

Werkzeuge für den Informatikunterricht

---- Kara – Programmieren mit endlichen Automaten ----

1. Kurzvorstellung

Bei Kara geht es um einen Marienkäfer namens Kara, der in einer einfachen Welt lebt. Er kann programmiert werden, um so diverse Aufgaben zu erledigen, zum Beispiel Kleeblätter sammeln oder Pilze schieben. Karas Programme sind endliche Automaten und werden in einer grafischen Entwicklungsumgebung erstellt. Die Grundideen der Programmierung werden in Kara vermittelt. Positiv anzumerken sind 2 Eigenschaften: Endliche Automaten sind einfach zu verstehen und die Einarbeitungszeit in das Programm ist der klein. Außerdem ist die Umgebung sehr einfach gehalten, ohne mit komplexe Entwicklungsumgebungen sich auseinander setzen zu müssen. So lernen Schüler und Schülerinnen auf spielerische Art und Weise das Programm Kara kennen. Kara kann auf mehreren Schulstufen eingesetzt werden.

2. Einordnung in die Lehrpläne

Das Tool ist für die BS und die BGy nicht zu empfehlen, kann man aber bestimmt auch mal eingebaut werden.

Dafür kann man es auf dem Gymnasium teilweise anwenden, insbesondere in der Klasse 11 im Lernbereich 2, informatische Modelle.

Sogar in Klasse 12 könnte man Kara anwenden, beispielsweise im Lernbereich 8 A – Theoretische Informatik – theoretische Grundlagen von Programmiersprachen. ABER nur, als spielerischen Einstieg in das Thema. (Den SuS soll die Grundidee von Algorithmen und dem programmieren wieder ins Gedächtnis gerufen werden).

An sich ist das Tool Kara für das Gymnasium nicht sonderlich geeignet.

Ich würde das Tool auf jeden Fall in der Oberschule zum Einsatz bringen. In der Klasse 8 im Lernbereich 2 – Informationen verarbeiten: Modell – Algorithmus – Lösung.

Kennen grundlegender Programmstrukturen	Auswahl schülergerechter Software
- Begriff: Algorithmus (Endlichkeit, Eindeutigkeit, Ausführbarkeit, Allgemeingültigkeit)	Endlichkeit der Beschreibung und der Ausführung beachten Zusammenarbeit mit MA
- Programmstrukturen: Folge, Wiederholung, Verzweigung	Ausgangspunkt: umgangssprachliche Handlungsbeschreibung
- eine Darstellungsform	Struktogramm, Ablaufplan, Zustandsdiagramm
Beherrschen der Umsetzung des Modells an einfachen Beispielen	Teilschritte aus Problemen herauslösen: Zeichnen einer Figur, Vertauschen von zwei Elementen, Ansteuern eines Kanals
Kennen des Problemlöseprozesses	⇒ Methodenkompetenz
- Problemanalyse, Lösungsentwurf, Umsetzung, Test, Dokumentation	Entwurf: auch Modell/Algorithmus Umsetzung: auch Implementierung Test: auch Kritik und Korrektur
- Lösen eines einfachen Problems unter Nutzung der Programmstrukturen	Sortieren, einfache Steuerung, einfache Makros, Erzeugen von Grafiken
Übertragen der Kenntnisse zum Problemlöseprozess auf	Partnerarbeit
- selbstständiges Lösen einfacher Probleme	Ampelsteuerung, einfache Automaten, Aufgaben in einfachen grafischen Programmierumgebungen
- kritische Bewertung der Resultate	⇒ Werteorientierung: Handlungs- und Urteilsfähigkeit

3. Lernziele

Die Schüler-innen lernen die Grundlagen der Programmierung kennen.

Die Schüler-innen lernen mithilfe von endlichen Automaten eine die fundamentale Idee von Algorithmen kennen.

Die Schüler-innen lernen ein Berechnungsmodell der Informatik (endliche Automat) kennen.

Die Schüler-innen beherrschen den Umgang mit der Entwicklungsumgebung Kara.

Die Schüler-innen wenden ihre erworbenen Kenntnisse über das Programm an.

Die Schüler-innen untersuchen Aufgaben und versuchen sie zu lösen.

Die Schüler-innen können Probleme/Aufgaben mittels des Programmes Kara lösen.

Kognitive Lernziele:

Die SuS kennen die Grundlagen der Informatik.

Die SuS finden zu einer gegebenen Problemstellung den geeigneten Lösungs-Algorithmus.

Die SuS wenden ihre Kenntnisse zur Erstellung der Lösung an.

Die SuS setzen ihre erlernten Kenntnisse zur Erstellung eines lauffähigen Programms sinnvoll ein.

Psychomotorische Lernziele:

Die SuS finden zu einer gegebenen Problemstellung den geeigneten Lösungs-Algorithmus.

Die SuS setzen ihre erlernten Kenntnisse zur Erstellung eines lauffähigen Programms sinnvoll ein.

Affektive Lernziele:

Die SuS erkennen die Notwendigkeit von endlichen Automaten in einem Informatiksystem.

Die SuS erkennen den wichtigen Einfluss von Automaten und Algorithmen in der Informatik.

4. Kompetenzentwicklung

Fachkompetenz:

Die Schüler-innen sind in der Lage Aufgabenstellungen mithilfe des Programmes Kara zu lösen.

Die Schüler-innen können grundlegende Aufgaben zum Thema endliche Automaten lösen.

Die Schüler-innen können den Marienkäfer so programmieren, dass er bestimmte Befehle ausführt.

Die Schüler-innen verstehen die Logik, welche sich hinter dem „programmieren“ befindet.

Die Schüler-innen verstehen den Aufbau von endlichen Automaten.

Lern-/Methodenkompetenz:

Die Schüler-innen können sich Informationen selbst beschaffen und diese auch verarbeiten.

Sozialkompetenz:

Die Schüler-innen entwickeln Lösungen zu gegebenen Problemen im Team.

Selbstkompetenz:

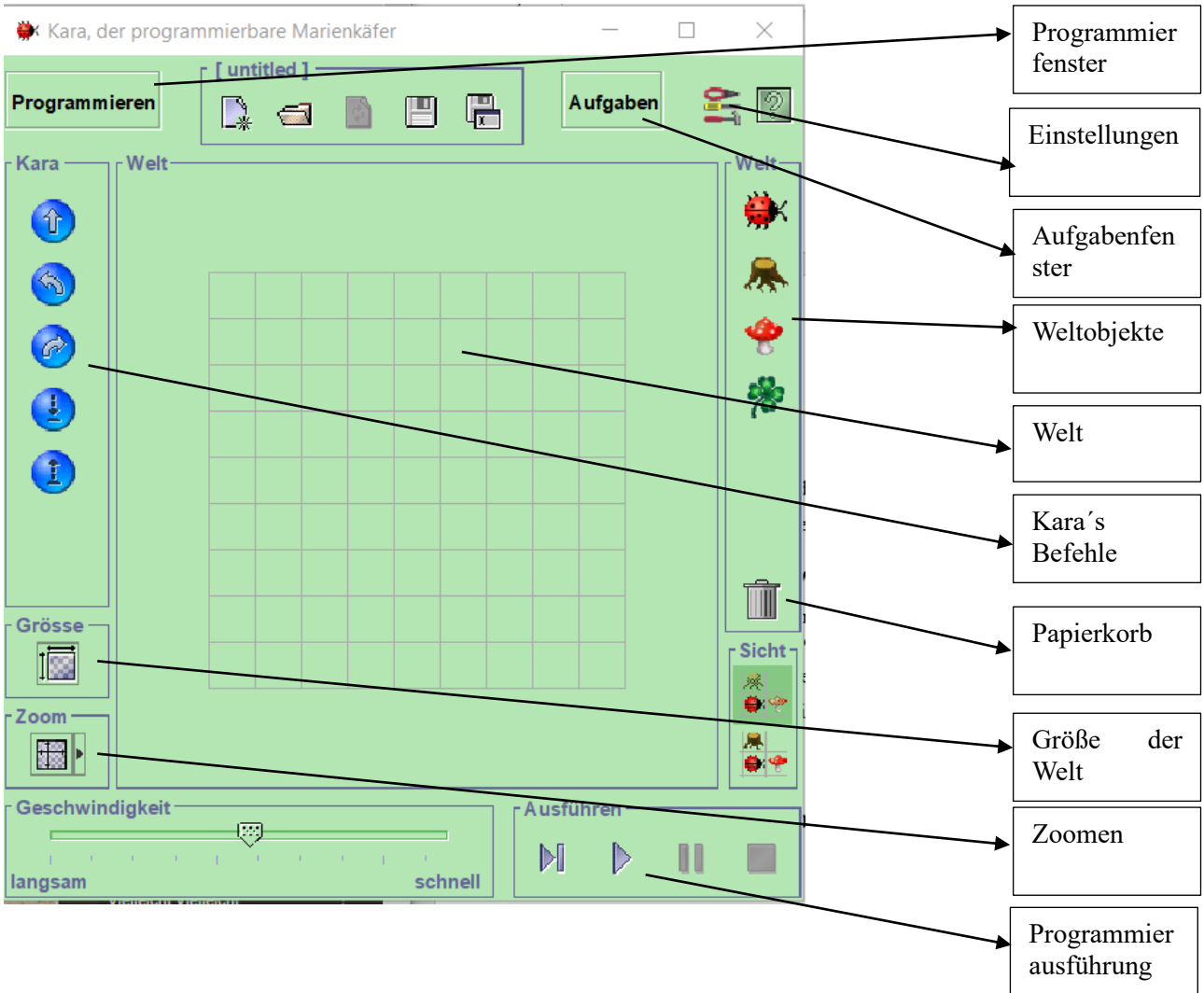
Die Schüler-innen beherrschen den Umgang mit dem Programm Kara.

Die Schüler-innen wenden das Modell der endlichen Automaten an.

Die Schüler-innen wenden Kenntnisse über die grundlegende Programmieridee und endliche Automaten im Lösen von Aufgaben an.

1. Prinzipieller Aufbau

Das Weltfenster



Das Programmfenster

The screenshot shows the 'Kara programmieren' window with the following components and callouts:

- Programmeinstellung:** A toolbar at the top containing icons for file operations (new, open, save, save as).
- Zustand bearbeiten:** A vertical toolbar on the left with icons for creating a new state, starting a simulation, and deleting a state.
- Automatenansicht:** A state transition diagram showing a state 'Laufen bis Baum' with a self-loop and a transition to a 'Stop' state.
- Anzuzeigender Zustand:** A label 'Laufen bis Baum' indicating the current state of the simulation.
- Sensorenwerte:** A section titled 'Kara macht:' with a table for sensor values:

Sensor	Value
yes	X
no	X
- Nächster Zustand:** A dropdown menu showing 'Stop' and 'Laufen bis B...'. Below it is a text input field containing an asterisk (*).
- Befehle einfügen:** A vertical toolbar on the left with icons for inserting commands.
- Zustandsansicht:** A large text area at the bottom for entering commands.

- Kara lebt in einer rechteckigen Welt, welche aus einzelne Felder besteht
- 4 Weltobjekte: Kara; unbewegliche Baumstümpfe; verschiebbare Pilze und Kleeblätter, welche Kara aufnehmen und legen kann
- um Kara zu programmieren benötigt man Sensoren. Kara kann mithilfe dieser seine Umgebung wahrnehmen.

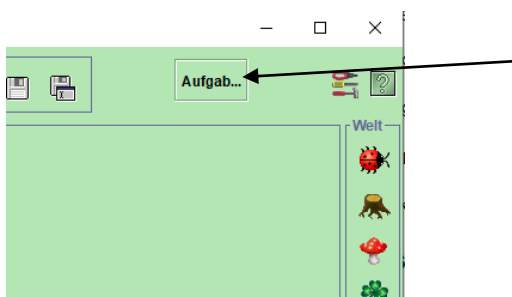


- Kara kennt nur diese Befehle:

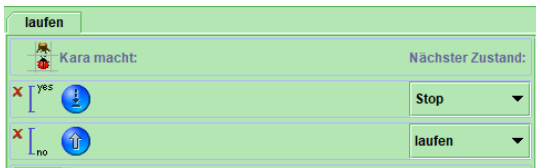


2. Handhabung

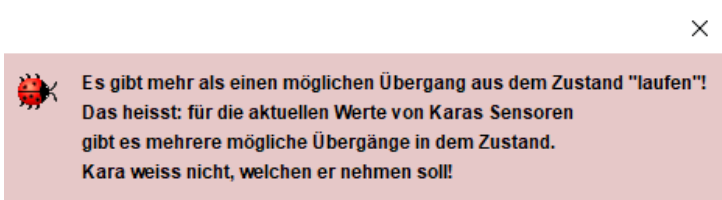
- Öffnen der Entwicklungsumgebung
- Per Drag & Drop werden die Weltobjekte in die Welt gezogen
- Auf diesem Button kann man sich Welten (Aufgaben und Lösungen) anzeigen lassen



- Im Programmierfenster wird Kara programmiert!
- Zuerst drückt man auf „new“, dann schaut man nach, welche Zustände werden benötigt, diese per Drag & Drop zu den „benutzte Sensoren“ hinzufügen
- Für den Zustand, muss man schauen welche Sensoren braucht Kara
- danach sind die Übergänge an der Reihe, sie entscheiden wann Kara in welchen Zustand geht und welche Aktionen auf die vorherige folgt



- Diese Tabelle wird zeilenweise gelesen
- Links sind die Sensoren abgebildet, welche für den Zustand ausgewählt wurden
- Der Übergang gibt die Befehle an, die Kara ausführen hat und in welche Befehle es übergehen soll
- Jeder Zustand ist eine Reihe von „wenn-dann“- Anweisungen an Kara
- Bei einem Logikfehler bspw. Zweimal „yes“ an Antwort, erscheint eine Fehlermeldung



Zustände ohne Übergänge quittiert Kara mit einer Fehlermeldung. Nur der STOP-Zustand wird von Kara als „beendend“ akzeptiert!



Zum Abschluss ist zu sagen, dass die Bedienungsanleitung sehr nützlich und informativ ist!

3. Screencast

Kurz link, Link und QR-Code zum Manual

<https://kurzelinks.de/2rvw>

<https://www.youtube.com/watch?v=sjMXv8RPbHg&t=27s>



Aufgabenstellung:

Übungsaufgaben zu Kara

Modul E-Learning

WiSe 19/20

Marc Böttger

Aufgabe 1

1.1 Öffnen Sie die Entwicklungsumgebung Kara. Ändern Sie die Weltgröße auf eine Breite von 12 und eine Höhe von 3. Setzen sie die Baumstämme um die komplette Welt herum. Füge anschließend den Marienkäfer Kara in das am weitesten links liegenden verfügbaren Feld ein. Fülle die restlichen freien Plätze beliebig mit Kleeblättern aus (mindestens 1, maximal 6).

1.2 Programmieren Sie Kara so, dass es den Weg bis zum nächsten Baumstamm läuft und da auch endet.

1.3 Setze Kara wieder in die Ausgangsposition. Programmieren Sie Kara so, dass es den Weg bis zum nächsten Baumstamm findet, aber dabei zusätzliche Arbeit verrichtet. Kara soll bei jedem freien Feld ein Kleeblatt hinlegen, falls schon ein Kleeblatt am Boden liegt, soll Kara es aufheben. Am Ende des Weges, wenn es vor einem Baumstamm steht, soll er anhalten.

Aufgabe 2

Öffnen Sie das Programm Kara. Gehen Sie in der Entwicklungsumgebung auf <Aufgaben>, suchen Sie nach „Kleeblattsuche im Wald 3(einfach)“. Lesen Sie sich die Aufgabe durch und programmieren Sie Kara so, dass es das Kleeblatt findet, aber es darf sich nicht um 180 Grad drehen. Zusätzlich öffnen Sie unter <Welten> den „Wald mit Rundgang 1“, für eine entsprechende Welt.

Aufgabe 3

Öffnen Sie das Programm Kara. Gehen Sie in der Entwicklungsumgebung auf <Aufgaben>, suchen Sie diesmal nach „Slalom (mittel)“. Lesen Sie sich die Aufgabe durch und lösen Sie die Aufgabe. Zusätzlich öffnen Sie unter <Welten> einen beliebigen „Zufälligen Slalom“, für eine entsprechende Welt.

Wichtig! Das Anklicken und benutzen der Lösungen wird mir am Computer angezeigt. Es wird eine entsprechende Benotung durchgeführt. ;)

Zusatz: Öffnen Sie die <Aufgaben>, Wählen Sie ganz unten „Random Walks (Simulation einfach)“ aus und lösen Sie die Aufgabe.