

Programmwurf Gruppe C

Wenn eine Rakete einen Satelliten in den Orbit bringt, so muss sie zwei Dinge erreichen:

Sie muss anfangs eine Vertikalbeschleunigung aufbauen, um rasch die Höhe zu erreichen, in der später der Satellit die Erde umkreist. Die Vertikalbeschleunigung wird in potenzielle Energie umgesetzt, d.h. in Hubarbeit. Würde sie nur senkrecht beschleunigen, so würde sie wie bei einem Kanonenschuss eine Gipfelhöhe erreichen und dann zur Erde zurückfallen.

Daher wird nach einer bestimmten Zeit die Rakete langsam in die Horizontale umgelenkt und sie baut nun eine horizontale Geschwindigkeit auf. Ist die nötige Geschwindigkeit erreicht, die benötigt wird, um einen Orbit zu erreichen, so erfolgt der Brennschluss, d.h., die Triebwerke werden abgeschaltet.

Schreiben Sie ein Programm, das diese Berechnung durchführt

Mathematischer Hintergrund:

Die Rakete startet zuerst senkrecht zur Erdoberfläche mit einem Winkel $\alpha=90$ Grad. Sie behält 10-90 s lang diese Ausrichtung, abhängig vom Typ und der Startbeschleunigung.

Danach dreht sie sich langsam in die Horizontale mit einer konstanten Winkelgeschwindigkeit von 0,5 bis 2 Grad/s. Nach 90 s würde sie bei einer Drehrate von 1°/s die Horizontale erreicht haben. Ist der Winkel 0° erreicht, so erfolgt keine weitere Drehung. In dieser Position beschleunigt sie weiter in der Horizontalen.

Die Beschleunigung **a** errechnet sich nach der Formel

$$a = F/m$$

(F: Schub in Newton, m: Masse in Kilogramm)

Sie zerfällt in einen Teil in horizontaler (**ah**) bzw. vertikaler Richtung (**av**), abhängig vom momentanen Winkel α der Rakete

$$a_h = \cos(\alpha) * a$$

$$a_v = \sin(\alpha) * a$$

Die momentane Geschwindigkeit **vh** bzw. **vv** (für horizontale bzw. vertikale Richtung) errechnet sich durch Integration der Beschleunigung in diese Richtung:

$$v_h = \int a_h dt$$

$$v_v = \int a_v dt$$

Genauso wird der Weg S (horizontale und vertikale Komponenten sind **sh** und **sv**) über Integration der Geschwindigkeit berechnet:

$$s_h = \int v_h dt$$

$$s_v = \int v_v dt$$

Da es in Delphi keine Integrationsfunktion gibt, müssen sie für ein kleines Intervall **dt** die Werte addieren. Da die Krümmung der Erdoberfläche in der Rechnung nicht berücksichtigt wird, wird bei dieser Rechenvorschrift irgendwann die Höhe wieder sinken, wenn **vv** negativ ist, was bei realen Aufstiegsbahnen nicht gegeben ist. Sie müssen daher um zu ermitteln, dass die Orbitalhöhe erreicht wird die maximale Höhe **sv** beim Aufstieg ermitteln. Diese muss höher als die Orbitalhöhe sein.

Von den vielen Störeinflüssen auf eine Rakete soll nur die Erdanziehungskraft berücksichtigt werden. Sie entspricht einer Beschleunigung g zur Erdoberfläche.

$$g = \text{grav} * M_{\text{erde}} / d^2$$

grav: Gravitationskonstante = $6,6726 \cdot 10^{-11}$

M_{erde} : Masse der Erde: $5,976 \cdot 10^{24}$ kg

d: Abstand zum Erdmittelpunkt. Die Erdoberfläche hat einen mittleren Abstand von 6371000 m vom Zentrum.

G wirkt nur auf die vertikale Beschleunigung. Sie sinkt um g ab.

Eine Rakete besteht aus mehreren Stufen, jede mit einer Vollmasse (mv), eine Leermasse (ml) und einer Brennzeit (t). Der Antrieb hat einen konstanten Schub F . Sie verbrennt in jeder Sekunde daher einen konstanten Teil des Treibstoffs mt :

$$mt = (mv - ml) / t$$

Aufgrund dessen wird die Rakete bei konstantem Schub immer leichter und die Beschleunigung steigt an.

Bei der heutigen Technologie ist es nicht möglich, mit nur einer Stufe einen Orbit zu erreichen. Bei mehrstufigen Raketen zählt man sowohl die Nutzlast wie auch die oberen Stufen zur Voll- und Leermasse der ersten Stufe hinzu. Nach dem Ausbrennen der ersten Stufe wird diese abgeworfen und es findet dieselbe Berechnung mit der zweiten Stufe statt, die natürlich eine andere Vollmasse, Leermasse, Brennzeit und Schub hat.

Die letzte Stufe führt zusätzlich zur eigenen Masse noch den eigentlichen Satelliten in eine Umlaufbahn.

Die Sollgeschwindigkeit v für eine gegebene Entfernung von dem Erdmittelpunkt d ist berechenbar nach:

$$v = \sqrt{(\text{grav} * M_{\text{erde}} / d)}$$

Für einen Erdorbit muss $v_h = v$ sein.

Schreiben Sie ein Programm, das einen Start einer bis zu dreistufigen Rakete simuliert. Es soll die nötigen Daten der Stufen, sowie die Nutzlast, Zielorbithöhe, vertikale Beschleunigungsphase und Drehrate vom Anwender abfragen.

Das Programm soll die Berechnung stoppen, wenn entweder der Zielgeschwindigkeit v erreicht ist oder der Treibstoff verbraucht ist. Ausgegeben werden soll die momentane Höhe, Geschwindigkeit sowie (wenn noch vorhanden) der übrig gebliebene Treibstoff.

Optionales

Neben diesen Grundanforderungen wären folgende Punkte wünschenswert:

Beliebig viele Stufen: Schreiben sie das Programm so um, dass es mit beliebig vielen Stufen arbeitet.

Ermittelung der besten Aufstiegsbahn: Variieren sie die Dauer der senkrechten Aufstiegsphase und den Drehwinkel so, dass eine maximale Nutzlast (oder die kürzeste Brennzeit, da der Resttreibstoff als zusätzliche Nutzlast angesehen werden kann) resultiert. Beachten sie auch, dass bei Brennschluss mindestens die Orbithöhe erreicht sein muss. Ermitteln sie die maximale Nutzlast und geben sie die Werte für senkrechten Aufstieg und Drehwinkel aus.

Form2

Nutzlast: Senkrechter aufstieg:

Zielorbit: Neigung pro Sekunde:

$v=7790,0 \text{ m/s}^2$, $h=198$ $w=3403$ Treibstoff: 39

	Vollmasse:	Leermasse:	Brennzeit:	Schub:
1:	124400	6900	143,6	2450000
2:	33500	2700	214,6	445000
3:	12200	1690	465	70000

Hier die Daten einer Titan IIIA Trägerrakete.

Anforderungen an das Programm

Neben der primären korrekten Lösung der Aufgabe sollte ihr Programm auch so gestaltet sein, dass sowohl User mit ihm einfach umgehen können, wie auch andere Personen, die ihren Quelltext verwenden müssen, leicht die Funktionsweise nachvollziehen können, bzw. ihn einfach wiederverwenden können.

- Der Code sollte gut strukturiert und übersichtlich sein. Verlagern sie Funktionalitäten in eigene Funktionen und Prozeduren, tauschen sie Daten über Parameter aus.
- Die Funktionen sollten ausreichend kommentiert und dokumentiert sein.
- Das Programm ist gegen Fehleingaben abzusichern (z.B. nicht numerische Werte, Eingaben sind auf Plausibilität zu prüfen.)
- Das Programm sollte den Kriterien für benutzerfreundliche Anwendungen genügen und intuitiv zu bedienen sein.
- Der Quellcode sollte auch ohne Kommentierung leicht lesbar und verständlich sein.
- Umlaute in Bezeichnern sind nicht erlaubt.

Abgabe

Programmwürfe sind wie folgt zu kennzeichnen:

- Erstellen sie einen eigenen Ordner für das Programm.
- Speichern sie alle Projektdateien nur in diesem Ordner. So vermeiden Sie, dass sie später fehlende Teile haben oder Verknüpfungen sich auf absolute Verzeichnispfade beziehen, die eine Compilierung unmöglich machen.
- Benennen sie das Projekt nach dem Schema „**C_Vorname_Nachname**“.
- Benennen Sie niemals Dateien außerhalb Delphi um. Das Programm ist dann nicht mehr compilierbar.
- Packen sie alle zum Programm gehörenden Dateien in ein Zip Archiv. Dazu gehören mindestens:
 - Die Projektdatei mit der Endung .dpr
 - Pro Fenster eine Formulardatei mit der Endung .dfm
 - Pro Fenster eine Quelltextdatei mit der Endung .pas
 - Die Ressourcen Datei mit der Endung .res
- Testen Sie das Programm vor der Abgabe eingehend. Sie haben nur eine Abgabe, die benotet wird.
- Senden sie das Zip Archiv an bl@delphi-vorlesung.de

Sollte der Programmwurf nicht eigenständig erarbeitet sein und mehrere Studenten denselben Programmwurf abgeben, so gilt die Prüfung bei diesen Studenten als nicht bestanden.