

.NET

Praktikum 4

Verloren im Weltall

Die Bewegungen der Planeten werden durch die sogenannten Keplerschen Gesetze bestimmt. Diese gehen aber von einem Zentralgestirn aus, in dem der Grossteil der Masse konzentriert ist, was für unser Sonnensystem ja auch der Fall ist. Gäbe es dieses Zentralgestirn nicht, kämen kompliziertere Formeln zur Anwendung. Ab drei Massen existiert keine geschlossene Formel mehr, sondern die Bewegungen lassen sich nur noch numerisch bestimmen. Es sollen hier die Planetenbahnen ohne die Sonne berechnet werden. Wir ignorieren den Umstand, dass ohne diese es vermutlich niemanden geben würde, den diese Berechnung interessiert.



Aufgabe 1

Definieren Sie einen geeigneten (abstrakten) Basistyp *Orb*, den Sie für das Zeichnen der Himmelskörper und für die Berechnung der Bewegungen verwenden können. Leiten Sie von diesem Himmelskörper die Klassen *Planet* und *Spaceship* ab (unterschiedliche Klassen, weil wir Spaceship später noch erweitern).

Aufgabe 2

Zeichnen Sie die 3 Planeten und das Raumschiff an vordefinierten Positionen. Sie können dafür die Klassen aus dem Namespace `System.Drawing` verwenden.

Hinweise:

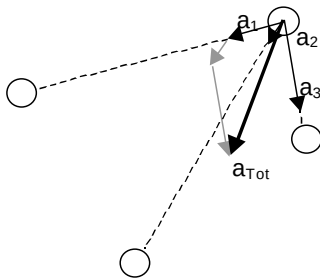
- Verwenden Sie Ihre IDE um ein Gerüst einer (Windows-)Anwendung zu erstellen
- Ähnlich wie bei Java hat jedes Frame eine Methode `OnPaint`, die überschrieben werden kann.: `protected override void OnPaint(PaintEventArgs e)`
- `Graphics` stellt einen Satz von Methoden zur Verfügung, mittels denen gezeichnet werden kann. Diese wird beim Aufruf der `OnPaint` Methode als Feld des Ereignisses mitgegeben. (`e.Graphics`)
- Mittels der Klasse `Bitmap` können Bilder geladen und dargestellt werden.

Aufgabe 3

Die Bewegungen sollen berechnet und angezeigt werden. Die Anziehungskraft zwischen zwei Massen berechnet sich aus (wie Ihnen sicherlich aus der Physik noch geläufig ist) $F = G * (M_1 * M_2) / r^2$. Die für eine Beschleunigung einer Masse notwendige Kraft berechnet sich aus $F = a * M$. Daraus lassen sich die resultierenden Beschleunigungen der einzelnen Massen in unserem System berechnen. Die resultierende Beschleunigung lässt sich einfach durch vektorielle Addition der einzelnen Beschleunigungskomponenten berechnen. Für die Berechnung der Beschleunigungskomponente hat sich folgende Formel bewährt:

$$\vec{a}_1 = |a_1| * \frac{\vec{r}_1}{|\vec{r}_1|}$$

Einheitsvektor in Richtung r



Gehen Sie im Beispiel von einem relativen Massenverhältnis der Planeten von 100 (Jupiter) zu 5 (Mars) zu 4 (Merkur) aus. Die Enterprise habe (wegen des Warp Antriebes und der geladenen Antimaterie) eine für ein Raumschiff enorme relative Masse von 1.

Hinweise:

- Die Gesamtbeschleunigung einer Masse setzt sich aus den Einzelbeschleunigungen zusammen. Achten Sie aber dabei, dass sie zuerst **sämtliche** Beschleunigungen und neuen Geschwindigkeiten berechnen und anschliessend die Massenpunkte verschieben (zwei Durchläufe).
- Sie können Ihr Vektorpaket wiederverwenden und um die Multiplikation (zwei mögliche Operandenreihenfolgen beachten!) und Division mit einem Skalar erweitern.
- Für die Animation kann die Timer Klasse verwendet werden
- Setzen Sie im Konstruktor der Form Klasse die entsprechende Properties

```
this.timer1.Enabled = true;
this.timer1.Interval = 50;
this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
```
- Fügen Sie eine private void timer1_Tick(object sender, EventArgs e) Methode hinzu.
- Mittels Refresh() kann der Frame neu gezeichnet werden
- Folgende Anfangswerte sollten den Mars um den Jupiter kreisen lassen.

```
space.Add(new Planet("jupiter", 600, 400, -0.038, 0, 100));
space.Add(new Planet("mars", 600, 600, 3.8, 0, 1));
```
- Folgende Anfangswerte sollten ein chaotisches Verhalten zeigen

```
spaceship = new Spaceship(600, 230, 3, 0, 1);
space.Add(spaceship);
space.Add(new Planet("jupiter", 600, 400, -0.099, 0, 100));
space.Add(new Planet("mars", 300, 400, 0, 1.5, 4));
space.Add(new Planet("merkur", 200, 200, 0, -1.5, 4));
```