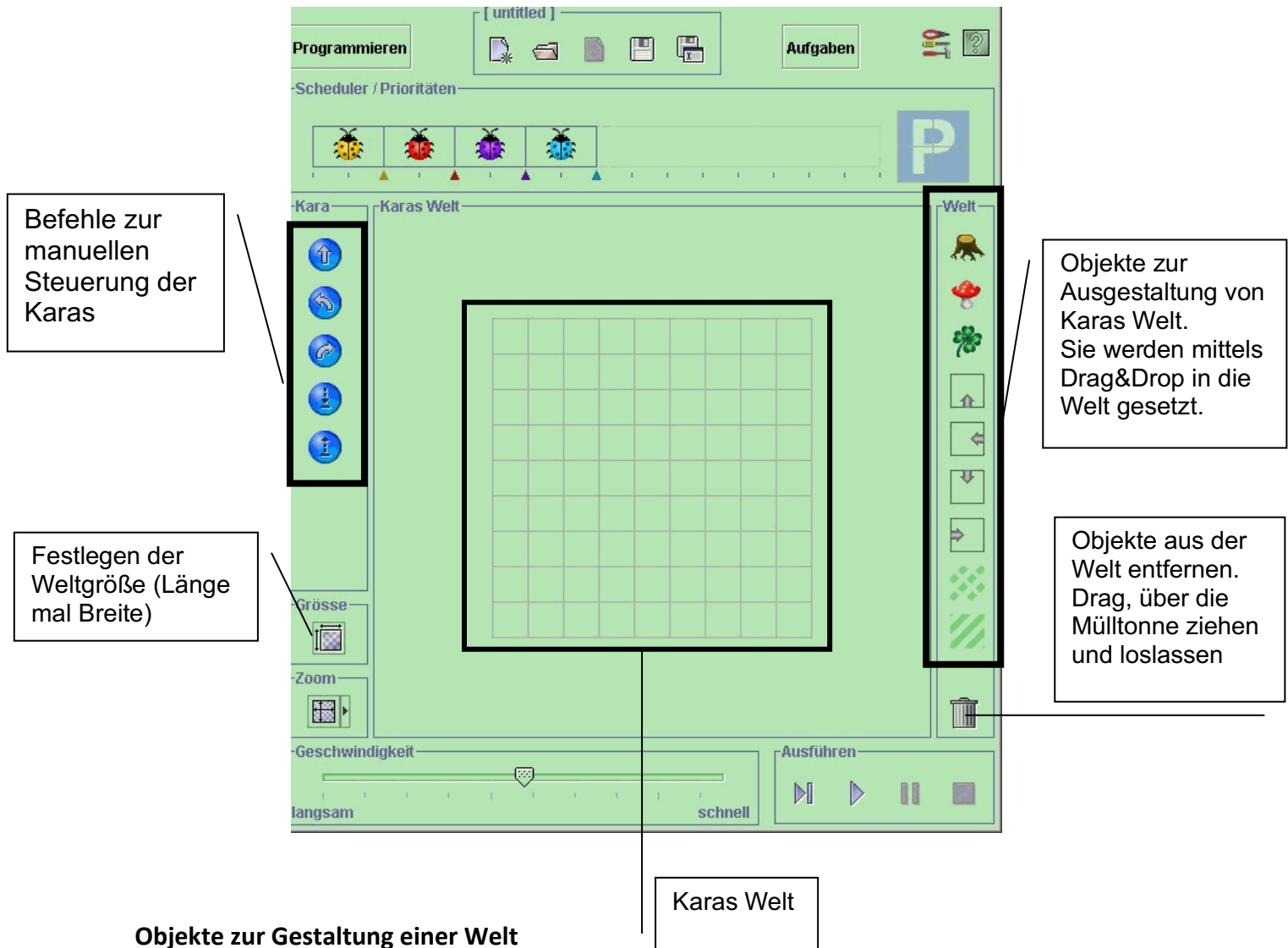


Kurzanleitung Multikara






Die Benutzeroberfläche

1. Das Weltfenster

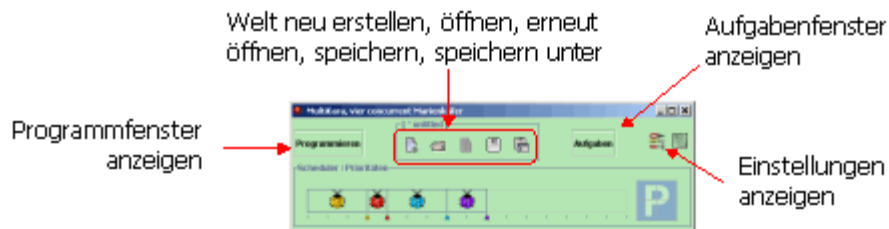


Objekte zur Gestaltung einer Welt

In der Welt des Marienkäfers gibt es:

-  Unbewegliche Baumstümpfe.
-  Pilze. Kara kann einen Pilz vor sich hin schieben. Mehrere Pilze kann Kara aber nicht gleichzeitig verschieben.
-  Kleeblätter, die Kara legen und aufnehmen kann. Kara kann unbeschränkt viele Kleeblätter legen und auch unbeschränkt viele aufnehmen.
-  4 verschiedene Marienkäfer: einen gelben, einen roten, einen violetten und einen blauen Kara. Jeder dieser Käfer hat ein eigenes Programm, mit dem er gesteuert wird. Strassensymbole, die Kara zeigen, von welcher Richtung her er ein Feld betreten darf.
-  Hier darf der Kara das abgebildete Feld nur von unten her betreten. Die Symbole können auch miteinander kombiniert werden. Das heißt, es dürfen mehrere Strassensymbole übereinander gelegt werden.

Diese Objekte können durch Drag and Drop in die Welt gezogen werden



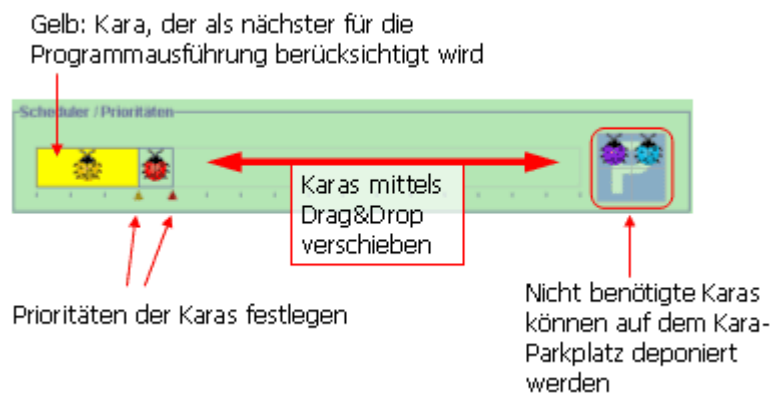
Kara Befehle

- Mache einen Schritt vorwärts
- Drehe um 90° nach links!
- Drehe um 90° nach rechts!
- Lege ein Kleeblatt hin!
- Nimm ein Kleeblatt auf!



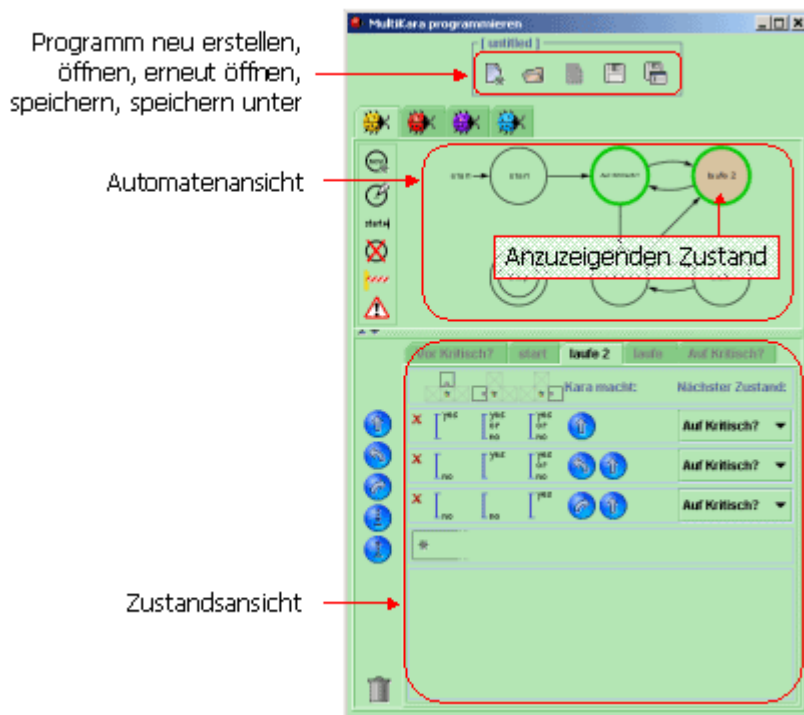
Scheduler/Prioritäten

Der Scheduler bestimmt welcher Kara zur Programmausführung an der Reihe ist..
Mit den Prioritäten wird festgelegt mit wie viel Prozessorzeit dem Kara zugeteilt wird, d.h. wie schnell er sich bewegt.



Ausführen eines Überganges mit einem bestimmten Kara (Manuelles Scheduling)

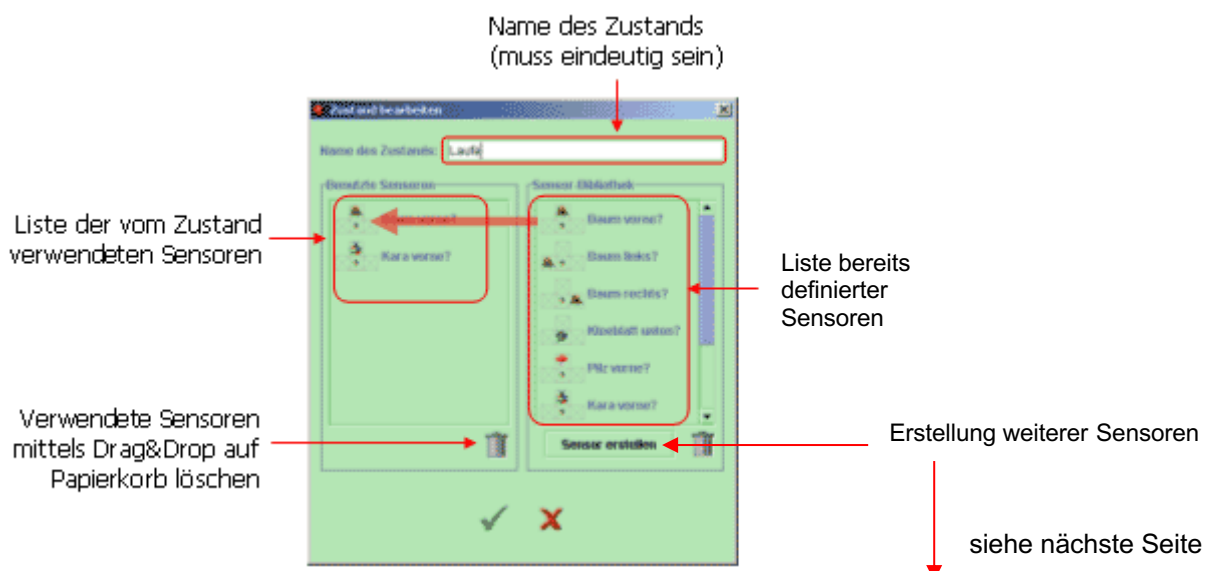
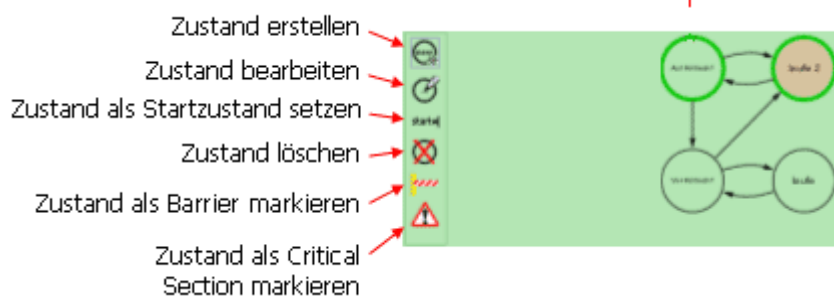
2. Das Programmfenster



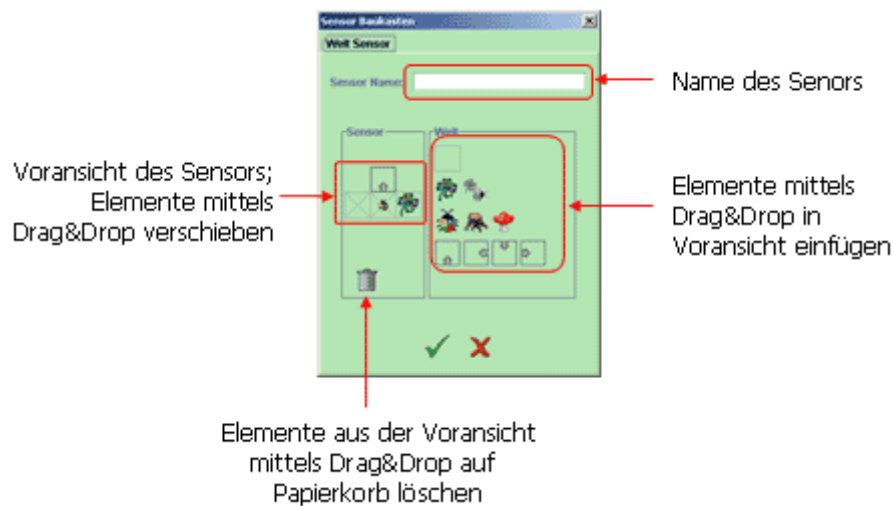
Arbeiten im Programmfenster

Zunächst muss zumindestens ein Zustand erstellt werden, der als Startzustand festgelegt wird. Jeder Zustand muss einen eindeutigen Namen erhalten und man kann Kara verschiedene Sensoren zur Verfügung stellen. (siehe Bild)

a) Zustände



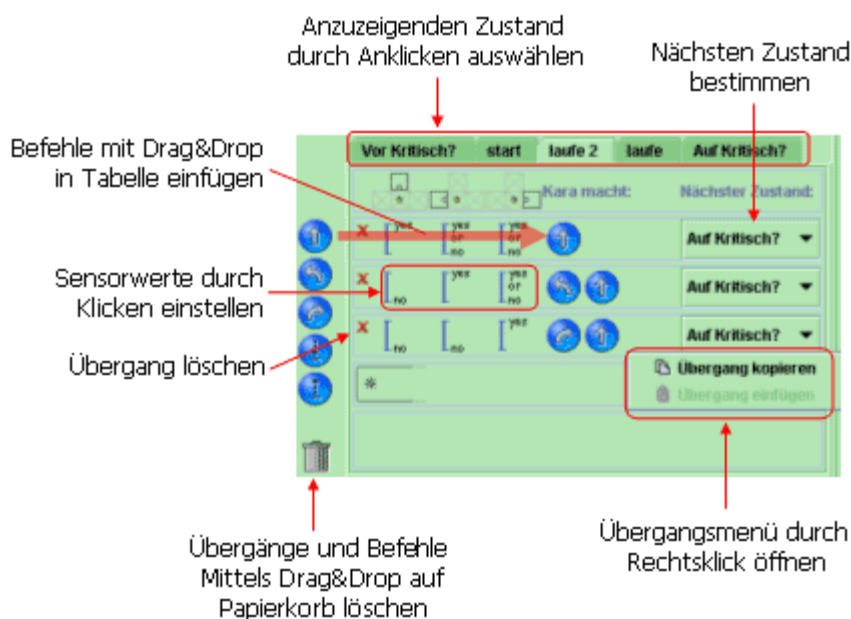
b) Sensoren



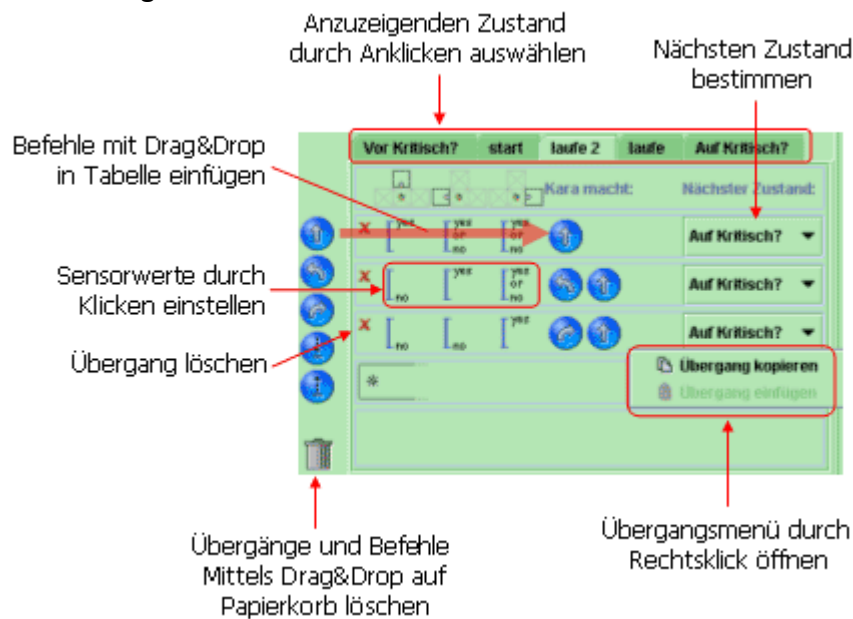
Kara Sensoren

- Stehe ich vor einem Baumstumpf?
- Ist links von mir ein Baumstumpf?
- Ist rechts von mir ein Baumstumpf?
- Stehe ich vor einem Pilz?
- Stehe ich auf einem Kleeblatt?
- Ist es erlaubt, das vorne liegende Feld von unten her zu betreten?

In der Zustandsansicht können dann die Befehle und Übergänge festgelegt werden



Die Befehle müssen im Programmfenster beim jeweiligen Zustand eingegeben werden. Die Ausführung des Befehls kann von den erhaltenen Sensorwerten abhängen.



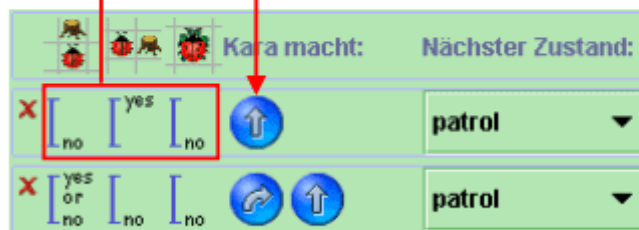
Wenn

vor Kara **kein Baum**
UND
rechts von ihm **ein Baum** ist
UND
Kara **nicht auf einem Kleeblatt** steht.

Dann

geht Kara **einen Schritt vor**
und **wechselt** in den **Zustand patrol**.

Bedingung Befehle

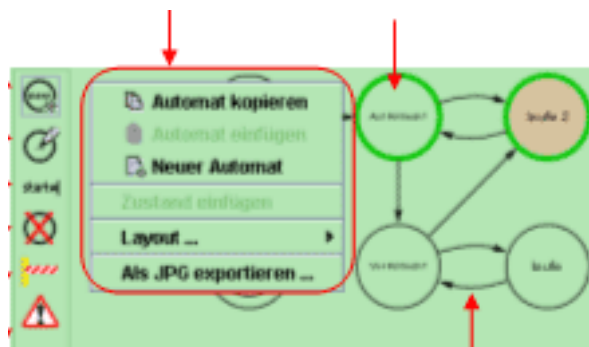


Sensor wird nicht berücksichtigt

Der Automat eines Karas kann vollständig kopiert und für einen anderen Kara verwendet werden.

Automatenmenu durch Rechtsklick in Hintergrund öffnen

Als Critical Section markierter Zustand

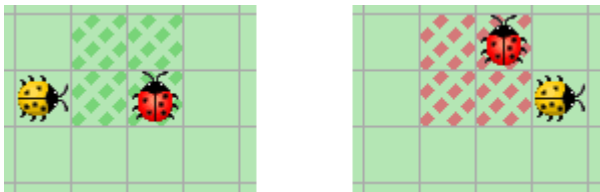


zu kopierender Automat

Prozesssynchronisation in Multikara

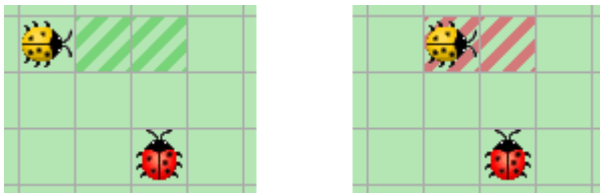
Meeting Room

Der Meeting Room dient dazu, mehrere Karas zeitlich zu synchronisieren. Ein Kara, der bei Programmausführung ein Meeting Room Feld betritt, wird Schlafen gelegt. Erst wenn alle Karas auf einem Meeting Room Feld sind, werden sie geweckt und können ihre Programmausführung fortsetzen. Dann verfärben sich die Meeting Room Felder rot, bis alle Karas den Meeting Room verlassen haben. In dieser Zeit können keine Marienkäfer den Meeting Room betreten.



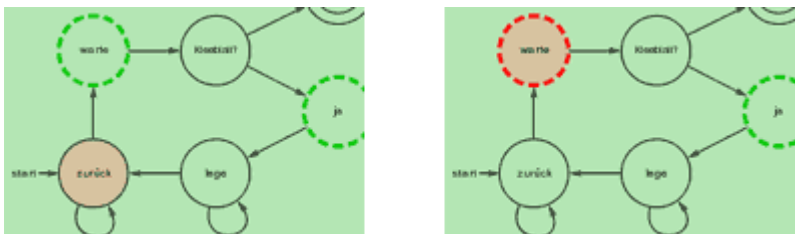
Monitor

Maximal ein Kara darf eines der Monitor-Felder besetzen. Betritt ein Kara ein Monitor Feld, so färben sich alle Monitor Felder rot. Wenn der Kara den Monitor wieder verlässt, werden die Felder wieder grün eingefärbt. Solange der Monitor besetzt ist, wird jeder Kara, der ihn betreten möchte, zurückgestellt und Schlafen gelegt.



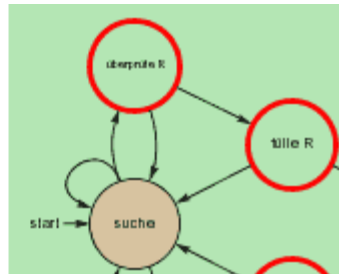
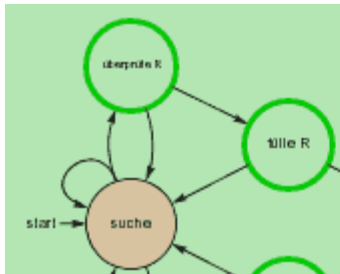
Barrier

Gekennzeichnet werden Barrier-Zustände mit einer dicken, gestrichelten Umrandung. Betritt ein Kara-Programm einen als Barrier markierten Zustand, so wird die Barrier aktiv und daher rot eingefärbt. Das Kara-Programm wird so lange schlafen gelegt, bis alle anderen Programme ebenfalls einen Barrier-Zustand betreten haben. Sind alle Kara-Programme in einem Barrier-Zustand, wechselt die Barrier auf inaktiv und wechselt die Farbe wieder zu grün.







Critical Section

Die Critical Section Zustände sind mit einer dicken Linie markiert. Betritt ein Kara-Programm einen Critical Section Zustand, so werden die Critical Sections aller anderen Kara-Programme rot eingefärbt. Allen anderen Programmen wird jetzt das Ausführen eines Überganges in eine Critical Section verboten. Die Programme werden gegebenenfalls Schlafen gelegt. Die Critical Section wird freigegeben, wenn der entsprechende Kara-Prozess einen Zustand betritt, der nicht als Critical Section markiert ist. Es kann sich also jederzeit höchstens ein Kara in einem Critical Section Zustand befinden.



Die Mechanismen im Überblick

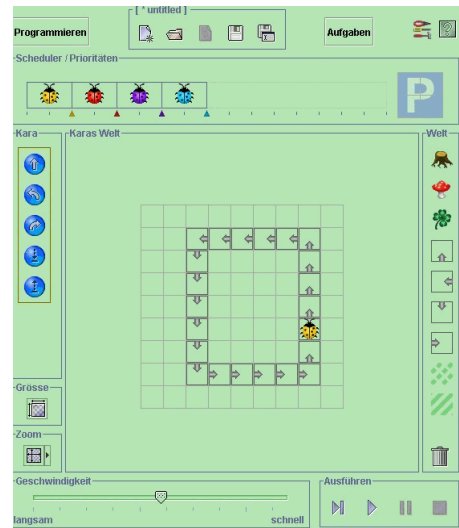
	inclusive "Synchronisation"	exclusive "Gegenseitiger Ausschluss"
Welt	 meeting room	 monitor
Automat	 barrier	 critical section

Quelle: Onlinehilfe von Kara (Lernumgebung Kara der Marienkäfer ETH Zürich)

Multi Kara Aufgabenblatt 1

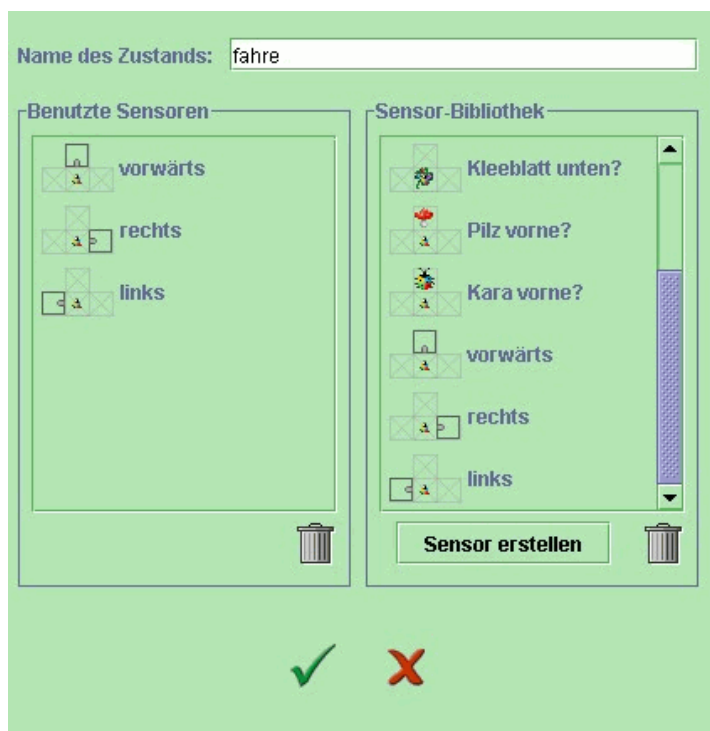
Starten Sie das Programm Kara mit der Umgebung Multi Kara – nebenläufige Programmierung. Bearbeiten Sie die folgenden Aufgaben mit Hilfe der Kurzanleitung für MultiKara

1. Erstellen Sie einen „Straßenrundkurs“ in der Welt und setzen Sie den gelben Kara auf diese Straße.
2. Bewegen Sie den gelben Kara mittels der Befehle im Weltfenster auf diesem Rundkurs.



Öffnen Sie nun das Programmfenster

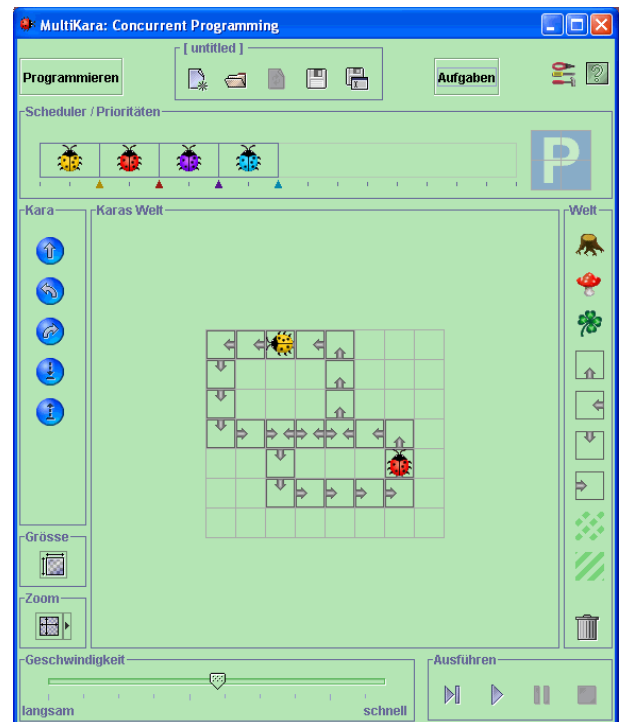
1. Erstellen Sie für den gelben Kara den Zustand *fahre* mit den folgenden Sensoren



2. Programmieren Sie den Zustand *fahre* so, dass der gelbe Kara immerfort auf dem Straßenrundkurs läuft, ohne diesen zu verlassen.

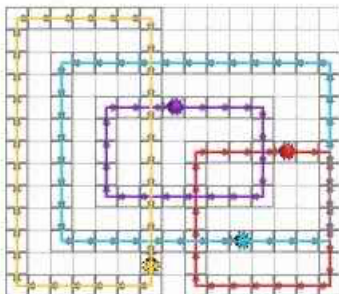
Zwei Karas (rot und gelb) möchten nun entlang ihrer Strasse fahren. Jeder Kara hat eine eigene Route (siehe Bild); Gegenverkehr oder Kreuzungen sind möglich.

3. Erstellen Sie die Routen gemäß dem Bild und kopieren Sie den erstellten Automaten zum roten Kara. Starten Sie anschließend das Programm.
4. Mit welchem Konflikt müssen Sie rechnen?
5. Wie könnte man diesen Konflikt lösen?
 → allgemeine Beschreibung
 → dann Erklärung mit Fachbegriffen
6. Verändern Sie nun das Programm so, dass beide Karas „konfliktfrei“ fahren.
 (2 Lösungsmöglichkeiten!)



Für schnelle Programmierer (freiwillig):

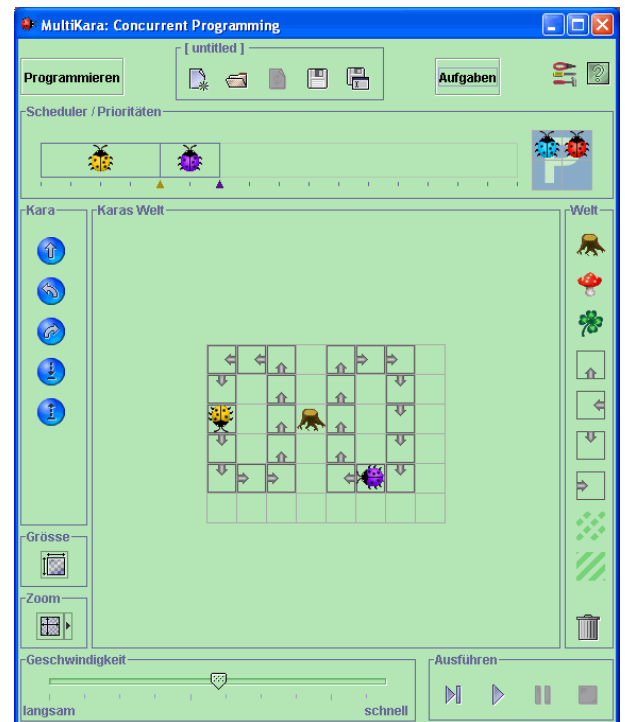
7. Laden Sie nun die **Welt Verkehrssimulation** und lösen sie die Aufgabe für vier Karas!
Hinweise Alle Karas laufen im Kreis. (siehe Bild)



Multi Kara – Aufgabenblatt 2

Zwei Karas repräsentieren Busse, die sich je auf einem Straßen-Rundkurs befinden, den sie endlos abfahren sollen. Die Bushaltestelle wird repräsentiert durch das mit einem Baum angedeutete Warte-Häuschen. Den Passagieren soll es möglich sein von einer Buslinie zur anderen umzusteigen.

1. Welche Abhängigkeiten bestehen zwischen den beiden Prozessen?
2. Mit welchen Mechanismen kann die Synchronisation der beiden Busfahrer-Käfer stattfinden.
3. Programmieren Sie die möglichen Lösungen.



Multi Kara – Aufgabenblatt 3

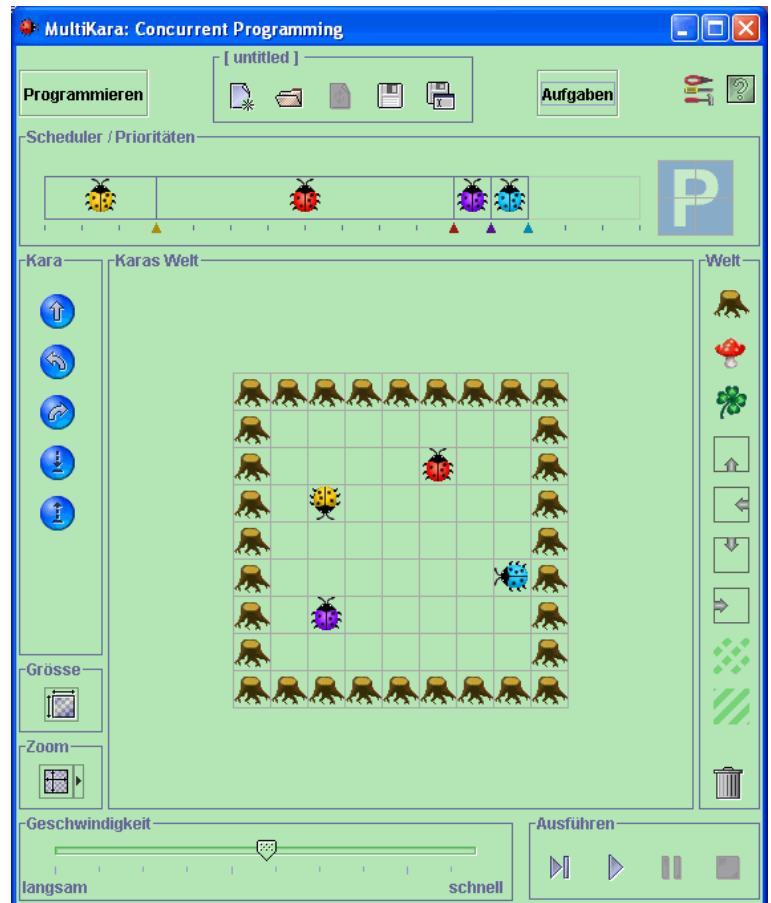
Übungsphase

→ Erproben der erlernten Konzepte

a) Aufgabe:

Vier Karas möchten ein von Bäumen begrenztes Rechteck beliebiger Größe füllen. Jeder Kara soll das gleiche Programm verwenden und alle Karas sollen am „Füllvorgang“ teilnehmen. Erstellen Sie das Programm!

Hinweis: Verwenden Sie das Barrier-Konzept!



b) Aufgabe: Erzeuger-Verbraucher-Problem (schwierig, nur für Profis!!!)

In einer leeren Welt soll ein Kara zufällig Kleeblattreihen, deren Länge ebenso zufällig ist legen. Die einzelnen Kleeblattreihen sollen einander nicht direkt berühren. Diagonal aneinander liegende Blöcke sind erlaubt. Die drei anderen Karas (Verbraucher) sollen zufällig herumlaufen und Kleeblattreihen suchen. Sobald ein Verbraucher-Kara eine Reihe von Kleeblättern gefunden hat, soll er die ganze Reihe von Kleeblättern durch eigene ersetzen. Stellen sie zunächst über das Werkzeugsymbol nichtdeterministische Automaten ein und programmieren Sie anschließend die Lösung!