

Werkzeuge für den Informatikunterricht

Kara

1. Kurzvorstellung

Kara ist ein Lernwerkzeug, mit dem man spielerisch und anschaulich die Grundlagen des Programmierens lernen kann. Kara ist auch der Name des programmierbaren Marienkäfers, den man durch eine 2-dimensionale Welt bewegen soll. Diese Welt besteht aus Bäumen, die für den Käfer eine Wand darstellen, Kleeblätter, die er aufheben und ablegen kann und Pilze, die er umherschoben kann. Im Weltenfenster wird die Welt gebaut und am Ende das Programm ausgeführt. Um Kara zu programmieren, muss man im Programmfenster einen Automaten erstellen. Hierfür muss man Zustände (Kreise) und Zustandsübergänge (Pfeile) festlegen. Das Programm startet in dem mit „Start“ markierten Zustand und springt über die Übergänge zwischen den Zuständen hin und her. Ein Zustandsübergang legt fest, wann Kara was machen muss. Er entsteht, wenn man in einem Zustand definiert, welche Aktionsfolge Kara ausführen muss, je nachdem was die Sensoren messen. Für jede Situation muss ein Folgezustand angegeben werden. Wenn dieser der Stop-Zustand ist, endet das Programm.

2. Einordnung in die Lehrpläne

Kara ist am Gymnasium und an der Oberschule mit Lehrplaninhalten als Werkzeug einsetzbar. Zielgruppe sollten SuS mit keiner oder wenig Programmiererfahrung sein.

Gymnasium:

Im Lernbereich 4 „Algorithmen und Programme“ der Klassenstufe 9 lässt sich Kara zur Vermittlung eines Algorithmenbegriffs und die graphische Beschreibung von Algorithmen und Programmen nutzen.

Im Lernbereich 2 „Informatische Modelle“ im Grundkurs 11/12 lässt sich Kara zur Vermittlung des Modellbegriffs und zum Arbeiten in einem informatischen Modell nutzen.

Darüber hinaus kann man Kara auch im Lernbereich 8A „Theoretische Informatik – Theoretische Grundlagen von Programmiersprachen“ im Grundkurs 11/12 einbauen, um das Automatenmodell Endlicher Automaten einzuführen.

Oberschule:

Im Lernbereich 2 „Informationen verarbeiten: Modell – Algorithmus – Lösung“ in der Klassenstufe 8 lässt sich Kara zur Vermittlung des Algorithmus-, Problemlöse- und Modellbegriffs nutzen. Es kann zum selbstständigen Lösen einfacher Probleme anhand einfacher Automaten verwendet werden

3. Lernziele

Kognitive Lernziele:

Die SuS lernen die den Begriff des Algorithmus kennen, indem sie die Eigenschaften eines Algorithmus anhand von Problemen für den Marienkäfer erläutern.

Die SuS kennen das Modell endlicher Automaten und zeigen dies indem sie aufzählen, welche Bestandteile einen Automaten definieren.

Die SuS kennen die Struktur eines Problemlöseprozesses und zeigen dies, indem sie planvoll einfache Probleme modellieren unter Verwendung von Heuristiken.

Die SuS verstehen das Prinzip der vollständigen Fallunterscheidung, indem sie alle möglichen Sensorbelegungen durchgehen.

Die SuS grenzen Probleminstanz und Algorithmus durch die Trennung von Welt- und Programmfenster voneinander ab.

Psychomotorische Lernziele:

Die SuS visualisieren Algorithmen als Zustandsdiagramm.

Die SuS setzen ein Modell anhand eines einfachen Beispiels in Kara um.

Affektive Lernziele:

Die SuS erkennen das Problemlösen nicht nur in Verbindung mit Programmcode bedeutsam ist.

4. Kompetenzentwicklung

Fachkompetenz:

Die SuS können den Begriff des Algorithmus erläutern.

Die SuS beherrschen die Wiedergabe des Modellbegriffs

Die SuS können Algorithmen in Kara entwerfen und realisieren.

Die SuS können Algorithmen als Zustandsdiagramm darstellen.

Die SuS modellieren Automaten anhand einer einfachen Problemlösung.

Lern-/Methodenkompetenz:

Die SuS sind in der Lage mithilfe von Heuristiken Probleme zu lösen.

Die SuS beherrschen das planvolle Vorgehen bei einer Modellierung.

Sozialkompetenz:

Die SuS können verschiedene Lösungen für das gleiche Problem wertschätzen

Selbstkompetenz:

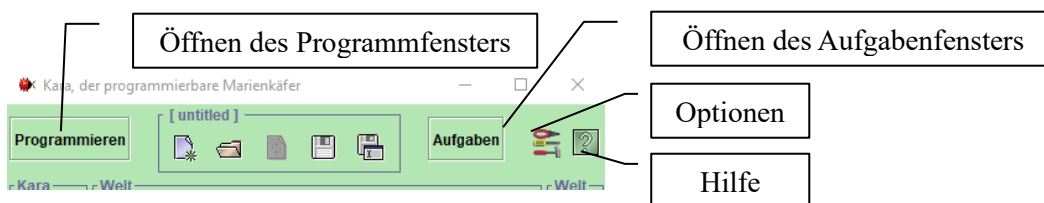
Die SuS können verschiedene Lösungsansätze diskutieren.

5. Prinzipieller Aufbau

Screens, Menüleisten, Optionen etc.

ggf. durch beschriftete Screenshots untersetzt

Kara ist grundlegend in zwei Fenster gegliedert: dem Weltfenster und dem Programmfenster. Das Weltfenster gilt der Erstellung der Probleminstanz und das Programmfenster der des Algorithmus. Beim Start von Kara öffnet sich das Weltfenster. Von dort aus kann man das Programm-, das Aufgabenfenster, die Optionen und die Hilfe öffnen.

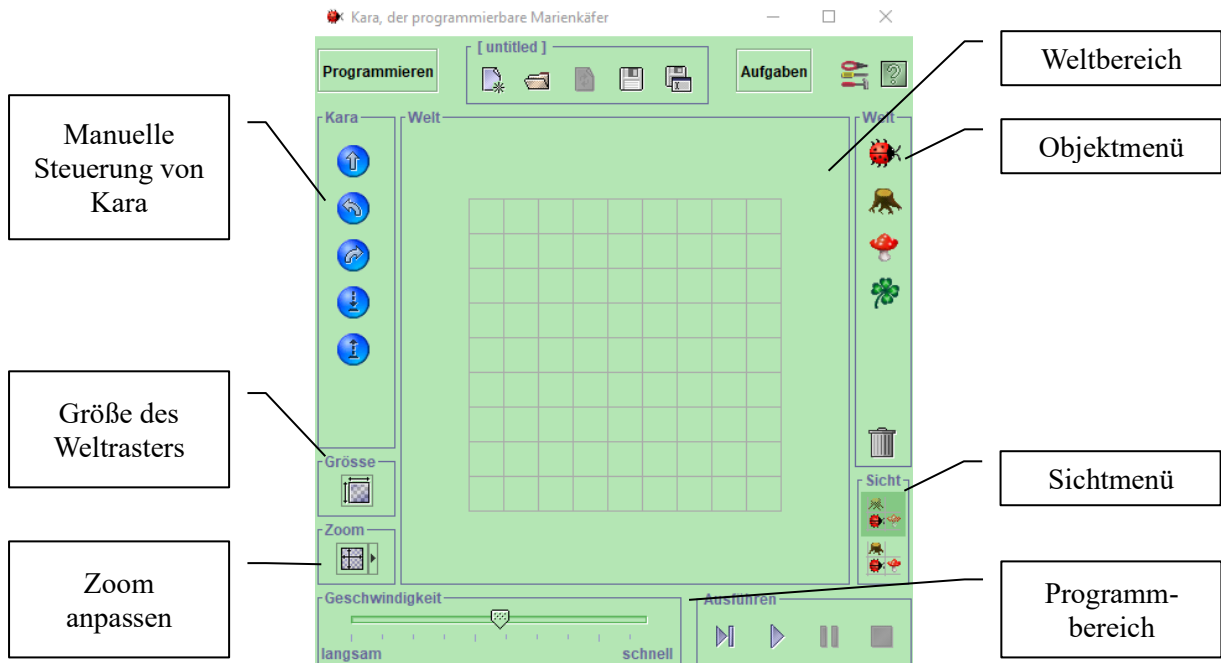


In beiden Fenstern (dem Weltfenster und dem Programmfenster) gibt es nebenstehendes Panel, welches zum Speichern, Öffnen und Erstellen von Dateien verantwortlich ist. Programm- und Weltdatei werden separat gespeichert.



In der Hilfe befindet sich eine Bedienungsanleitung zu Kara. Hier sind alle Befehle, Weltobjekte, Sensoren und Fenster erklärt.

Im Aufgabenfenster gibt es eine große Auswahl an vorgefertigten Aufgaben, geordnet nach Schwierigkeitsgrad. Dort ist die Aufgabe inklusive Testwelten eingebunden. Es gibt sowohl Kara-Versionen mit Lösungen als auch solche ohne.

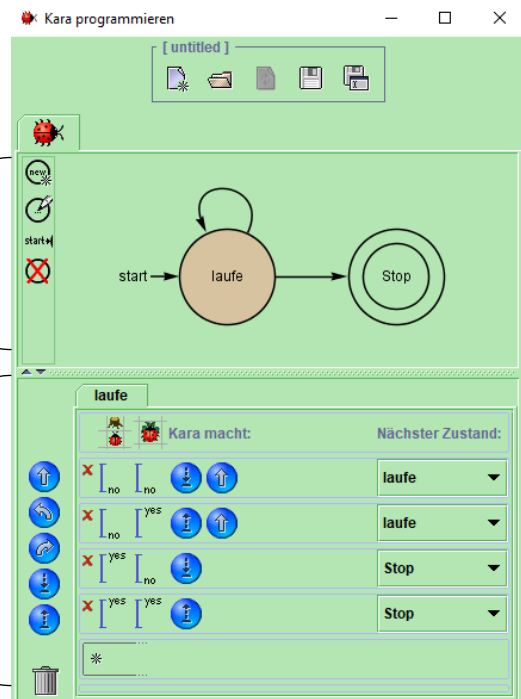


Im Programmfenster programmiert man den Marienkäfer Kara. Es ist in zwei Teile gegliedert: Der Automatenansicht und der Zustandsansicht. In der Automatenansicht sieht man das Zustandsdiagramm seines Automaten. Hier kann man Zustände erstellen, bearbeiten und löschen.

Automatenansicht

Wählt man in der Automatenansicht durch Klicken einen Zustand aus, öffnet sich dieser in der Zustandsansicht. Hier wird dargestellt, was Kara macht abhängig davon, was ihre Sensoren messen. Auf der rechten Seite sieht man die Folgezustände.

Zustandsansicht



6. Handhabung

Weltbereich und Objektmenü:

- Objekte per Drag&Drop reinziehen
- Zum Entfernen auf Papierkorb ziehen
- Auch als JPG exportierbar



Unbewegliche Baustümpfe



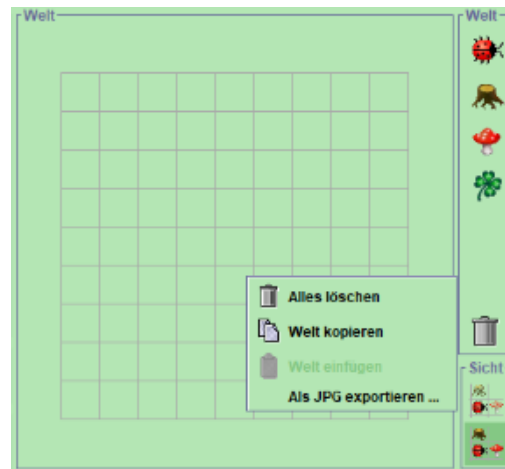
Pilz, die Kara vor sich herschieben kann maximal 1 gleichzeitig



Kleeblätter, die Kara aufheben und ablegen kann unbeschränkt



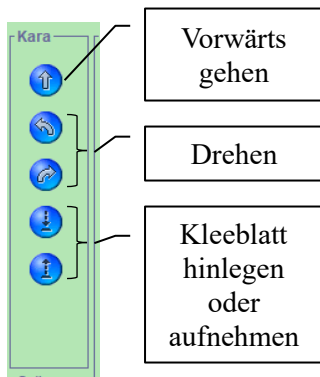
Kara



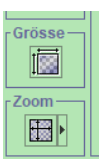
Sichtmenü:

alle Objekte außerhalb Karas Sichtweite halbtransparent darstellen (oder nicht)

Manuelle Steuerung von Kara: (zum Probieren)



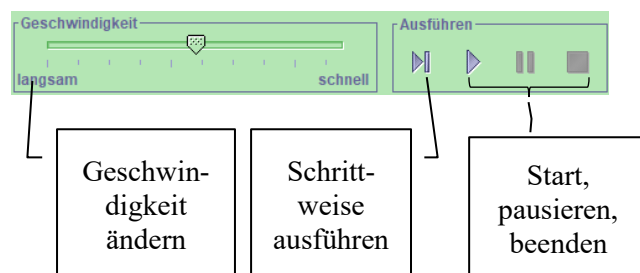
Größe und Zoom:



- Größe des Rastes
- Zoom zurücksetzen
- Zoom auch mit ALT und gedrückter linke Maustaste über Weltbereich ziehen
- Verschieben des sichtbaren Weltausschnitts mit gedrückter linker Maustaste über Weltbereich

Programmbereich:

- auch schrittweise Programmausführung für Debugging
- Aktueller Zustandsübergang wird im Programmfenster gelb hervorgehoben




Automatenansicht:

Neuer Zustand: 

- Zustand benennen
- Benutze Sensoren aus der Sensoren-Bibliothek nach links ziehen

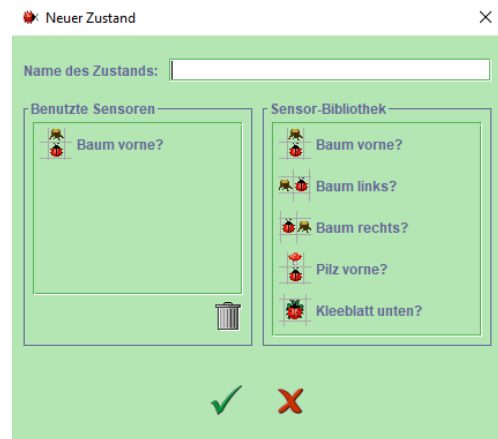
Ausgewählten Zustand bearbeiten: 

Ausgewählten Zustand löschen: 

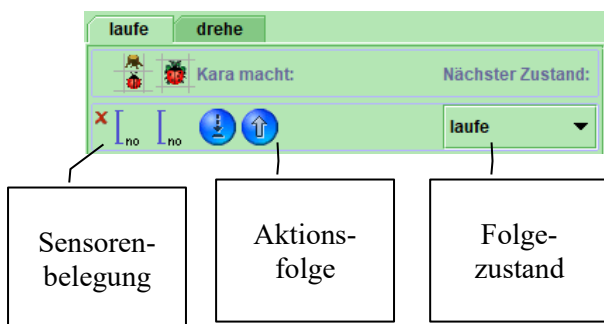
Als Startzustand definieren: 




Kontextmenü in der Automatenansicht:

- Als JPG exportieren
- Automatisches Layout oder Kreislayout
- Automatengröße an Fenster anpassen
- Zustand einfügen (Rechtsklick auf Zustand zum Kopieren)



Zustandsansicht:



- Zustände als Reiter
- Tabellarische Aufzählung der Zustandsübergänge
- Mehrfaches Klicken auf  um Sensorbelegung zu definieren
- Neuer Zustandsübergang 
- Zustandsübergang löschen 

Wenn die Sensorenmesswerte von Kara der Sensorenbelegung in den linken Spalten der Tabelle entsprechen, führt der Käfer die Aktionsfolge von links nach rechts aus und geht in den Folgezustand über

7. Vor- und Nachteile

Pro:

- spielerischer Umgang mit Automaten
- Veranschaulichung in Zustandsdiagramm
- nicht codebasiert
- entdeckendes Lernen
- leichter Debugging-zugang
- viele Beispielaufgaben
- Parallelen zu Java-Kara, Phyton-Kara, Turing-Kara etc.

Contra:

- Komplex und viele Fenster
- Zustandsdiagramm sollte bekannt sein
- Java-Umgebung (in neuen Versionen nicht mehr kostenfrei für Schulen)

8. Beispielaufgaben

Aufgabe 1: Kleeblattsuche

Erstelle eine neue Welt. Ändere die Weltgröße auf eine Breite von 10 und eine Höhe von 1. Platziere Kara am linken Rand der Welt und ein Kleeblatt irgendwo rechts von ihr. Programmiere Kara so, dass sie solange geradeaus läuft, bis sie ein Kleeblatt findet und es dann aufhebt. Es wird nur ein Zustand neben dem Stop-Zustand benötigt. Teste, ob dein Programm auch funktioniert, wenn das Kleeblatt woanders liegt.

Aufgabe 2: Blättersammler

Lösche alle Objekte aus der Welt von Aufgabe 1. Kara soll am linken Rand der Welt stehen. Platziere am rechten Rand einen Baumstumpf. Platziere auf dem Weg von Kara zum Baumstumpf drei Kleeblätter und zusätzlich eins unter Kara und eins vor dem Baumstamm. Programmiere Kara so, dass sie zum Baumstumpf läuft und dabei alle Kleeblätter aufhebt.

Zusatz: Modifiziere das Programm so, dass Kara auf den Feldern, wo kein Kleeblatt nun ein Kleeblatt ablegt, aber nur wenn sie noch genug im Gepäck hatte, die sie vorher aufgehoben hat. (Tipp: Nutze neue Zustände, um zu zählen, wie viele Kleeblätter Kara im Gepäck hat)

Aufgabe 3: Slalom

Erstelle eine neue 10x3 Felder große Welt. Platziere Kara am linken Rand in der Mitte und am rechten Rand in der Mitte ein Kleeblatt. Zwischen Kara und dem Ziel-Kleeblatt sollen beliebig viele Baumstämme stehen. Schreibe ein Programm, sodass Kara im Slalom um die Baumstämme läuft bis sie auf dem Ziel-Kleeblatt steht. Teste, ob dein Programm auch funktioniert, wenn mehrere Baumstämme direkt nebeneinanderstehen.

Zusatz: Erweitere die Welt auf 10x5 Felder. Programmiere Kara so, dass sie einen Pilz im Slalom schiebt.

Aufgabe 4: An der Wand entlang

Erstelle eine neue 10x10 Felder große Welt. Platziere Baumstümpfe als Wände eines Raumes auf allen Randfeldern. Entferne einen Baumstumpf in der Mitte vom linken Rand. Das ist die Tür Kara soll in der Tür starten. Kara soll einmal alle Wände des Raumes im Uhrzeigersinn ablaufen. Platziere auf ihrem Weg beliebig viele Kleeblätter. Programmiere Kara so, dass sie an der Wand entlang alle Kleeblätter aufsammelt. (Tipp: Kara muss erst eintreten. Wie erkennt Kara eine Ecke? Wie erkennt Kara, dass sie fertig ist?)

9. Screencast

Link zum Lehrvideo:

<http://bit.ly/karavideo>

Passwort: Kara

