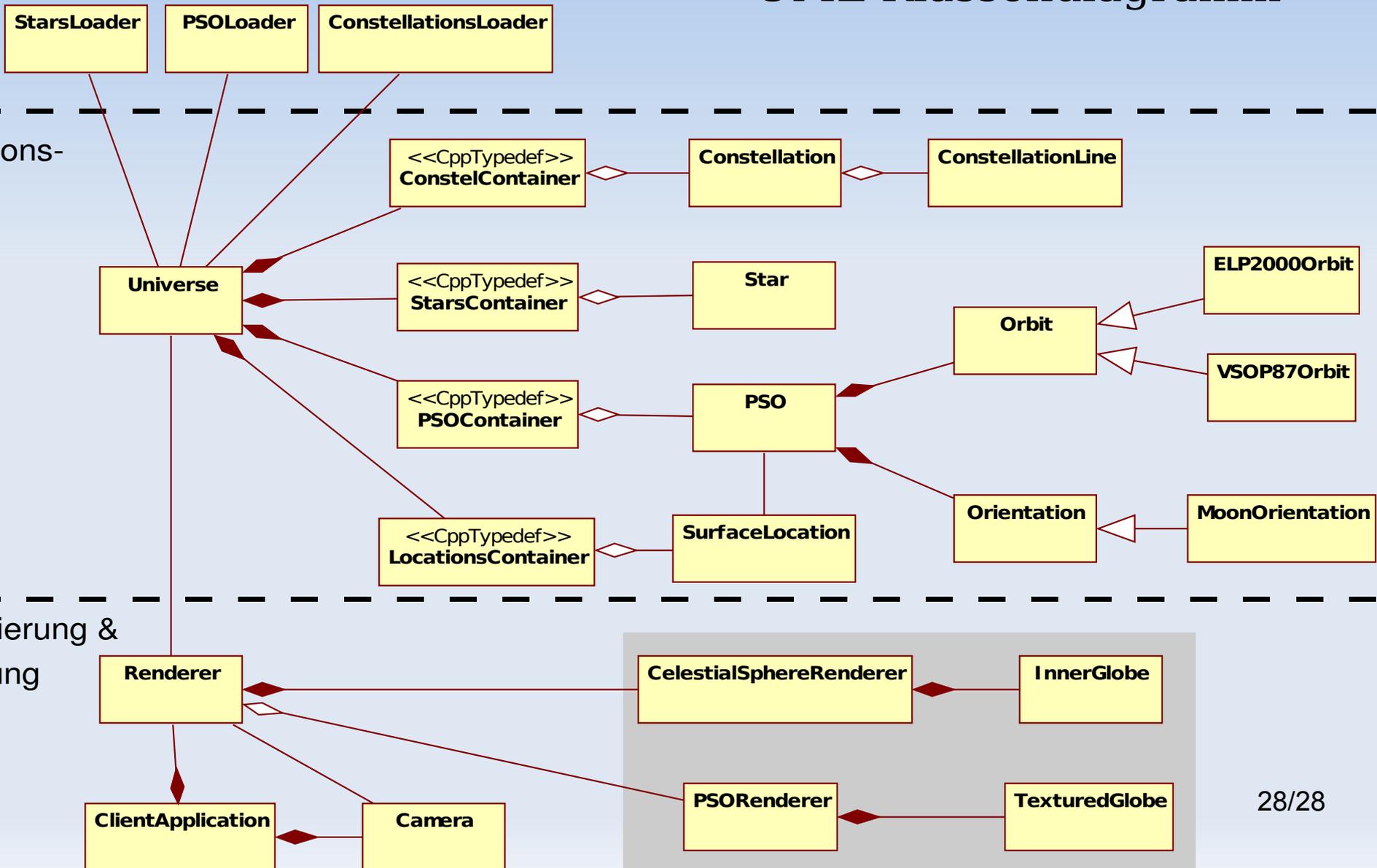
# UML-Klassendiagramm

Simulationsdatenquellen

Gottorfium: Vereinfachtes UML-Klassendiagramm

Simulationsmodell

Visualisierung & Bedienung



# Verbesserungsmöglichkeiten (I)

- **Genauigkeit des Himmelsmodells**
  - Eigenbewegung von Sternen
  - Lichtgeschwindigkeit
  - Verschiedene Spektralklassen / Sternfarben
- **Inhalte und Bedienung**
  - Nebel und Deep-Sky-Objekte (Galaxien)
  - Einblendung von Koordinatengittern
  - Möglichkeit zum Selektieren von Objekten und Einblenden von Informationen darüber

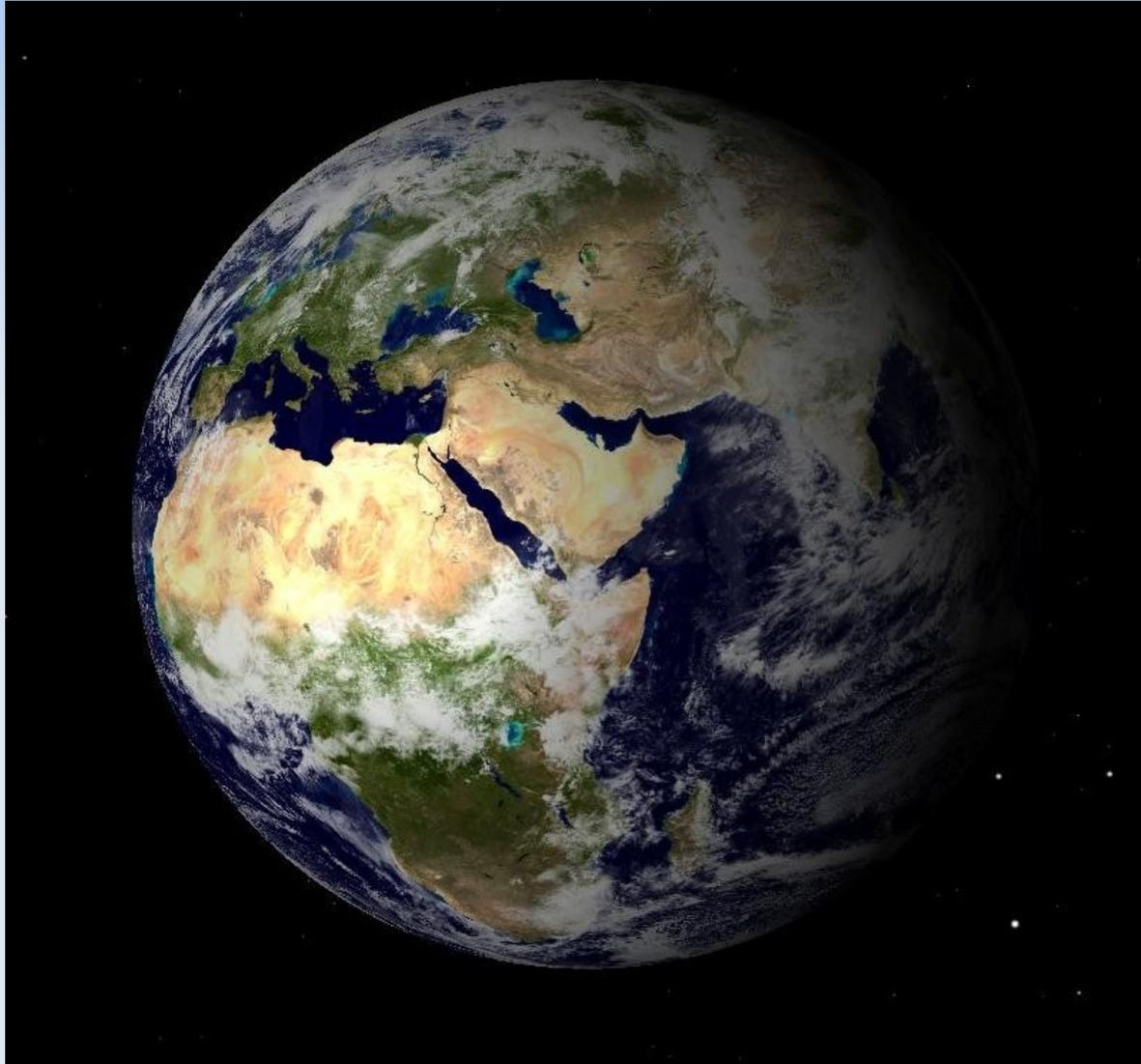
# Verbesserungsmöglichkeiten (II)

- **"Eye Candy"**
  - Atmosphäreneffekte
  - "Virtuelle Texturen" / Texture-Tiling
  - Bump Mapping für Oberflächen
  - Sonnenblendeffekt / Lens-Flares

# Vergleichbare, einflussgebende Software

- *Stellarium*
  - <http://www.stellarium.org/>
- *Cartes du Ciel*
  - <http://www.stargazing.net/astropc/>
- *Celestia*
  - <http://www.shatters.net/celestia/>
- *KStars*
  - <http://edu.kde.org/kstars/>

# Ende



Vielen Dank für Ihre Aufmerksamkeit!

# Projekt „Gottorfium“

## Simulation und Visualisierung eines Planetensystems und des Sternenhimmels

Sebastian Bechtold, Nils Weiher

Universität Heidelberg, Interdisziplinäres Zentrum für  
wissenschaftliches Rechnen (IWR)

Ein Softwarepraktikum für Fortgeschrittene im Wintersemester 2007/2008

# Gliederung

1. Ein Überblick: Was macht unser Programm?
2. Ephemeridenrechnung: Positionsbestimmung von Objekten im Planetensystem
3. Räumliche Orientierung von Objekten im Planetensystem
4. Objekte ausserhalb des Planetensystems: Sterne und Sternbilder
5. Implementierung
6. Abschluss

## Ein Überblick: Was macht unser Programm?

# Ein Überblick: Was macht unser Programm?

- **Visuelle Simulation eines Planetensystems und des Sternenhimmels**
- Ziel: Realitätsnahe Darstellung des Himmels abhängig von **Standort** und **Zeitpunkt**
- Grundsätzlich: **Zwei völlig unterschiedliche Klassen** von darzustellenden Objekten: Objekte **innerhalb** des Planetensystems (Planeten, Monde, Asteroiden etc.) und **ausserhalb** (Sterne, Nebel, Galaxien etc.)
- Konsequenz für die Programmierung: Völlig getrennte, unterschiedliche Darstellungsmethoden für beide Objektklassen (dazu später mehr)

## Zwei grundverschiedene Klassen darzustellender Objekte

- Objekte im Planetensystem:
  - Relativ geringe Entfernungen: Wenige Lichtminuten – max. einige Lichtstunden
  - Deutlich wahrnehmbare Bewegungen: I.d.R. Elliptische, periodische Umlaufbahnen
  - Nicht selbstleuchtend, werden vom Zentralstern beleuchtet
- Objekte ausserhalb des Planetensystems:
  - Gigantische Entfernungen: Min. 5 Lichtjahre (Alpha Centauri) – max. viele Millionen Lichtjahre (andere Galaxien)
  - Im Verhältnis zur Entfernung nahezu keine Bewegung
  - Selbstleuchtend

# Ephemeridenrechnung: Positionsbestimmung von Objekten im Planetensystem

# Die Keplerschen Gesetze (I)

- Die drei Keplerschen Gesetze beschreiben **grundlegende Gesetzmäßigkeiten von Planetenbahnen**
- Entdeckt von **Johannes Kepler** 1609 (erstes und zweites) bzw. 1618 (drittes)

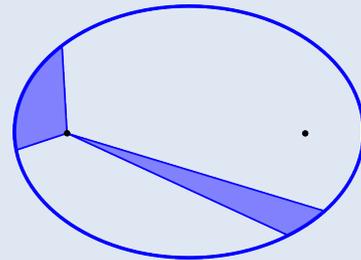
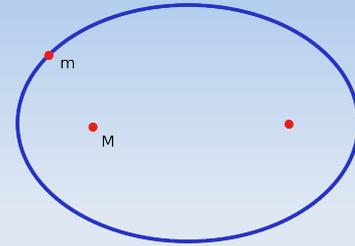


Johannes Kepler  
(1571-1630),  
Bildquelle: Wikipedia

## Die Keplerschen Gesetze (II)

1. Die Umlaufbahn eines Objekts ist eine Ellipse und das Schwerezentrum des Systems liegt in einem ihrer Brennpunkte
2. In gleichen Zeiten überstreicht der Fahrstrahl (Objekt – Schwerezentrum) gleiche Flächen.
3. Die Quadrate der Umlaufzeiten zweier Objekte entsprechen den Kuben ihrer großen Halbachsen:

$$(T_1 / T_2)^2 = (a_1 / a_2)^3$$

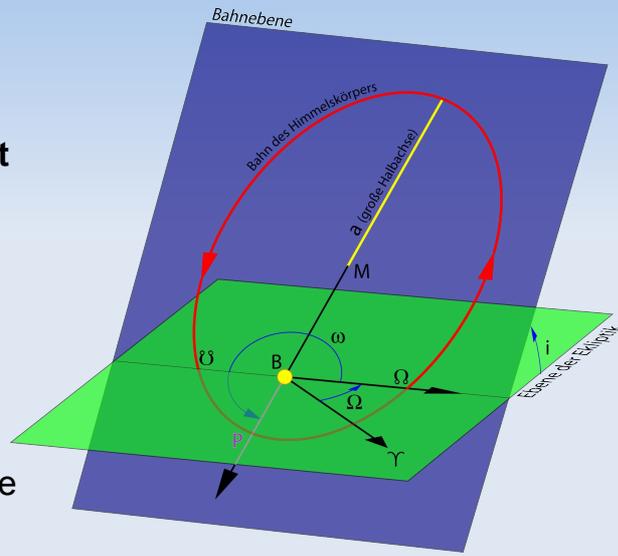


# Bahnelemente

- Variablen zur Beschreibung einer konkreten Umlaufbahn mit Hilfe der Keplerschen Gesetze
- Zahlreiche Bahnelemente definiert, allerdings viele redundant / voneinander abhängig
- 6 unabhängige Bahnelemente erforderlich für vollständige, eindeutige Bahnbeschreibung
- Positionsbestimmung für Objekt erfolgt durch Eingabe der Bahnelemente in Formeln, die aus den Keplerschen Gesetzen abgeleitet sind (hier nicht behandelt)

# 6 typische Bahnelemente zur Positionsberechnung

- **Große Halbachse** der Bahnellipse
- **Numerische Exzentrizität** der Bahnellipse
- **Mittlere ekliptikale Longitude** des Objekts
- **Longitude des aufsteigenden Knotens**
- **Longitude der Periapsis**
- **Inklination** der Bahnebene gegen die Ekliptikebene



Quelle der Zeichnung: Wikipedia, modifiziert

# Bahnberechnung: Von der Theorie zur Praxis

**Wichtig:** Alle Werte **ändern sich mit der Zeit** und beziehen sich deshalb immer auf einen gegebenen **festen Zeitpunkt**.

**Im folgenden Abschnitt:**

- Wie erhält man die Bahnelemente für einen gewünschten Zeitpunkt?
- Welche anderen Methoden zur Positionsberechnung gibt es?
- Welche Methoden setzen wir in der Simulation für welche Objekte ein?

11/28

Insbesondere die mittlere ekliptikale Longitude (Fortschritt des Objekts auf seiner Umlaufbahn) ändert sich sehr schnell

# Positionsberechnung für die Planeten

- Wir verwenden: *Variations séculaires des orbites planétaires 87 (VSOP87)*
- Von P. Bretagnon und G. Francou, Paris 1987
- Eine der genauesten Methoden zur Positionsberechnung für die acht großen Planeten
- Verfügbar in verschiedenen Versionen, welche Positionsangaben in unterschiedlichen Formaten liefern
- Wir verwenden Version, welche die Bahnelemente für gewünschten Zeitpunkt zurückliefert und aktualisieren so die Bahnelemente eines Objekts 1x pro Umlauf. Darauf basierend Positionsberechnung nach Kepler für jeden einzelnen Frame.

# Positionsberechnung für den Erdmond

- Wir haben für den Mond leider kein Verfahren zur Berechnung der Bahnelemente für gegebene Zeitpunkte (d.h. Ähnlich VSOP87) gefunden
- Häufig genutztes Verfahren *ELP2000* liefert lediglich direkt die Position des Mondes zurück
- Gezwungenermaßen verwenden wir auch *ELP2000*. Wir hätten aber lieber ein Verfahren, das Bahnelemente liefert, da diese für verschiedene Visualisierungen und Berechnungen sehr nützlich sind.

# Positionsberechnung für Monde anderer Planeten

- Prinzipiell: Sehr komplex wegen gravitativer Wechselwirkungen mit dem umkreisten Planeten (und anderen). Für die Monde jedes Planeten sind andere, spezielle Algorithmen erforderlich.
- Bisher haben wir keinen Mond ausser dem der Erde implementiert. Ephemeridenrechnungsverfahren für die Monde aller großen Planeten sind aber in freier Software wie *Stellarium* oder *Celestia* enthalten und könnten übernommen werden.

# Positionsberechnung für Asteroiden, Kometen und Raumfahrzeuge

- Oft starke und häufige Änderungen der Bahnelemente wegen geringer Masse (leichte gravitative Beeinflussbarkeit), z.B. durch
  - Gravitation von Sonne, Planeten und Monden
  - Sonnenwind
  - Aktive Kursänderungen (bei Raumfahrzeugen)
  - Reibung mit Atmosphäre (bei Satelliten in niedrigen Umlaufbahnen)
- Längerfristige Vorausberechnung von Bahnelementen bzw. Positionen daher praktisch nicht möglich
- Begrenzte Lösung: Online-Update der Bahnelemente von Bezugsquelle im Internet (implementiert z.B. in *KStars*)

# Räumliche Orientierung von Objekten im Planetensystem

# Räumliche Orientierung von Objekten im Planetensystem

- Zur vollständigen Beschreibung der räumlichen Lage eines Objekts gehört neben dem Ort auch die Orientierung im Raum.
- Für von uns modellierte Himmelskörper bedeutet dies:
  - Orientierung aller Achsen zu einem Referenzzeitpunkt
  - Zeitliche Veränderung der Achsenorientierungen

## Räumliche Orientierung konkret

- Richtung der **Rotationsachse** zu Referenzepoche J2000
- Richtungsvektor (Mittelpunkt – 0°Lon. / 0°Lat.) zu Referenzepoche J2000 (**Sternzeit** zu J2000)
- Zeitabhängige Änderung des Richtungsvektors der Rotationsachse (**Präzession**)
- Zeitabhängige Änderung des Richtungsvektors (Mittelpunkt – 0°Lon. / 0°Lat.) (**Rotationsgeschwindigkeit**)
- Quelle: *Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements 2006*

## Objekte ausserhalb des Planetensystems: Sterne und Sternbilder

## Die "Himmelskugel"

- Sterne sind **außerordentlich** weit vom Sonnensystem entfernt, ihre Entfernung wird daher **nicht** simuliert
- Alle Sterne sind auf einer Himmelskugel in gleichem Abstand zur Kamera platziert (Radius im Grunde egal)
- Himmelskugel hat stets die Kamera als Mittelpunkt, um die Illusion von großer (unendlicher) Entfernung zu erzeugen ("Skybox"-Konzept)
- Im OpenGL-Objektraum sind daher die Sterne i.d.R. Näher an der Kamera als die Objekte im Sonnensystem. Korrekte Überdeckung wird durch passende Wahl der Zeichenreihenfolge erreicht.

# Sternkataloge

- Laden von Sternkatalogen
  - modular konzipiert
  - Lade-Klassen für verschiedene Katalog-Formate
  - können gleichzeitig im Programm existieren
- Derzeit nur Klasse für den **Yale Bright Star Catalogue** (YBSC)
  - ~4000 von insgesamt ~6000 mit bloßem Auge sichtbaren Sterne

# Modellierte Sterneigenschaften

- Sternort (Position auf der Himmelkugel)
- Scheinbare Helligkeit (Magnitude) in einigen festen Abtufungen (noch keine "stetige Funktion implementiert)

# Konstellationen und Sternbilder

- Laden von Konstellation, wie bei Sternen, d.h. Unterstützung verschiedener Dateiformate
- momentane Implementierung
  - eigenes XML-Format
  - enthält Linien, sowie Namen der Konstellationen
  - Intern keine datenstrukturelle Verbindung von Sternen und Konstellationen
  - Konstellationslinien können auch ganz ohne geladene Sternenkataloge eingeblendet werden

## Sonstiges zur Implementierung

## Begrenzte Genauigkeit durch nicht berücksichtigte Faktoren...

- ... **bezüglich Berechnung der Beobachterposition:**
  - Planet ist keine perfekte Kugel (Abplattung durch Rotationsfliehkraft, Gebirge etc.)
  - Zeitliche Veränderung der Rotationsachsenorientierung nicht zu 100% berücksichtigt (Präzession ja, Nutation nein(?))
  - ... etc.
- ... **bezüglich Berechnung der Position von Objekten:**
  - Verzögerungen durch begrenzte Lichtgeschwindigkeit
  - Lichtablenkung durch große Massen (Raumkrümmung)
  - Lichtablenkung/-streuung durch Atmosphäre
  - ... etc.

# Programmarchitektur (I)

- Komplette implementiert in C++, massiv objektorientiert
- Grundsätzlich: Code für physikalische Simulation und Visualisierung getrennt (anders als z.B. bei Stellarium)
  - Erlaubt Implementierung unterschiedlicher Visualisierungsmethoden (z.b. Himmelskartenerzeugung als PNG, o.ä.)
  - Bisher nur interaktive OpenGL-Darstellung realisiert
- Modulare Architektur ermöglicht einfache Erweiterung und Austausch von Komponenten

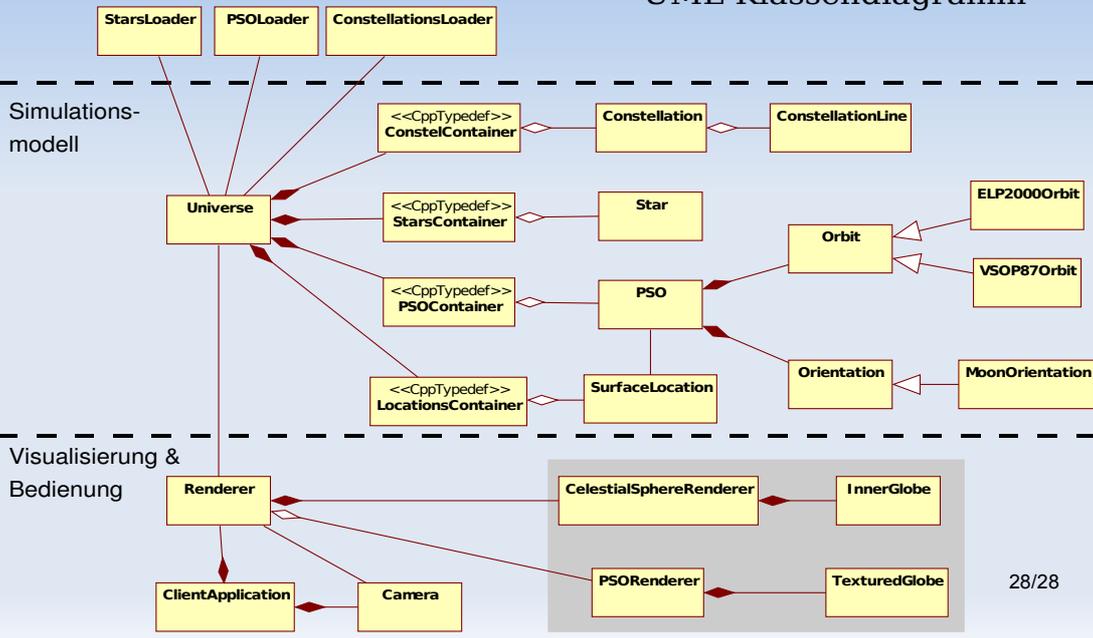
## Programmarchitektur (II)

- Integration der Gottorf-Globus-Texturen "minimalinvasiv"
- Keine Änderungen an Basiskomponenten, nur Ableitungen ohnehin vorhandener Klassen
- Code aus dem alten Gottorf-Globus-Programm im Wesentlichen unverändert übernommen

# UML-Klassendiagramm

Simulationsdatenquellen

Gottorfium: Vereinfachtes UML-Klassendiagramm



# Verbesserungsmöglichkeiten (I)

- **Genauigkeit des Himmelsmodells**
  - Eigenbewegung von Sternen
  - Lichtgeschwindigkeit
  - Verschiedene Spektralklassen / Sternfarben
- **Inhalte und Bedienung**
  - Nebel und Deep-Sky-Objekte (Galaxien)
  - Einblendung von Koordinatengittern
  - Möglichkeit zum Selektieren von Objekten und Einblenden von Informationen darüber

## Verbesserungsmöglichkeiten (II)

- **"Eye Candy"**
  - Atmosphäreeneffekte
  - "Virtuelle Texturen" / Texture-Tiling
  - Bump Mapping für Oberflächen
  - Sonnenblendeffekt / Lens-Flares

# Vergleichbare, einflussgebende Software

- *Stellarium*
  - <http://www.stellarium.org/>
- *Cartes du Ciel*
  - <http://www.stargazing.net/astro/c/>
- *Celestia*
  - <http://www.shatters.net/celestia/>
- *KStars*
  - <http://edu.kde.org/kstars/>

Ende



Vielen Dank für Ihre Aufmerksamkeit!

32/28