

L1_1.1 Die Programmumgebung PythonKara

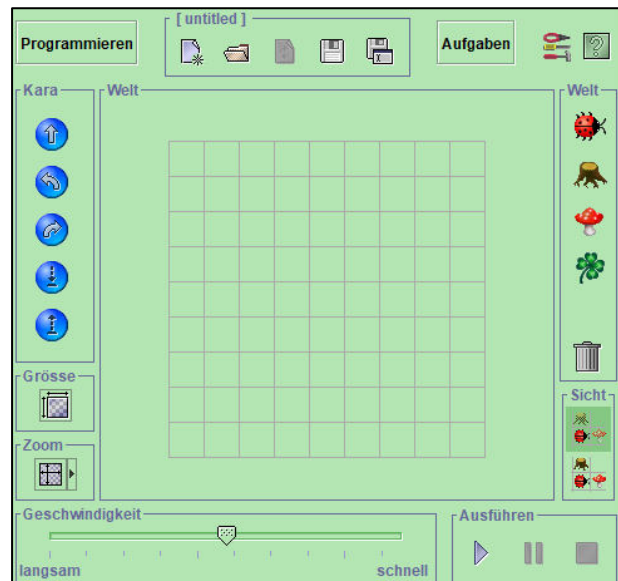
Kara ist eine interaktive Entwicklungsumgebung zur spielerischen Einführung in die Programmierung, die Anfang der 2000er Jahre an der Technischen Hochschule Zürich entwickelt wurde.

Im Laufe der Zeit wurden mehrere Versionen von Kara für die Programmierung mit unterschiedlichen Programmiersprachen zur Verfügung gestellt – u.a. auch für Python.

Sobald PythonKara gestartet wurde, erscheint die abgebildete Bildschirmoberfläche.

Die Vorlage enthält eine 9 x 9 Felder große Spielfläche – die Welt – und stellt verschiedene Akteure zur Verfügung.

In der Programmierungsumgebung PythonKara interagiert der Käfer *Kara* mit Akteuren (*Baum*, *Pilz* und *Blatt*) in einer Welt aus Quadraten.



Erläuterungen zum Weltfenster von PythonKara

Welt erstellen, speichern und öffnen:

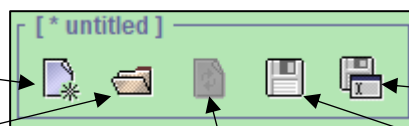
Eine neue Welt erstellen

Eine Welt öffnen

Eine Welt erneut öffnen

Eine Welt speichern unter

Eine Welt speichern



Akteure (Elemente) der Welt:



Der Käfer (kara)

Der Baum (tree)

Der Pilz (mushroom)

Das Blatt (tree)

Abfalleimer

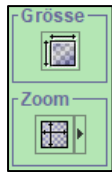
Elemente in die Welt einfügen:

Element anklicken und auf einem Rechteck der Welt platzieren (Drag & Drop)

Elemente aus der Welt entfernen:

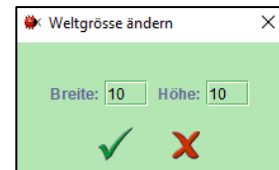
Drag auf dem zu löschenden Element starten (Klick) und über dem Abfalleimer loslassen.

Größe und Ansicht der Welt:



Festlegen der Weltgröße. Die Standardgröße von 9 x 9 Feldern kann bis zu 1000 x 1000 Felder vergrößert werden.

Die Ansicht der Welt (Fenstergröße) kann verkleinert bzw. vergrößert werden.



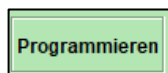
Direkte Steuerung des Käfers :



(zugehörige Python-Befehle)

- | | |
|---------------------------------------|-----------------------------------|
| Der Käfer geht einen Schritt vorwärts | → <code>kara.move();</code> |
| Der Käfer dreht sich nach links | → <code>kara.turnLeft();</code> |
| Der Käfer dreht sich nach rechts | → <code>kara.turnRight();</code> |
| Der Käfer legt ein Blatt ab | → <code>kara.putLeaf();</code> |
| Der Käfer hebt ein Blatt auf | → <code>kara.removeLeaf();</code> |

Programmierung des Käfers



Die Schaltfläche startet den Programmeditor. Hier werden die einzelnen Aktionen festgelegt, die während des Programmablaufs „abgearbeitet“ werden sollen. Man spricht hierbei von der Kodierung des Programms

Das Editor-Fenster zeigt die Befehle (Aktionen) und Sensoren, die zur Steuerung des Käfers zur Verfügung stehen (Zeilen 1 – 8).

Der Hashtag zu Beginn der Zeilen 1 bis 12 bedeutet, dass es sich in diesen Zeilen um keine Python-Anweisungen handelt, sondern um Kommentare.

Einzeiliger Kommentar → #

Mehrzeiliger Kommentar → """
(Drei Anführungszeichen zu Beginn eines mehrzeiligen Kommentars und drei Anführungszeichen danach.)

```
[untitled]
1 # BEFEHLE: kara.
2 # move() turnRight() turnLeft()
3 # putLeaf() removeLeaf()
4 #
5 # SENSOREN: kara.
6 # treeFront() treeLeft() treeRight()
7 # mushroomFront() onLeaf()
8 #
9
10 # hier können Sie eigene Methoden definieren
11
12 # hier kommt das Hauptprogramm hin, zB:
13 while not kara.treeFront():
14     kara.move()
15
```

Außerdem enthält das Editor-Fenster bereits einen Programmcode, der den Käfer so lange einen Schritt gehen lässt, bis er vor einem Baum steht.

Bei der Entwicklung und Implementierung eigener Programme können die Zeilen 10 – 12 (Kommentare) gelöscht werden. Die Zeilen 13 und 14 (Programmcode) müssen gelöscht werden.

Sensoren zur Steuerung des Käfers



`kara.treeFront()`

Kara prüft, ob er vor einem Baum steht und liefert einen Wert vom Typ boolean zurück (true / false).



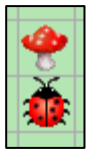
`kara.treeLeft()`

Kara prüft, ob links von ihm ein Baum steht und liefert einen Wert vom Typ boolean zurück (true / false).



`kara.treeRight()`

Kara prüft, ob rechts von ihm ein Baum steht und liefert einen Wert vom Typ boolean zurück (true / false).



`kara.mushroomFront()`

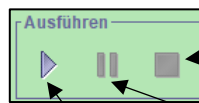
Kara prüft, ob er vor einem Pils steht und liefert einen Wert vom Typ boolean zurück (true / false).



`kara.onLeaf()`

Kara prüft, ob er auf einem Blatt steht und liefert einen Wert vom Typ boolean zurück (true / false).

Steuerung der Programmausführung



Startet das Programm

Hält den Programmablauf an

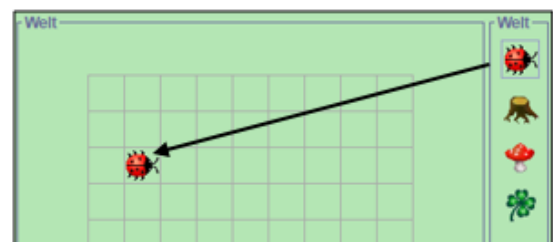
Stoppt das Programm



Regelt die Geschwindigkeit des Programmablaufs

Akteure verwenden und Aktionen zuweisen

Um Akteure auf der Welt zu platzieren, wird das gewünschte Symbol in der Leiste der Element-Leiste angeklickt und per Drag & Drop auf ein beliebiges Feld der Welt gezogen.

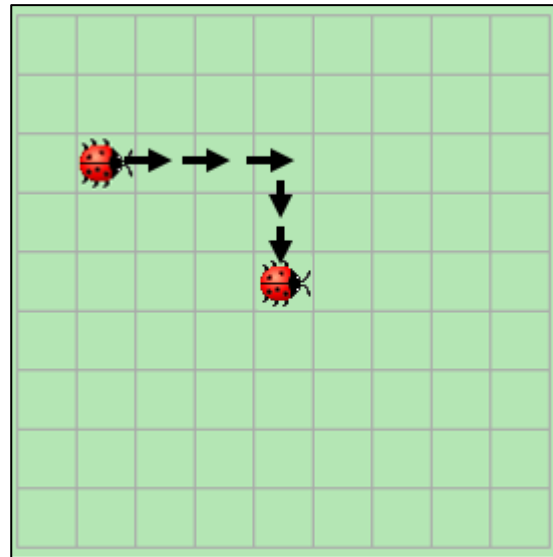


Elemente können auch wieder von der Welt entfernt werden. Dazu wird das gewünschte Element in der Welt angeklickt und per Drag & Drop in den Abfalleimer gezogen.

J1	BPE 5: Grundlagen der Programmierung Informationsmaterial	Informatik
----	---	------------

Um den Käfer vom Ausgangspunkt in der dritten Zeile und zweiten Spalte zum Endpunkt in der fünften Zeile und fünften Spalte mit Blick zum rechten Rand der Welt zu bringen, muss er folgende Aktionen ausführen:

- Kara, gehe einen Schritt vorwärts
- Kara, gehe einen Schritt vorwärts
- Kara, gehe einen Schritt vorwärts
- Kara, drehe nach rechts
- Kara, gehe einen Schritt vorwärts
- Kara, gehe einen Schritt vorwärts
- Kara, drehe nach links



Um diese Anweisungen durch ein Programm ablaufen zu lassen ist folgender Python-Programmcode notwendig:

Nachdem das Programm gespeichert wurde (siehe unten), kann es mit der Schaltfläche 'Programm laufen lassen'

ausgeführt werden.



Falls das Python-Programm Fehler enthalten sollte, kann mit einem Klick auf die Fehlermeldung unterhalb des Programmeditors die Stelle ermittelt werden, an welcher der Fehler aufgetreten ist.

```

JavaScriptKara programmieren
[* test.js]
1  /* BEFEHLE: kara.
2  *   move() turnRight() turnLeft()
3  *   putLeaf() removeLeaf()
4  *
5  * SENSOREN: kara.
6  *   treeFront() treeLeft() treeRight()
7  *   mushroomFront() onLeaf()
8  */
9  kara.move()
10 kara.move()
11 kara.move()
12 kara.turnRight()
13 kara.move()
14 kara.move()
15 kara.turnLeft()

```

```

10 kara.move()
11 kara.move()
12 kara.move()
13 kara.turnRight()
14 kara.move()
15 kara.move()
16 kara.moveLeft()

```

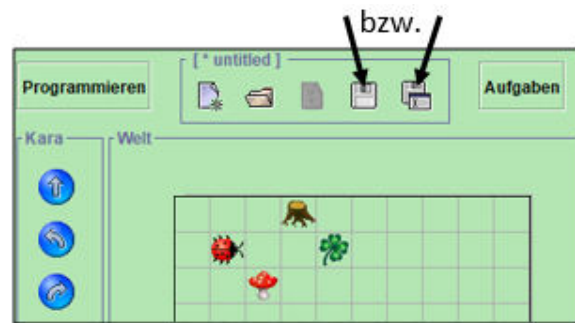
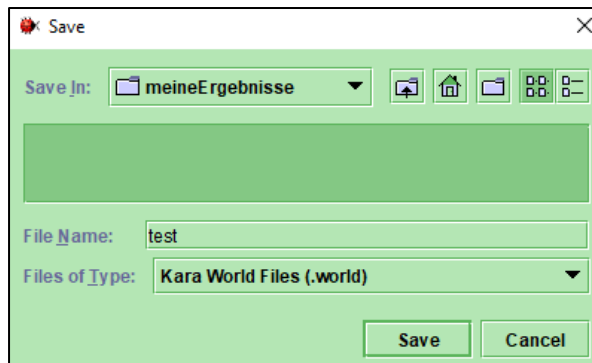
AttributeError: 'javainstance' object has no attribute 'moveLeft', in line 16

Speichern der Welten und Programme

Beim Speichern eines mit PythonKara entwickelten Projekts muss unterschieden werden, ob nur der eingefügte Programmcode oder auch die in der Kara-Welt platzierten Akteure gespeichert werden sollen.

Speichern der Kara-Welt

Eine Welt, in der Akteure (Käfer, Bäume etc.) platziert sind, kann gespeichert werden. Sie steht bei erneutem Öffnen in entsprechender Form wieder zur Verfügung.

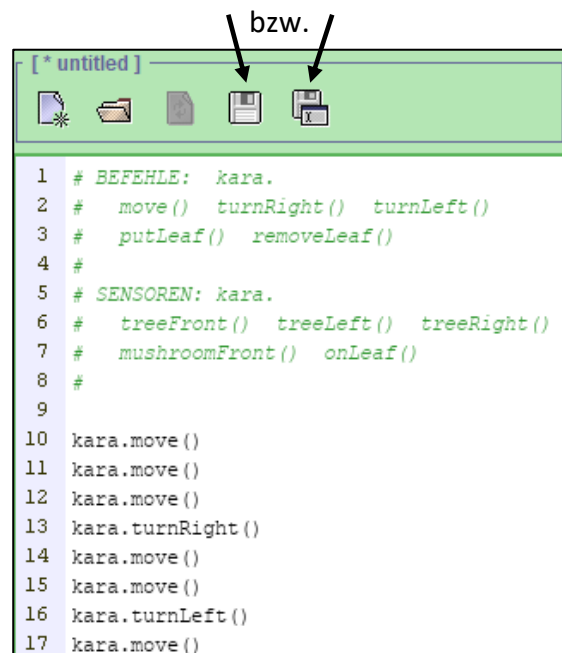
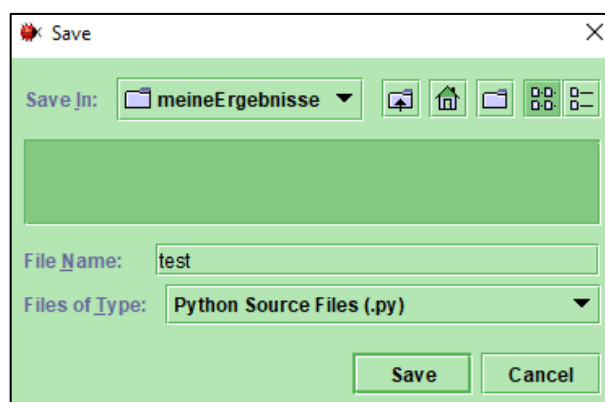


Die Datei erhält den Dateinamenszusatz 'world'.

In Ihrem Ergebnisordner befindet sich anschließend die Datei *test.world*.

Speichern des Programmcodes

Der Programmcode zu einem Python-Projekt wird in einer separaten Datei gespeichert.



Die Datei erhält den Dateinamenszusatz 'py'. In Ihrem Ergebnisordner befindet sich danach die Datei *test.py*.

Um ein vorhandenes Python-Projekt zu bearbeiten, müssen somit immer zwei Dateien geöffnet werden, die World-Datei und die Python-Programmdatei.