

ESCAPE FROM MARS: VERZWEIGUNGEN

Mission 01

PEETIE braucht mehr Energie! Aber in seinem Weg liegen Felsen, die er umfahren muss.



PEETIE muss also 14 Schritte nach links gehen und dabei darauf achten, ob ihm ein Fels im Weg liegt. Falls ja, muss der Fels umfahren werden. Die Lösung für dieses Problem sieht im Struktogramm und im fertigen Programm so aus:

EFM
escape-from-mars.de

```
public void run()
{
    for each (int i in 1..14)
    {
        if (istFelsVoraus())
        {
            aussenrumgehen();
        }
        else
        {
            geheLinks();
        }
    }
}
```

Deklaration: i als Ganzzahl		
Zähle i von 1 bis 14, Schrittweite 1		
J	istFelsVoraus?	N
aussenrumgehen		geheLinks

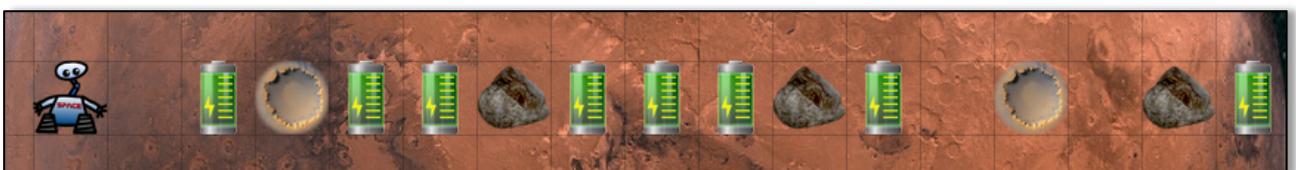
Aufgabe

Teste, ob die obige Lösung funktioniert! Damit programmierst du in **PEETIEs** run()-Funktion, dass **PEETIE** alle Batterien aufsammelt, aber nicht gegen die Felsen stößt.

Verwende dazu die neue if-Verzweigung und die beiden neuen Befehle `istFelsVoraus` und `aussenrumgehen`.

Mission 01b: Nichts als Krater

Jetzt muss **PEETIE** auch noch Kratern ausweichen! Die Situation sieht so aus:



Aufgabe

Schreibe ein Programm, mit dem **PEETIE** alle Batterien aufsammelt. Diesmal muss er jedoch nicht nur die Felsen, sondern auch die Krater umgehen. Verwende die Befehle `istFelsVoraus`, `istKraterVoraus`, `aussenrumgehen`.

Zur Erinnerung:
ODER machen wir mit `||`
UND machen wir mit `&&`



Mission 02: Der Raketensammler

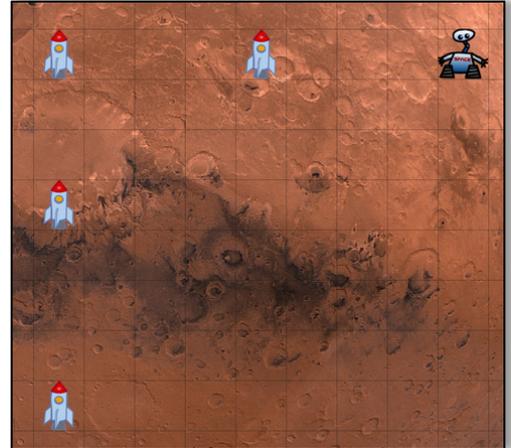
PEETIEs Schicksal hängt davon ab, ob er sich verteidigen kann. Deshalb muss er Raketen einsammeln, die auf der Marsoberfläche verstreut sind.

Aufgabe

PEETIE soll alle Raketen einsammeln. Er muss prüfen, ob er eine Rakete berührt – wenn ja, soll er sie aufnehmen.

Neue Befehle:

- `raketeWirdBeruehrt`
- `raketeAufnehmen`



Mission 03: Minen von SyberDrone

Auf der Marsoberfläche gibt es jetzt Sprengkörper, die genauso aussehen wie Antriebsraketen. Wenn **PEETIE** versucht, so ein Ding aufzunehmen, kann ihn die Explosion zerstören. **PEETIE** muss also vorher herausfinden, ob es sich um eine Rakete oder einen Sprengkörper handelt!



Aufgabe

Entwickle ein Struktogramm für folgende Situation:

PEETIE soll alle Raketen aufnehmen. Wenn er eine Rakete berührt, soll er vor dem Aufnehmen prüfen, ob sie sicher ist. Wenn sie nicht sicher ist, muss er sie entschärfen und dann erst aufnehmen.

Neue Befehle:

- `raketeIstSicher`
 - `entschaerfen`
- (bereits bekannt: `raketeAufnehmen`, `raketeWirdBeruehrt`)

Mission 04: Angriff auf PEETIE

PEETIE wird angegriffen! Wir müssen PEETIE retten, indem wir ihn mit Tasten steuern!



Aufgabe

PEETIE muss sich mit der Tastatur steuern lassen:

Wenn Leertaste gedrückt ist, dann wird eine Rakete abgefeuert.

Mit den Tasten a und d kann PEETIE nach links und rechts ausweichen.

Neue Befehle:

- leertasteIstGedueckt
- raketeAbfeuern
- tasteIstGedueckt("taste")
(Anführungszeichen nicht vergessen!)

Beispiel:

```
if ( tasteGedueckt( "h" ) )  
    geheHoch( )
```