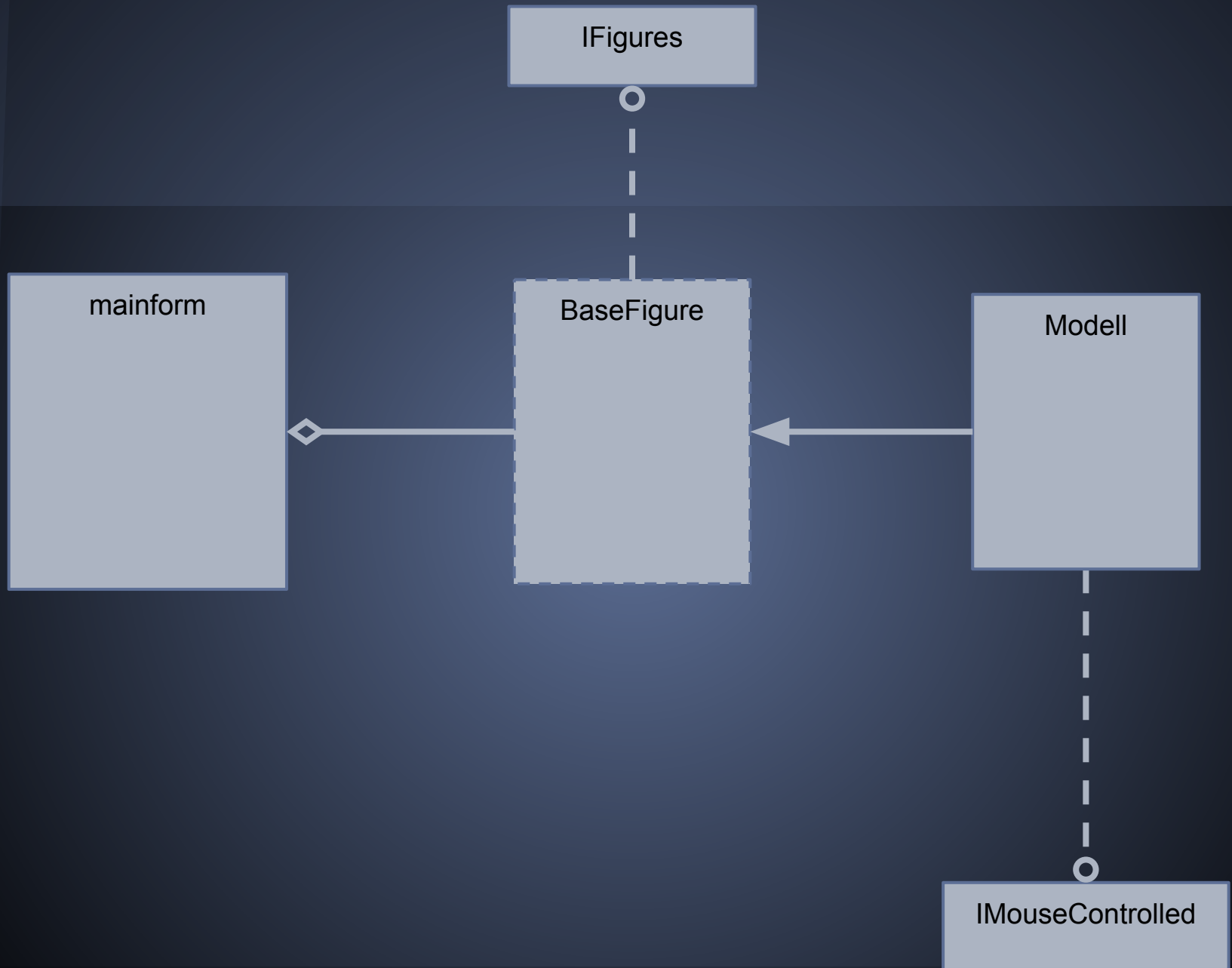


3DLab - Modelle

OpenGL in C#

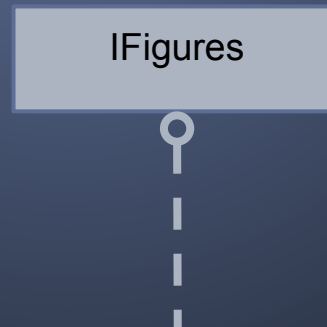
Was ist 3DLab?

- Darstellen von Modellen auf OpenGL
- eigene Modelle mit simplen Interface
- Einbinden neuer Modelle leider noch zur Entwurfszeit



Das Interface IFigures

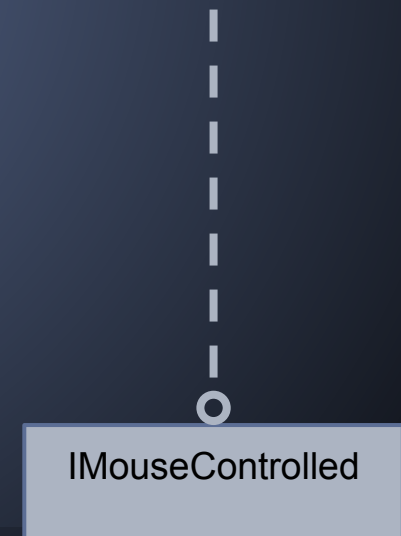
- Init, Tick, Draw, Keypressed
- Init initialisiert das Modell
- Tick simuliert das Modell
- Draw zeichnet das Modell
- Keypressed ermöglicht das Interagieren mit dem Modell



Das Interface Mousecontrolled

Ermöglicht das Reagieren auf Mauseingaben im Modell

- mx,my, Mousebutton
- mx Mausbewegung auf X-Achse
- my Mausbewegung auf Y-Achse
- Mousebutton gibt Klicks weiter



BaseFigures (abstract)

Implementiert die Grundfunktionalität der Modelle und realisiert dabei das Interface IFigures

- Cam - Enthält das Kameraobjekt
- Enabled - Ist Modell aktiviert?
- Visible - Ist Modell sichtbar?



Steuerung

Zur Steuerung der Ansicht wird das Kameraobjekt durch den Raum bewegt:

W,S,A,D - Vorwärts, Rückwärts, Seitwärts

Q,E, Mausrad - hoch, runter auf der z-Achse

Mausbewegung - Blickrichtung

Strg - Bewegung * 1000

Alt - Bewegung * 0.1

Shift - Bewegung * 100

Sonstige Steuerungen:

F11, F12 - Module vor/zurück schalten

F3 - Schaltet durch Lichtmodi (Sonnen als Lichtquellen, keine Lichtquellen)

J - Aktiviert/Deaktiviert Skybox (Weltraumumgebung)

ESC - Beendet Programm

Das Staubmodell

Eine Partikelsimulation

Das Staubmodell

- Simuliert Gravitation zwischen Staubpartikeln im Raum
- Staubpartikel verschmelzen bei Kontakt und bilden größere Staubpartikel
- Bilden dann eine Art Planetensysteme

Klassenstruktur - Staubmodell

- PPoint - Das Schweifobjekt
- o2Object - Das Staubobjekt
- o2Sun - Große Staubkörner leuchten
- PlanetAdvanced3 - Controller

PPoint

Sich bewegende Partikel ziehen einen Schweif hinter sich her. Dieser zeigt die Bewegung und macht die Partikel besser sichtbar.

- x, y, z - Position im Raum
- Alpha - Schweifpunkte haben eine Lebenszeit und verschwinden langsam

o2Object

Mass, Size, Farbe, Position, Direction, DrawTrace, Moveobject

- Mass, Size - Die Darstellungsgröße ist Abhängig von der Masse
- Direction - In diese Richtung bewegt sich der Partikel
- Drawtrace - Erstellt Schweifobjekte
- Moveobject - verändert Bewegungsrichtung beeinflusst durch alle anderen Objekte

o2Sun

ein o2Object, welches zusätzlich als Lichtquelle dient. Die erste entstandene Sonne ist die Hauptsonne. Diese lässt sich zentrieren und weitere hinzugefügte Partikel werden mit der ersten kosmischen Geschwindigkeit bezüglich der Sonne gespawnt und bewegen sich somit (zumindest anfänglich) auf einer Kreisbahn um sie.

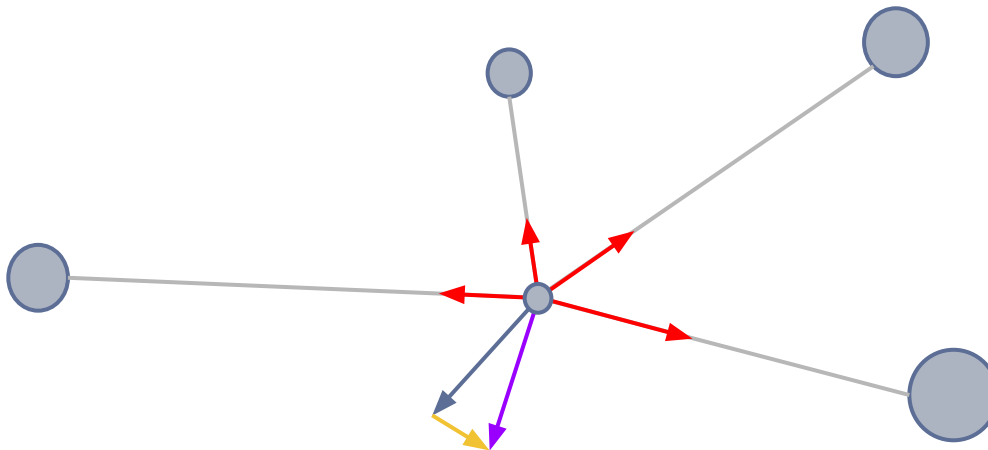
PlanetAdvanced3

Beinhaltet die Steuerung und die Kontainer für die Objekte - Partikel. Es verwaltet also die Partikel und gibt die Ticks aus dem Mainthread an die o2Objects weiter. 4 Threads kümmern sich dann um die Bewegung der Partikel.

Bewegungsrichtung

Die Bewegungsrichtung wird durch alle anderen Objekte beeinflusst. Der Abstand und die Masse haben einen Einfluss auf die Beschleunigung in zueinander.

Die Bewegungsrichtung wird dann auf die Position addiert und der Partikel damit bewegt.



Die roten Vektoren stellen die Beschleunigung in eine Richtung dar. Diese werden aufsummiert (gelber Pfeil) und zur aktuellen Bewegungsrichtung (schwarzer Pfeil) addiert und bilden so die neue Bewegungsrichtung (violetter Pfeil).

Die Anziehungskraft



$$a = -G * 2.Mass / |\overrightarrow{P1P2}|^3 * \overrightarrow{P1P2}$$

G: Gravitationskonstante

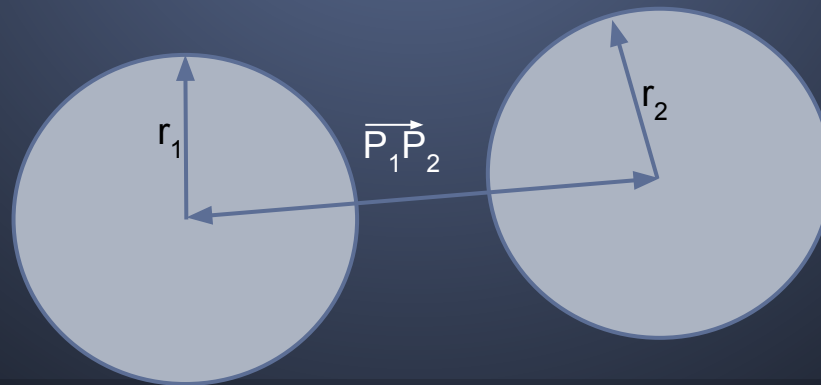
a: Beschleunigung von 1 in Richtung Objekt 2

Kollisionsberechnung

Da die Partikel als Kugeln dargestellt werden, kann man Kollisionen schnell und einfach über den Abstand der Mittelpunkte und die Radien berechnen.

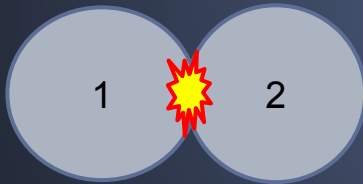
Wenn der Abstand der Mittelpunkte kleiner als die Summe der Radien ist, sind die Partikel kollidiert:

$$r_1 + r_2 < |\vec{P_1 P_2}|$$



Kollision

Wenn 2 Partikel kollidieren, wird die Masse des Kleineren dem Größeren hinzugefügt. Dabei wird auch der Impuls des Kleineren auf den Größeren übergehen.



$$\text{impulse} = \text{Direction}_1 * \text{Mass}_1 + \text{Direction}_2 * \text{Mass}_2$$

$$\text{Mass}_1 = \text{Mass}_1 + \text{Mass}_2$$

$$\text{Direction}_1 = \text{impulse} / \text{Mass}_1$$

Steuerung

- Linke Maus, 1,2 , 5, Mittlere Maus - Added Partikel (1=100, 5=50, 2=200)
- Strg-P - aktiviert/deaktiviert Schweif
- P - Simulation Stop/reinitialisiert Modell
- O - Simulation Halt/Play
- Backspace - Setzt Hauptsonne in den Ursprung
- Strg-Backspace - Hält Sonne im Ursprung fest
- +,- Simulationsgeschwindigkeit (größere Schritte mit Strg)

Die Tangentialebene

Funktionen im \mathbb{R}^3

Die Funktion

$$f(x) = x * \sin(y) + y * \sin(x)$$

Dargestellt wird hier die Funktion im \mathbb{R}^3 und die Tangentialebene in einem beliebigen Punkt auf der Funktion.

Die Tangentialebene

$$\begin{aligned} T(x,y) = & f(x,y)/dx * (x - x_0) \\ & + f(x,y)/dy * (y - y_0) \\ & + f(x,y) \end{aligned}$$

Die Tangentialebene wird im Programm transparent und nur in der Nähe von $P_0 (x_0, y_0, f(x_0, y_0))$ dargestellt und der Punkt durch einen Teil der Gerade durch P_0 parallel zur z-Achse dargestellt.

Bedienung

Der Punkt P_0 an dem die Tangentialebene anliegt kann mit gedrückter rechter Maustaste und Bewegung der Maus verschoben werden.

Das Partikelfeld

Punkte folgen Funktionen im \mathbb{R}^3

Partikelfeld

Die einzelnen Partikel bewegen sich mithilfe einer Bildungsvorschrift bezüglich der letzten Position. Es sind 6 Bildungsvorschriften implementiert die sich auch kombinieren lassen.

Funktion 1

$$x = x + \sin(z * 0.01) - \sin(y * 0.01)$$

$$y = y + \sin(x * 0.01) - \sin(z * 0.01)$$

$$z = z + \sin(y * 0.01) - \sin(x * 0.01)$$

Funktion 2

$$x = x + 5 * \cos(0.02 * y)$$

$$y = y + 5 * \cos(0.02 * x)$$

$$z = z + 0.2 * \cos(0.025 * y)$$

Funktion 3

$$x = x + \sin(z * 0.01) - \tan(y * 0.01)$$

$$y = y + \sin(x * 0.01) - \tan(z * 0.01)$$

$$z = z + \sin(y * 0.01) - \tan(x * 0.01)$$

Funktion 4

$$x = x + \cos(z * 0.01) - \cos(y * 0.01)$$

$$y = y + \cos(x * 0.01) - \cos(z * 0.01)$$

$$z = z + \cos(y * 0.01) - \cos(x * 0.01)$$

Funktion 5

$$x = x + \tan(z * 0.01) - \tan(y * 0.01)$$

$$y = y + \tan(x * 0.01) - \tan(z * 0.01)$$

$$z = z + \tan(y * 0.01) - \tan(x * 0.01)$$

Funktion 6

$$x = x + \tan(z * 0.01) - \cos(y * 0.01)$$

$$y = y$$

$$z = z + \cos(y * 0.01) - \tan(x * 0.01)$$

Bedienung

Die Funktionen an denen sich die Punkte bewegen, lassen sich einzeln ein und ausschalten. Es können auch alle gleichzeitig eingeschaltet werden.

Auf jeden Punkt wird zu jedem Durchlauf die Funktion angewendet und dieser damit bewegt.

Bedienung 2

1,2,3,4,5,6 - Funktionen ein/aus

M - Darstellung der Partikel (Punkt, Flächen, Würfel)

L - Initialisiert Punktfeld (als Ebene)

B - Initialisiert Punktfeld (als Quader)