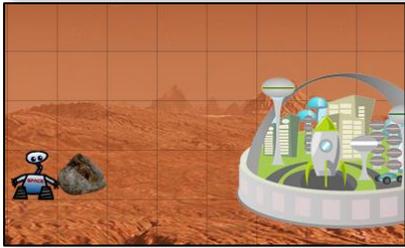


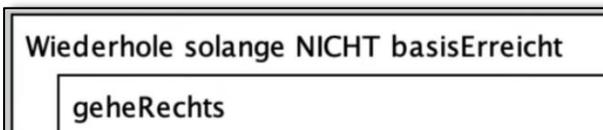
ESCAPE FROM MARS: KOPFGESTEURETE SCHLEIFE (WHILE)



Mission 01: Der lange Weg zur Basis

PEETIE steht kurz vor dem Abflug! Doch er braucht wichtige Ressourcen aus der Basis der Drohnen von *SyberDrone*. Der Weg dorthin ist jedoch lang und steinig ...

Struktogramm:



Codierung:

```
while ( !basisErreicht() )
    geheRechts( )
```

Diese Lösung ist ein Anfang, aber noch nicht perfekt! Probleme kriegt Peetie, wenn ein Fels im Weg ist ☹️

Aufgabe

EFM
escape-from-mars.de

Job für Dr. White
Schreibe für Peetie folgendes Programm:

Peetie geht so lange Richtung Basis, bis er die Basis erreicht hat (kopfgesteuerte Schleife / while-Schleife).
Wenn sich vor Peetie ein Fels befindet, umgeht Peetie ihn (istFelsVoraus ; aussenrumgehen).

Mission 2: Quarzmunitio

Wie ein Fliegenschwarm umkreisen Pandora Daltons Drohnen die Basis - und PEETIE hat keine Raketen mehr. Da kommt Dr. White auf eine geniale Idee: PEETIE kann mit seinem Proben-Ofen aus dem Mars sand Quarzpatronen herstellen!

Aufgabe

Job für Dr. White

Berechne, wie viel Quarzmunitio Peetie herstellen kann. Jede Quarzpatrone verbraucht Ladung von Peetie. Neuer Befehl: `stelleQuarzmunitioHer`

Peetie hat zu Beginn eine Ladung von 3000 GWh. Pro hergestellter Quarzpatrone verbraucht er 130 GWh. Für den Rückflug zur Erde muss seine Ladung aber noch mindestens 1000 GWh betragen!

Verwende eine kopfgesteuerte Schleife, um zu berechnen, wie viele Quarzpatronen Peetie herstellen kann.

Tipp 1: Du brauchst zwei Variablen: eine für die Anfangsladung und eine für die Anzahl hergestellter Quarzpatronen.

Tipp 2: Wenn du es allein nicht schaffst - im Verzeichnis `_material` in deinem Greenfoot-Szenario liegt ein Struktogramm. Das kannst du einfach umsetzen.

Pro-Version: Versuche die Aufgabe so zu verbessern, dass tatsächlich MEHR als 1000 GWh Ladung ÜBRIGBLEIBEN!

Mission 03: Kampf um die Basis

Schwer bewaffnete Kampfdrohnen nähern sich dem armen, kleinen PEETIE. Immerhin kann er jetzt automatisch Quarzmunitio auf die Drohnen schießen. Doch das Grundprogramm muss Dr. White liefern...

Aufgabe

Job für Dr. White

Peetie schießt selbstständig auf die Drohnen, die die Basis bewachen. Das soll er aber nur tun, bis keine Drohnen mehr da sind (sonst beschädigt er die Basis). Verwende dazu eine kopfgesteuerte Schleife (*Wiederhole solange `holeAnzahlFighterdrohnen > 0`*).

Wenn nur noch 4 Drohnen übrig sind, soll Peetie mit dem neuen Befehl `baldFertig` eine Nachricht abschicken.

Peetie soll zwischendurch ausgeben, wie viele Drohnen noch übrig sind (`textAusgeben("blabla")`)

Tipp 1: Du brauchst keine Variablen - mit `holeAnzahlFighterdrohnen` kannst du dir immer die aktuelle Anzahl der angreifenden Drohnen holen.

Tipp 2: Vergiss nicht - wenn du auf Gleichheit prüfst, verwendest du zwei Gleichheitszeichen (`if (holeAnzahlFighterdrohnen() == 4 ...)`)

Mission 04: 30.000 GWh

Gleich hat PEETIE es geschafft! Jetzt muss er nur noch den Schutzschild der Basis wegpusten. Aber er darf nicht zu viel von seiner eigenen Energie verbrauchen, sonst kommt er nie mehr zurück zur Erde...

Aufgabe

Job für Dr. White

Die SyberDrone-Basis hat einen Schutzschild, dessen Leistung wir durch Schüsse mit der Quarzmunition auf 0 bringen wollen. Aber nicht deutlich weniger, sonst gehen alle Ressourcen in der Basis kaputt.

Dragan hat vier Struktogramme geschickt, die aber fehlerhaft zu sein scheinen. Entscheide dich, welches Struktogramm du benutzen willst und implementiere es. Bügle mögliche Fehler aus!!

Neuer Befehl: `quarzmunitionAbfeuern`

E5M04 Basisschild VERSION 1

Deklaration und Initialisierung: `basisSchildleistung` als Ganzzahl = 30000

Deklaration und Initialisierung: `schussSchaden` als Ganzzahl = 4225

wiederhole bis `basisSchildleistung` = 0

`quarzmunitionAbfeuern`

Zuweisung: `basisSchildleistung` = `schussSchaden`

E5M04 Basisschild VERSION 2

Deklaration und Initialisierung: `schussSchaden` als Ganzzahl = 4225

Deklaration und Initialisierung: `basisSchildleistung` als Ganzzahl = 30000

wiederhole solange `basisSchildleistung` <= 0

`quarzmunitionAbfeuern`

Zuweisung: `basisSchildleistung` = `basisSchildleistung` - `schussSchaden`

E5M04 Basisschild VERSION 3

Deklaration und Initialisierung: `schussSchaden` als Ganzzahl = 4225

Deklaration und Initialisierung: `basis` als Ganzzahl = 30000

wiederhole solange `basis` >= 0

`quarzmunitionAbfeuern`

E5M04 Basisschild VERSION 4

Deklaration und Initialisierung: `schussSchaden` als Ganzzahl = 4225

Deklaration und Initialisierung: `basisSchildleistung` als Ganzzahl = 30000

wiederhole solange `basisSchildleistung` >= 0

`quarzmunitionAbfeuern`

Mission 05: Perfekte Programmierung

Die letzten Programme von Dragan waren einfach nur Pfusch! Jetzt scheint er sich aber zusammengerissen zu haben und hat etwas Taugliches geschickt. Obwohl... Man muss beim Programmieren genau darauf achten, dass sich keine Fehler einschleichen...

Aufgabe

Job für Dr. White

Schnapp dir das Struktogramm von Dragan, mit dem Peetie auf den Schutzschild der Basis ballert. Das ist besser als das letzte, hat aber immer noch Fehler.

Implementiere das direkt bei Peetie und korrigiere die Fehler, während du programmierst.

ESM05 Basisschild verbessert (wahrscheinlich auch fehlerhaft)

Deklaration und Initialisierung: schussSchaden als Ganzzahl = 4225	
Deklaration und Initialisierung: basisSchildleistungMaximum als Ganzzahl = 30000	
Deklaration und Initialisierung: basisSchildleistung als Ganzzahl = basisSchildleistungMaximum	
Deklaration und Initialisierung: anzahlSchuesse = 0	
wiederhole solange basisSchildleistung > schussSchaden	
quarzmunitionAbfeuern	
Zuweisung: basisSchildleistung = basisSchildleistung - schussSchaden	
basisSchildleistung < basisSchildleistungMaximum/4	N
J	/
textAusgeben("Schildleistung auf weniger als 50%", 450, 50)	
basisSchildleistung > basisSchildleistungMaximum/2	N
J	/
textAusgeben("Schildleistung auf weniger als 25%", 450, 50)	
textAusgeben("Schildleistung: " + basisSchildleistung, 130, 100)	
textAusgeben("Abgefeuerte Schüsse: ", 140, 150)	