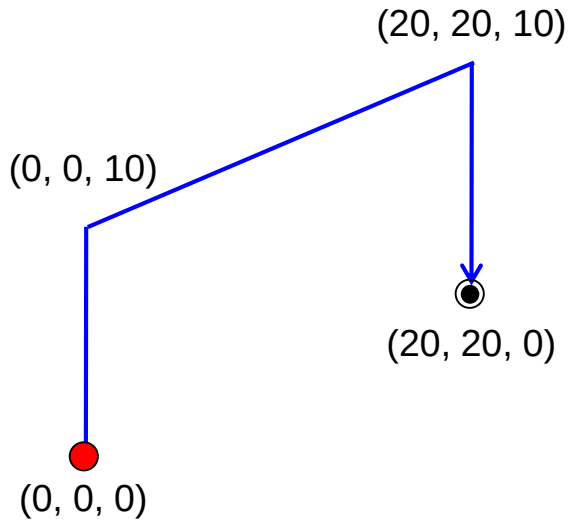
Praktikum (1)

- Im **zweiten Teil des Praktikums** bringen wir das Ufo zum Fliegen.
- Es soll von (0, 0, 0) nach (x, y, 0) fliegen. Die Flughöhe ist z.
  - Die Abbildung zeigt x = 20.0, y = 20.0, z = 10.0.
- Es sollen dabei aussagekräftige Meldungen auf die Konsole ausgegeben werden.
- Testen Sie das fertige Programm mit
  - x = 20.0, y = 20.0, z = 10.0
  - x = -100.0, y = 20.0, z = 10.0
  - x = -1.0, y = -1.0, z = 100.0
  - x = 0.0, y = 40.0, z = 8.0
- Gehen Sie dabei schrittweise vor, wie auf den folgenden Folien gezeigt.



Praktikum (2)

Aufgaben

1. Erstellen Sie eine Datei ufo_autopilot.py.
2. Implementieren Sie in dieser Datei die folgenden Funktionen:

a)

def distance(x1, y1, x2, y2):

Parameter: float x1, y1, x2, y2 : zwei Punkte im kartesischen Koordinatensystem

Rückgabewert: float : Abstand zwischen (x1, y1) und (x2, y2)

Diese Funktion wird benötigt, um rechtzeitig vor dem Erreichen des Ziels abzubremsen.

b)

def angle(x1, y1, x2, y2):

Parameter: float x1, y1, x2, y2 : zwei Punkte im kartesischen Koordinatensystem

Rückgabewert: float : Winkel φ in Grad, $0° \leq \varphi < 360°$
(siehe Praktikum 1, Aufgabe 1)

Mit dieser Funktion wird der Drehwinkel des Ufos bestimmt.

c)

def flight_distance(x1, y1, x2, y2, z):

Parameter: float x1, y1, x2, y2 : zwei Punkte im kartesischen Koordinatensystem
float z : Flughöhe

Rückgabewert: float : zu fliegende Strecke =Summe von drei Teilstrecken

Diese Funktion braucht man, um die tatsächlich geflogene Strecke mit der berechneten Strecke zu vergleichen.

d)

def format_flight_data(sim):

Parameter: sim : Referenz auf die Simulation

Rückgabewert: str : formatierte Zeichenkette mit Flugzeit, x-, y-, z-Koordinate des Ufos

Auf die Daten kann mit sim.getTime(), sim.getX(), sim.getY(), sim.getZ() zugegriffen werden.
Beispiel eines Rückgabewerts: `"8.5 s: [  10.2  -3.1  10.0]"`

Für die Teilaufgaben a) bis c) haben wir bereits Skripte erstellt. Sie müssen sie nur als Funktionen schreiben. Für d) können Sie die Lösung der letzten Aufgabe im Praktikum 1 verwenden.

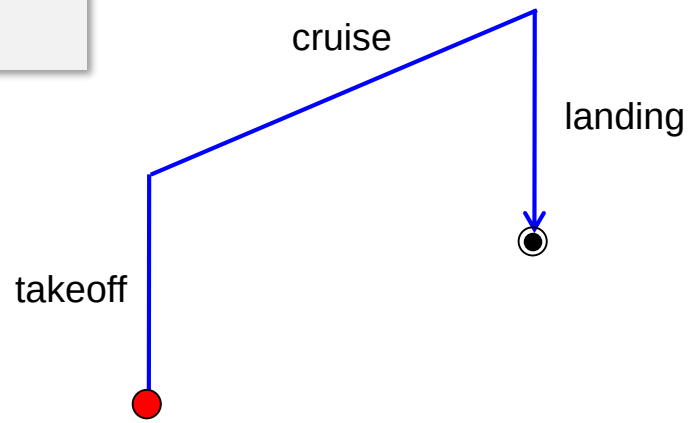
e)

def fly_to(sim, x, y, z):

Parameter: sim : Referenz auf die Simulation
float x, y : Punkt im kartesischen Koordinatensystem
float z : Flughöhe

Rückgabewert: Keiner

Diese Funktion fliegt das Ufo ins Ziel und ruft lediglich die drei folgenden takeoff, cruise und landing Funktionen auf. Alle drei Funktionen haben keinen Rückgabewert.



Praktikum (5)

Aufgaben

def takeoff(sim, z):

Parameter: sim : Referenz auf die Simulation
float z : Flughöhe

Rückgabewert: Keiner

def cruise(sim, x, y):

Parameter: sim : Referenz auf die Simulation
float x, y : Punkt im kartesischen Koordinatensystem

Rückgabewert: Keiner

def landing(sim):

Parameter: sim : Referenz auf die Simulation

Rückgabewert: Keiner

Die Funktionen sind fast komplett vorgegeben und können in die Datei ufo_autopilot.py hineinkopiert werden.

Sie sollen die Funktionen lediglich um laufende Meldungen des Flugfortschritts auf die Konsole ergänzen. Verwenden Sie dazu die Funktion format_flight_data.

f)

```
def fac(n):
Parameter:      int n mit n > 0
Rückgabewert:   int : Fakultät von n
```

Berechnen Sie die Fakultät mit der Rekursionsformel fac(1) = 1, fac(k) = k * fac(k – 1). Diese Funktion wird erst in einer späteren Praktikumsaufgabe benötigt.

3. Das Hauptprogramm ufo_main.py ist vorgegeben. Sie brauchen es nur von Moodle herunterladen. Ergänzen Sie das Hauptprogramm an den gekennzeichneten Stellen

   i. um die Konsoleingabe des Ziels x, y und der Flughöhe z

   ii. um eine Konsolausgabe der zu fliegenden Distanz

   ```
   flight_distance(0, 0, x, y, z)
   ```

   iii. um eine Konsolausgabe der tatsächlich geflogenen Distanz

   ```
   sim.getDist()
   ```

   Wenn Sie nach (–100.0, 20.0) fliegen, werden Sie feststellen, dass das Ufo ein Stück neben dem Zielpunkt landet. Das ist nicht gut. Falls das Ufo z.B. auf einer Straße landet, wird es überfahren und die zu liefernde Pizza schmeckt nicht mehr.

   Warum ist das so? Wie kann man das ändern?

Praktikum (8)

- Abgabe:
  - Testen Sie das fertige Programm ausgiebig. Wenn alle Tests erfolgreich waren, verpacken Sie die beiden py-Dateien ufo_autopilot.py und ufo_main.py in eine zip-Datei. Laden Sie die zip-Datei anschließend in Moodle hoch.
  - Bitte laden Sie die zip-Datei rechtzeitig hoch.

Praktikum (9)

- Checkliste: Überprüfen Sie vor dem Hochladen der Abgabe, ob Folgendes erfüllt ist:
  - Die Abgabe ist eine zip-Datei, die zwei py-Dateien enthält.
  - Die py-Datei ufo_autopilot.py enthält die Funktionen distance, angle, flight_distance, format_flight_data, fly_to, takeoff, cruise, landing, fac. Außer diesen Funktionen und benötigten Import-Anweisungen ist kein weiterer Code enthalten.
  - Die py-Datei ufo_main.py enthält das Hauptprogramm. Außer dem Hauptprogramm und benötigten Import-Anweisungen ist kein weiterer Code enthalten.
  - Das Programm hat keine Syntaxfehler.
  - Das Programm ist mit verschiedenen Eingabewerten getestet und fehlerfrei.