Unser Projekt

Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in der Astrophysik und wird nie mathematisch perfekt lösbar sein. Da die Beschleunigung auf einen Körper nicht konstant sind numerische Methoden zur Annäherung nötig. Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in der Astrophysik und wird nie mathematisch perfekt lösbar sein. Da die Beschleunigung auf einen Körper nicht konstant sind numerische Methoden zur Annäherung nötig.

Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in der Astrophysik und wird nie mathematisch perfekt lösbar sein. Da die Beschleunigung auf einen Körper nicht konstant sind numerische Methoden zur Annäherung nötig.

Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in der Astrophysik und wird nie mathematisch perfekt lösbar sein. Da die Beschleunigung auf einen Körper nicht konstant sind numerische Methoden zur Annäherung nötig.

Idee und motivation

Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in der Astrophysik und wird nie mathematisch perfekt lösbar sein. Da die Beschleunigung auf einen Körper nicht konstant sind numerische Methoden zur Annäherung nötig. Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in der Astrophysik und wird nie mathematisch perfekt lösbar sein. Da die Beschleunigung auf einen Körper nicht konstant sind numerische Methoden zur Annäherung nötig.

Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in der Astrophysik und wird nie mathematisch perfekt lösbar sein. Da die Beschleunigung auf einen Körper nicht konstant sind numerische Methoden zur Annäherung nötig.

Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in

Berechnung der Gravitationskraft

Um Graviton zwischen vielen unabhängigen Körpern zu berechnen muss die kraft zwischen jedem Objekt zu jedem Objekt berechnet werden, für diese N-Body Simulation gibt es keine einfache Formel die angibt wann sich welcher Körper wo befindet. Diese N-Body Problem ist ein bekanntes Problem in der Astrophysik und wird nie mathematisch perfekt lösbar sein. Da die Beschleunigung auf einen Körper nicht konstant sind numerische Methoden zur Annäherung nötig.

Um die kraft die auf einen Körper wirkt auszurechnen benutzten wir Newtons Gravitationsgesetz:

Mit der kraft die auf einen Körper wirkt können wir seine Beschleunigung ausrechnen:

Numerische Methoden

Da die Beschleunigung a nicht konstant ist müssen wir sie numerisch integrieren. Dazu gibt es etliche Methoden wir haben die weitverbeitesten getestet und untersucht. Dabei war uns vor allem die Geschwindigkeit wichtig da wir natürlich nur begrenz Rechenleistung zur Verfügung haben.

Semi-implicit Euler method (2. Order)

Wobei der zeitschritt ist bei dem wir annehmen die Beschleunigung sei konstant da wir nur begrenzt Rechenleistung haben.

Die Euler Methode ist die einfachste und auch die schnellste, doch sie ist auch recht ungenau was schon nach einer kurzen Zeit eine extreme Auswirkung auf unsere Simulation haben kann.

Runge-Kutta method (4. Order)

Die Runge-Kutta Methode(RK4) ist deutlich genauer doch sie ist auch deutlich rechenintensiver, was sie für unser Vorhaben ehr unpassend macht. Als wir am Anfang aber noch nur unser Sonnensystem simuliert haben war sie genau und hat zu am wenigsten Energie Verlust geführt. Da die Runge-Kutta auch noch erweiterbar ist so dass sie Ordnung 8 oder mehr hat könnten wir sie in Zukunft wieder einsetzten, wenn wir die Berechnungen an ein Rechenzentrum abgeben würden um somit genauere Ergebnisse zu bekommen.

Leapfrog method (2. Order)

Kick drift kick (KDK):

Drift kick drift (DKD):

Die KDK-Methode ist laut unseren Auswertungen am genausten benötigt aber auch mehr Rechenleistung als die DKD-Methode da zweimal die Beschleunigung ausgerechnet werden muss. Trotzdem haben wir für die Galaxien Simulationen meistens, dass KDK-leapfrog verfahren benutzt, da es das beste Verhältnis zwischen Genauigkeit und rechen Intensivität bietet

Vergleich der Methoden

Um die Genauigkeit der Methoden zu testen haben wir den Energie Verlust berechnet und diesen verglichen.

Die folgenden Energie Verluste wurden mit Daten von unserem Sonnensystem berechnet. Diese Daten haben wir vom NASA Horizons System, dass alle Positionen und Geschwindigkeiten von den planten in unserem Sonnensystem zu einem bestimmten Zeitpunkt bereitstellt.

Methoden bei unterschiedlichem bei gleicher gesamten zeit:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Euler: |  |  |  | Runge Kutta (RK4): | |  |
| 𝑛𝑡 | ∆𝑡 |  |  |  |  |  |
| 10 | 100000 | -8,64E+30 |  | 10 | 100000 | -2,29E+31 |
| 100 | 10000 | -1,05E+30 |  | 100 | 10000 | -2,79E+30 |
| 1000 | 1000 | -1,08E+29 |  | 1000 | 1000 | -2,84E+29 |
| 10000 | 100 | -1,11E+28 |  | 10000 | 100 | -2,88E+28 |
| 100000 | 10 | -1,39E+27 |  | 100000 | 10 | -3,17E+27 |
| 1000000 | 1 | -4,23E+26 |  | 1000000 | 1 | -6,06E+26 |
|  |  |  |  |  |  |  |
| DKD leapfrog: | |  |  | KDK leapfrog: | |  |
|  |  |  |  |  |  |  |
| 10 | 100000 | 3,30E+27 |  | 10 | 100000 | 1,90E+25 |
| 100 | 10000 | -2,51E+26 |  | 100 | 10000 | 5,22E+24 |
| 1000 | 1000 | -3,14E+26 |  | 1000 | 1000 | 7,66E+23 |
| 10000 | 100 | -3,16E+26 |  | 10000 | 100 | 6,42E+22 |
| 100000 | 10 | -3,16E+26 |  | 100000 | 10 | 6,27E+21 |
| 1000000 | 1 | -3,16E+26 |  | 1000000 | 1 | 4,72E+20 |

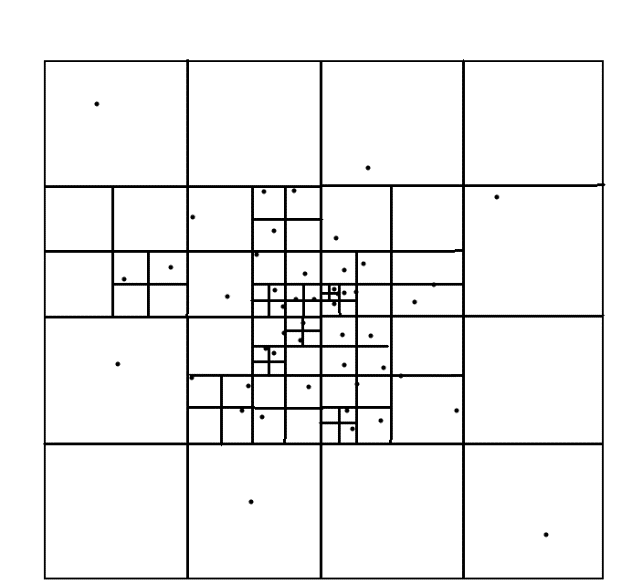
Wie in der Auswertung zu sehen schneidet KDK leapfrog am besten ab, Runge Kutta ist hier sehr ungenau, weil so groß ist, denn Runge Kutta wird erst ab einem sehr kleinem die beste Option. Nach dieser Auswertung haben wir für die Finalen Berechnungen immer KDK leapfrog als numerische Methode benutzt und zum schnellen testen Euler.

Barnes Hut Algorithmus

Da die Gravitationskraft für jeden Körper in der Simulation mit jedem Körper berechnet werden muss ist die Laufzeitkomplexität was zur folge hat, dass größere und damit auch realistischere mengen von Körpern zu einem quadratischen Anstieg des rechen Aufwands führen. Bei Hunderttausend Objekten und zehntausend zeitschritten wären es also schon 100 Billionen Berechnungen was natürlich mit unseren Rechnern einfach zu lange dauern würde. Um dieses Problem zu beheben haben wir den Barnes Hut Algorithmus implementiert den mit diesem können wir die Laufzeitkomplexität auf senken was einen riesigen unterschied macht:

Um dies umzusetzen wird der Raum in Würfel unterteilt und alle Objekte in einer Baum Struktur verteilt. Dabei wird der Würfel, im Baum also die node, so lange aufgeteilt bis auf der untersten Ebene in jeder node nur ein Objekt ist. Dieser baum wird bei jedem zeitschritt mit den neuen Objekt Positionen neu gebaut damit die Würfel auch bei Verschiebungen die richtigen Objekte in sich haben.

Außerdem wird der massenschwerpunkt der node berechnet was die Genauigkeit des Algorithmus um ein Vielfaches verbessert.



Nachdem der Baum gebaut wurde schaut der Algorithmus für jeden Körper zu jeder Obersten Node:

= Schwellwert Theta ( desto näher zu 0 desto genauer)

r = Radius der aktuellen node

d = Distanz von Körper zu Node Massenschwerpunkt

Baum aus würfeln in 2D dargestellt

Falls

-> Berechne Kraft zur Node

Sonnst

-> gehe tiefer in die nächsten 8 nodes rein um die Genauigkeit zu erhöhen

Da der Algorithmus rekursiv ist gibt es die Möglichkeit eines Stack Overflows, deswegen gibt es eine maximale tiefe im Moment 200 diese hat aber kein Einfluss auf die Genauigkeit der Simulation da sie nur im äußersten Notfall eingreift und der Distanz unterschied zu vernachlässigen ist.

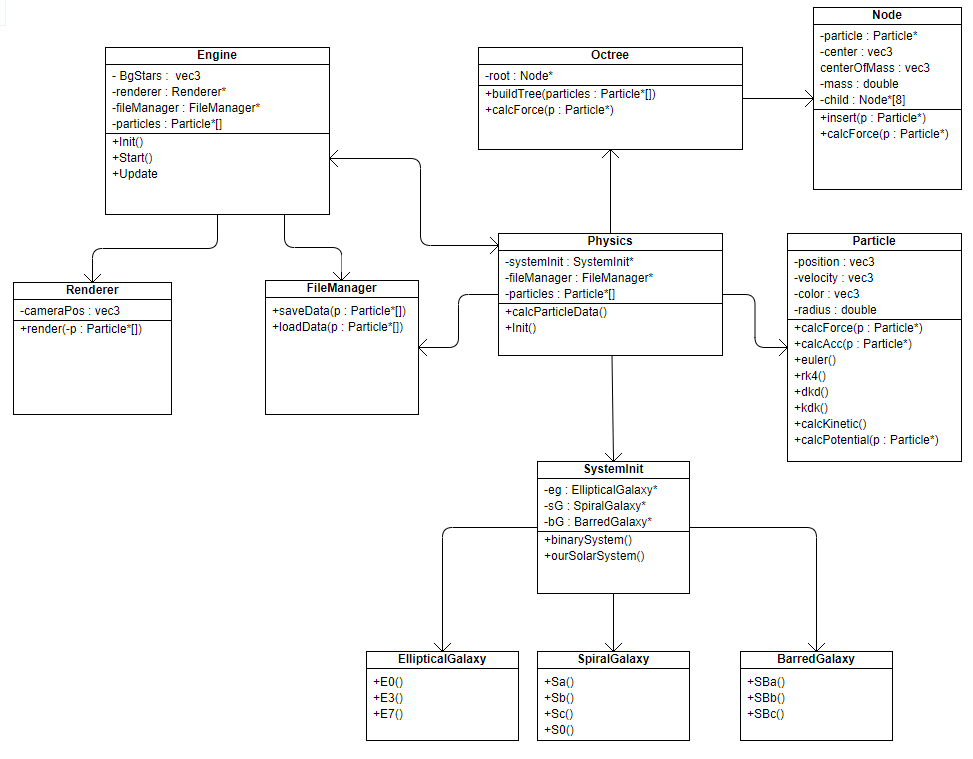
Software Struktur

Um die simulierten daten auch ansprechend anzuzeigen haben wir uns für OpenGL entschieden. Dazu haben wir eine Rendering Engine geschrieben die die Partikel in 3D raum anzeigt.

Verwendete Bibliotheken:

* GLEW (OpenGL Extension Wrangler Library) für die vereinfachte Nutzung von OpenGL Funktionen.
* GLFW (Graphics Library Framework) für Fenster und Kontext Erstellung
* GLM (OpenGL Mathematics) für die Double Vektoren und Matrizen

Vereinfachtes UML mit wichtigsten Attributen und Klassen:



Am Anfang ruft Physics SystemInit auf um einmal die Start Positionen/ Geschwindigkeiten aller Objekte festzulegen. Dabei können entweder unser Sonnensystem oder Ganze Galaxien nach Hubbles Klassifikation Ausgewählt werden.

Nach dem der Startposition berechnet Physics mit dem Barnes Hut Algorithmus(Octree) die Positionen für alle Körper und übergibt diesen dann an den File Manager dieser speichert die Daten in binär als Datei. Die Engine kann darauf hin sich die Daten zum aktuellen Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen. Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen.

Unser Gesamte code ist auf GitHub öffentlich einsehbar: <https://github.com/Philip-Spaeth/simulation>

Synchronisierung der Berechnungen

Um den Prozess der Berechnungen zu beschleunigen, wurden Multithreading ins Programm eingebaut. Multithreading ist die synchrone Verteilung der Berechnungen und abzuarbeitenden Aufgaben auf alle Kerne des Prozessors. Dies beschleunigt den Vorgang der Berechnungen um die Anzahl der Kerne. Ein normaler PC hat im Durchschnitt 4-8 Kern. Ein Server hingegen hat häufig 32-128 Kerne, was die Berechnung um ein Vielfaches Beschleunigt.

Ein weiterer Ansatz wäre es die Berechnungen auf die Grafikkarte zu übertragen, da diese im Durchschnitt 10-20-mal so schnell rechnet wie ein Prozessor.

Eine weitere Überlegung war die Berechnungen auf mehrere Computer zu verteilen. Ein Computer hätte bei diesem System die Fäden in der Hand und steuert die anderen Computer. Die Computer wären über die Netzwerk Schnittstellen miteinander verbunden.

Beide dieser Ansätze wurden aber bis her noch nicht weiterverfolgt, da die Variante mit einem Server am schnellsten und einfachsten umzusetzen ist. Außerdem ist es schwieriger genug Computer zu bekommen, auf welchen zur gleichen Zeit gerechnet werden kann.

Simulation von Elliptischen Galaxien

Anfangs haben wir uns auf Elliptische Galaxien Fokussiert da sie wegen ihrer fehlenden spiral arme einfacher zu initialisieren sind. „In den meisten Elliptischen Galaxien ist eher weniger bis gar keine dunkle Materie enthalten welche sie einfach zu Simulieren macht da wir die Masse einfach je nach Helligkeit verteilen.“(Stimmt vielleicht nicht)

Galaxie Messier 89:

Nach der Hubble Klassifikation eine E0 Galaxie, also praktisch Kugelförmig. Die Massen Verteilung lässt sich An Hand der Helligkeit bestimmen. Und nimmt hier unserem Fall Richtung mit linear zu.



Elliptische Galaxie mit der Größe, Gesamt Masse und Verteilung von Messier 89

Nach dem der Startposition berechnet Physics mit dem Barnes Hut Algorithmus(Octree) die Positionen für alle Körper und übergibt diesen dann an den File Manager dieser speichert die Daten in binär als Datei. Die Engine kann darauf hin sich die Daten zum aktuellen Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen. Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen.

*Nach dem der Startposition berechnet Physics mit dem Barnes Hut Algorithmus(Octree) die Positionen für alle Körper und übergibt diesen dann an den File Manager dieser speichert die Daten in binär als Datei. Die Engine kann darauf hin sich die Daten zum aktuellen Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im* *system besser zu verstehen. Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen.*

Massen Verteilung:

*Nach dem der Startposition berechnet Physics mit dem Barnes Hut Algorithmus(Octree) die Positionen für alle Körper und übergibt diesen dann an den File Manager dieser speichert die Daten in binär als Datei. Die Engine kann darauf hin sich die Daten zum aktuellen Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen. Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen.*

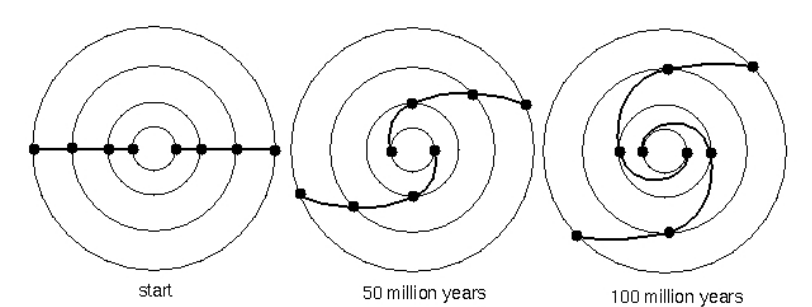
*Geschwindigkeit Von innen nach außen:*

*Nach dem der Startposition berechnet Physics mit dem Barnes Hut Algorithmus(Octree) die Positionen für alle Körper und übergibt diesen dann an den File Manager dieser speichert die Daten in binär als Datei. Die Engine kann darauf hin sich die Daten zum aktuellen Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen. Zeitpunkt wieder rausladen was uns die uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser*

Simulation von Spiral Galaxien

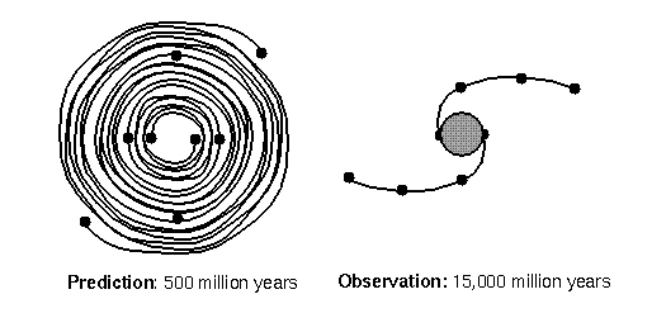
Um nun Spiral Galaxien zu Simulieren müssen wir neue Ansätze verfolgen da die Masse nicht einfach bei allen winkeln gleich verteilt ist, sondern es Spiralarme gibt in denen mehr Masse ist. Doch wenn man einfach versucht die Start Positionen der Sterne so festzulegen wie sie beobachtbar sind und sie dann auf einem relativ runden Orbit um das Zentrum kreisen lässt wie bei der elliptischen Galaxie stößt man relativ schnell auf das Problem das auch wenn wir die spiral arme am Anfang richtig initialisieren sich diese schon nach wenigen Umläufen komplett auflöst. Dadurch dass sich die Sterne näher am Zentrum wegen der höheren kraft schneller bewegen drehen sich die arme sehr schnelle auf so dass schon nach 200 Millionen Jahren keine arme mehr erkennbar wären.

Dieses Problem wird in der Astronomie Winding Problem genannt:



<https://faculty.washington.edu/ivezic/Teaching/Astr509/lecture12.pdf>

Die Spiralarme müssten sich also schon längst aufgelöst haben da Galaxien ja schon mehre Milliarden Jahre alt sind:



<https://faculty.washington.edu/ivezic/Teaching/Astr509/lecture12.pdf>

Um dieses Problem zu lösen haben C.C. Lin und Frank Shu Mitte der 1960er die Density wave theory entwickelt welche auch heute immer noch das beste Model ist um die Bewegung von spiral armen zu beschreiben.

Density wave theory

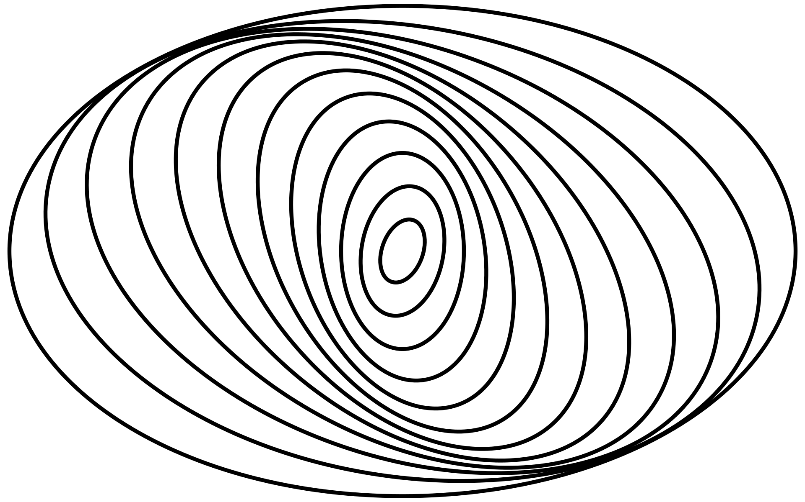
Sie nimmt an, dass in den helleren Regionen der Galaxie die spiral arme zu sehen sind da sich an diesen stellen die Orbits am nächsten sind und dadurch dichtewellen entstehen. Um dies zu erreichen müssen alle Orbits von innen nach außen immer ein bisschen gedreht werden um diese dichte wellen zu erzeugen.

Also muss der Orbit gedreht werden, wenn wir weiter nach außengehen:

r: Distanz zum Zentrum

A: Die Amplitude, die die Größe der Drehung repräsentiert.

k: Eine Konstante, die die Steilheit der Spiralstruktur beeinflusst.



Dadurch können wir jetzt auch Spiral Galaxien realistisch simulieren ohne dass sich die spirale Struktur auflöst.

Messier 74:

Nach der Hubble Klassifikation eine SA(s)c Galaxie, also eine spirale Galaxie mit einer stark ausgeprägten Spiralstruktur.

Simulation mit Density wave theory ohne dunkle Materie:



Spirale Galaxie mit der Größe, Gesamt Masse und Verteilung von Messier 74

Modifizierte Density wave theory für barred Galaxien

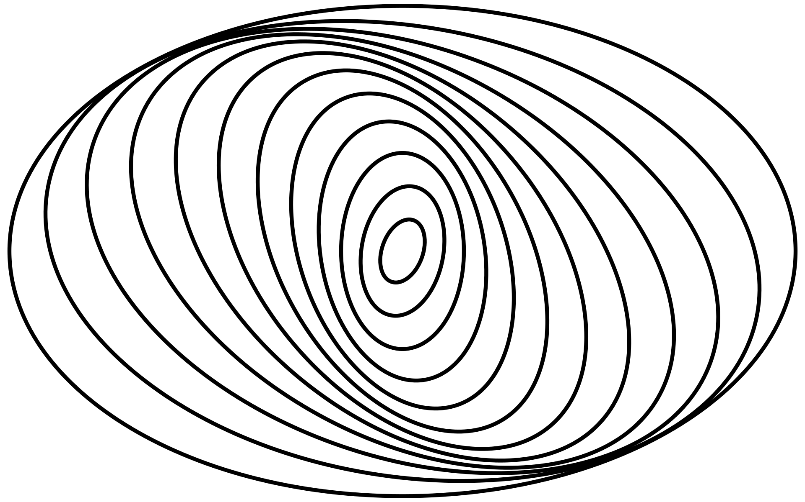
Sie nimmt an, dass in den helleren Regionen der Galaxie die spiral arme zu sehen sind da sich an diesen stellen die Orbits am nächsten sind und dadurch dichtewellen entstehen. Um dies zu erreichen müssen alle Orbits von innen nach außen immer ein bisschen gedreht werden um diese dichte wellen zu erzeugen.

Also muss der Orbit gedreht werden, wenn wir weiter nach außengehen:

r: Distanz zum Zentrum

A: Die Amplitude, die die Größe der Drehung repräsentiert.

k: Eine Konstante, die die Steilheit der Spiralstruktur beeinflusst.



Dadurch können wir jetzt auch Spiral Galaxien realistisch simulieren ohne dass sich die spirale Struktur auflöst.

Messier 74:

Nach der Hubble Klassifikation eine SA(s)c Galaxie, also eine spirale Galaxie mit einer stark ausgeprägten Spiralstruktur.

Simulation mit Density wave theory ohne dunkle Materie:



Spirale Galaxie mit der Größe, Gesamt Masse und Verteilung von Messier 74

Verteilung dunkler Materie in der Milchstraße

Um nun Spiral Galaxien zu simulieren müssen wir neue Ansätze verfolgen da die Masse nicht einfach bei allen Winkeln gleich verteilt ist, sondern es Spiralarme gibt in denen mehr Masse ist. Doch wenn man einfach versucht die Start Positionen der Sterne so festzulegen wie sie beobachtbar sind und sie dann auf einem relativ runden Orbit um das Zentrum kreisen lässt wie bei der elliptischen Galaxie stößt man relativ schnell auf das Problem das auch wenn wir die spiral arme am Anfang richtig initialisieren sich diese schon nach wenigen Umläufen komplett auflöst. Dadurch dass sich die Sterne näher am Zentrum wegen der höheren kraft schneller bewegen drehen sich die arme sehr schnelle auf so dass schon nach 200 Millionen Jahren keine arme mehr erkennbar wären.

Dieses Problem wird in der Astronomie Winding Problem genannt:

*Geschwindigkeit Von innen nach außen:*

*Nach dem der Startposition berechnet Physics mit dem Barnes Hut Algorithmus(Octree) die Positionen für alle Körper und übergibt diesen dann an den File Manager dieser speichert die Daten in binär als Datei. Die Engine kann darauf hin sich die Daten zum aktuellen Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen. Zeitpunkt wieder rausladen was uns die uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser*

*Geschwindigkeit Von innen nach außen:*

*Nach dem der Startposition berechnet Physics mit dem Barnes Hut Algorithmus(Octree) die Positionen für alle Körper und übergibt diesen dann an den File Manager dieser speichert die Daten in binär als Datei. Die Engine kann darauf hin sich die Daten zum aktuellen Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen. Zeitpunkt wieder rausladen was uns die uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser*

*Geschwindigkeit Von innen nach außen:*

*Nach dem der Startposition berechnet Physics mit dem Barnes Hut Algorithmus(Octree) die Positionen für alle Körper und übergibt diesen dann an den File Manager dieser speichert die Daten in binär als Datei. Die Engine kann darauf hin sich die Daten zum aktuellen Zeitpunkt wieder rausladen was uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser zu verstehen. Zeitpunkt wieder rausladen was uns die uns die Möglichkeit gibt die berechnete Zeit vor und zurück zu spulen um die Veränderung im system besser*

Quellen

Um nun Spiral Galaxien zu simulieren müssen wir neue Ansätze verfolgen da die Masse nicht einfach bei allen Winkeln gleich verteilt ist, sondern es Spiralarme gibt in denen mehr Masse ist. Doch wenn man einfach versucht die Start Positionen der Sterne so festzulegen wie sie beobachtbar sind und sie dann auf einem relativ runden Orbit um das Zentrum kreisen lässt wie bei der elliptischen Galaxie stößt man relativ schnell auf das Problem das auch wenn wir die spiral arme am Anfang richtig initialisieren sich diese schon nach wenigen Umläufen komplett auflöst. Dadurch dass sich die Sterne näher am Zentrum wegen der höheren kraft schneller bewegen drehen sich die arme sehr schnelle auf so dass schon nach 200 Millionen Jahren keine arme mehr erkennbar wären.