

## Kara – Einführung in die Bedienung



### Inhalt:

1. Download von Kara
2. Karas Welt
3. Kara bewegt sich / Direktsteuerung
4. Programmieren von Kara
  - 4.1. Zustände definieren und bearbeiten
  - 4.2. Übergänge programmieren
  - 4.3. Ablauf des Programms
5. Tipps
6. Aufgaben

## Kara, der Marienkäfer

Kara ist ein Marienkäfer, der in einer gestaltbaren Bildschirmwelt lebt und dessen Verhalten programmiert werden kann. Im Gegensatz zu einer Programmiersprache wird das Verhalten von Kara im Form von Zuständen und deren Übergänge dargestellt.

### Download von Kara

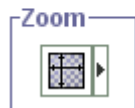
Kara kann als ausführbares Java-Programm über die URL <http://www.swisseduc.ch/informatik/karatojava/download.html> bezogen werden. Dabei müssen mehrere Varianten unterschieden werden:

- ◆ In diesem Skript wird ausschließlich die einfachste Version „Kara (3 MB)“ behandelt. Die weiteren Kara-Varianten erschließen sich einem dann fast von selbst.
- ◆ Dann muss geklärt werden, welches Java-Runtime-Environment (JRE) auf dem PC installiert ist. Dies geschieht am sichersten über den Aufruf des Java Control Panels mithilfe von „Systemsteuerung | Java“ (bzw. „Systemsteuerung | Weitere Systemsteuerungsoptionen | Java“ in der Kategorienansicht); die Betätigung des Buttons „Anwendungsinfo...“ führt zur Anzeige der installierten Java-Version. Zum Zeitpunkt der Abfassung dieses Dokuments werden im Wesentlichen die Version 1.5.x oder 1.6.x installiert sein. Dementsprechend ist die downzuladende Kara-Version auszuwählen. In diesem Skript wird auf die Version für Java 1.6 eingegangen, die Unterschiede zur Version für 1.5 sind marginal.
- ◆ Des Weiteren muss unterschieden werden zwischen einer Version mit und ohne Lösungen für die im Programm aufrufbaren Aufgabenstellungen. Wenn man auf diese im Unterricht zurückgreifen möchte, bietet sich für den Einsatz bei den Schülern die Version ohne Lösungen an.

## Karas Welt

Die Welt von Kara öffnet sich nach einem Doppelklick auf die JAR-Datei. Es öffnet sich nebenstehendes Fenster.

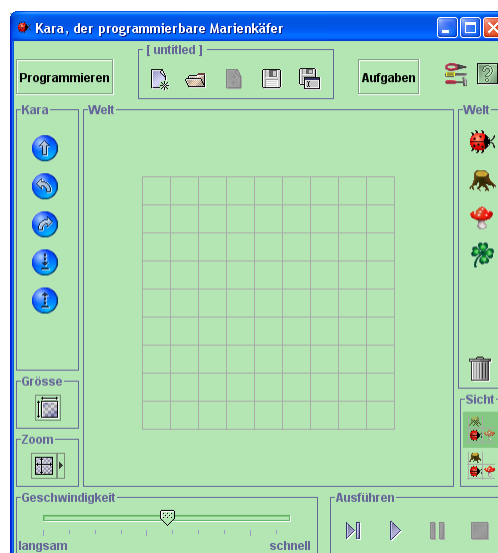
Der Bereich „Welt“ in der Mitte stellt den Bereich dar, in dem sich Kara, der Marienkäfer, bewegen kann.



Die Abmessungen dieses Bereichs kann man durch Klicken auf den Bereich „Grösse“ verändern. Gerade bei größeren Abmessungen kann es erforderlich sein, mithilfe von „Zoom“ (z. B. auf Fenstergröße) den Welt-Bereich vollständig anzeigen zu lassen. Wenn man die Alt-Taste gedrückt hält und mit der linken Maustaste gedrückt in vertikaler Richtung durch den Welt-Bereich zieht, kann man einen schnellen Zoom durchführen.

Wenn Karas Welt größer als der Bereich in der Mitte ist, kann er mithilfe der Maus verschoben werden, wenn mit der linken Maustaste auf ein Feld geklickt wird, das kein Objekt enthält, und dann gezogen wird.

Mithilfe der Werkzeuge oben rechts kann die Hintergrundfarbe des Fensters geändert werden.



Die rechts abgebildeten Objekte „Kara“ (Marienkäfer), „Baumstumpf“, „Pilz“ und „Kleeblatt“ können durch Drag-and-Drop auf die Kästchen des Welt-Bereichs gezogen bzw. innerhalb des Welt-Bereichs verschoben werden. Dabei gelten folgende Regeln:

- ◆ Kara ist einzigartig – ihn kann es in der Welt also nur einmal geben.
- ◆ Es dürfen sich keine zwei Objekte auf einem Feld befinden. Ausnahme ist die Kombination Kara und Kleeblatt bzw. Pilz und Kleeblatt.

Eine alternative Platzierungsmöglichkeit besteht darin, ein Objekt rechts einmal anzuklicken und dann durch einen weiteren Klick auf den Welt-Bereich zu setzen. Auf diesem Wege können mehrere Objekte des gleichen Typs direkt nacheinander eingefügt werden. Die Auswahl wird durch Anklicken des Bereichs zwischen den Objekten und dem Papierkorb aufgehoben.

Objekte im Welt-Bereich können durch Drag-and-Drop auf den Papierkorb gelöscht werden sowie durch Auswahl des Objekts durch einmaligen Klick rechts und weiteren Klick auf das zu löschende Objekt.

Der gesamte Welt-Bereich kann durch Rechtsklick auf den Welt-Bereich und Auswahl von „Alles löschen“ gelöscht werden. Die Abmessungen des Welt-Bereichs bleiben dabei erhalten. Im Gegensatz dazu bewirkt ein Klick auf „Neue, leere Datei“ (im Bereich oben in der Mitte), dass zusätzlich die Standard-Abmessung  $9 \times 9$  Felder wiederhergestellt wird.

Die Objekte mit Ausnahme von Kara werden im Welt-Bereich teilweise heller dargestellt, wenn im Bereich „Sicht“ die obere Variante gewählt wird. In diesem Fall werden nur die Objekte dunkler dargestellt, für die Kara in dieser Position einen Sensor besitzt (siehe unten). Für den Anfang sollte eher die untere Variante ausgewählt werden.



Beim Programmieren (siehe unten) ist es häufig erforderlich, schnell wieder die „Ausgangsposition“, also die Situation von Karas Welt zum Beginn des Programms wiederherzustellen. Aus diesem Grund sollte eine Welt sofort nach dem Erstellen abgespeichert werden. Die Welt-Datei bekommt dann die Dateierweiterung „.world“. Der Dateiname wird nach dem Speichern oben angezeigt.


Sobald sich die Welt aufgrund von manuellen Änderungen per Drag-and-Drop bzw. Direktsteuerung (siehe unten) oder nach Ausführung eines Programms (siehe unten) gegenüber der gespeicherten Version verändert hat, erscheint vor dem Dateinamen ein Stern. In diesem Fall kann durch Betätigung von „Datei erneut laden“ sofort die gespeicherte Version wiederhergestellt werden. (Bei der Abfrage „Wollen Sie die Änderungen speichern?“ bitte beherzt auf „Nein“ klicken!)


Wenn man eine Grafik der Welt z. B. für das Einfügen in Arbeitsblätter erstellen möchte, geschieht dies am einfachsten durch einen Rechtsklick auf die Welt und Auswahl von „Als jpg exportieren ...“. Im nachfolgenden Bildschirm kann man wählen, ob man die gesamte Welt (in der Vergrößerung 100 %) oder den sichtbaren Teil der Welt (in der aktuell gewählten Vergrößerung) als JPG-Grafik abspeichern möchte. Das gespeicherte Bild verwendet für die Darstellung grundsätzlich die untere Variante von „Sicht“, die Objekte werden also nicht hell dargestellt.

## Kara bewegt sich






Kara kann sich durch die Welt von einem Feld zum anderen um einen Schritt in die Blickrichtung bewegen und dabei ggf. einen Pilz vor sich herschieben. Er kann sich um  $90^\circ$  nach links oder rechts drehen bzw. Kleeblätter aufnehmen oder ablegen.


Baumstümpfe sind unbeweglich. Wenn Kara auf ein Feld laufen soll, auf dem sich ein Baumstumpf befindet, gibt es eine Fehlermeldung.

Pilze können von Kara verschoben werden – allerdings nur ein Pilz zur Zeit. Wenn sich hinter dem Pilz ein weiterer Pilz oder ein Baumstumpf befindet, gibt es beim Versuch, den Pilz zu verschieben, eine Fehlermeldung. 

Kleeblätter können von Kara aufgenommen oder abgelegt werden. Dabei ist der Vorrat an Kleeblättern, der Kara transportiert bzw. transportieren kann, unbeschränkt. Wenn Kara versucht, ein weiteres Kleeblatt auf einem Feld abzulegen, auf dem sich schon eines befindet, oder wenn er versucht, ein Kleeblatt von einem Feld aufzunehmen, auf dem sich gar keins befindet, gibt es eine Fehlermeldung. 

Kara kann mithilfe der Tasten im Bereich „Kara“ direkt gesteuert werden. Die einzelnen Tasten (= Aktionen) bedeuten der Reihe nach von oben nach unten:

- ◆ Einen Schritt vorwärts in die Richtung, in die Kara guckt; ggf. wird ein Pilz vor Kara verschoben. Kara kann über den Rand hinaus auf die andere Seite springen bzw. einen Pilz auf diesem Weg auf die andere Seite verschieben. 
- ◆ Drehung um 90° nach links; Kara bleibt auf dem Feld. 
- ◆ Drehung um 90° nach rechts; Kara bleibt auf dem Feld. 
- ◆ Kara legt ein Kleeblatt auf das Feld ab, auf dem er steht. 
- ◆ Kara nimmt das Kleeblatt von dem Feld auf, auf dem er steht. 

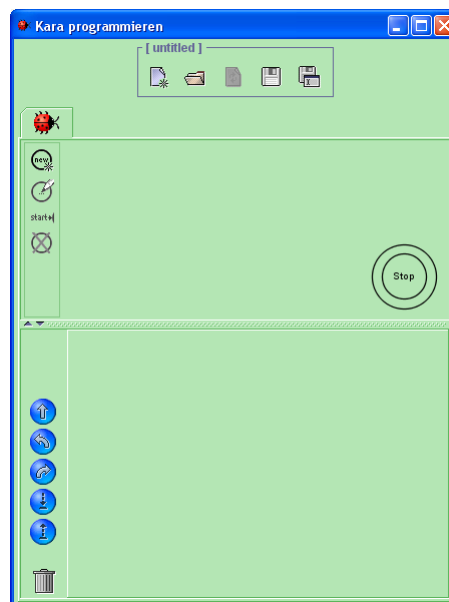
Beim Vorwärtsgen sowie beim Kleeblattablegen und -aufnehmen sind die oben genannten Einschränkungen zu beachten, die zu einer Fehlermeldung führen. 

Die Direktsteuerung sollte unbedingt vor dem Programmieren (siehe unten) durchgeführt werden, um ein Gefühl für die Steuerung von Kara zu bekommen

## Programmieren von Kara

Zum Aufruf des Programmierfensters den Button „**Programmieren**“ im Welt-Fenster betätigen. Es öffnet sich zusätzlich zum Welt-Fenster nebenstehendes Fenster, das drei Bereiche enthält:

- ◆ Oben befindet sich wie beim Welt-Fenster die Symbolleiste für die Dateioperationen.
- ◆ In der Mitte können die Zustände definiert werden.
- ◆ Unten kann festgelegt werden, wie Kara auf bestimmte Situationen reagiert.



### Zustände definieren und bearbeiten

Mithilfe der Tasten im **Zustände-Bereich** können Zustände definiert und geändert werden. Einen existierenden Zustand kann man durch Anklicken mit der linken Maustaste markieren; seine Farbe ändert sich dann je nach gewähltem Stil (siehe unten).



Mit „**Neuen Zustand erzeugen**“ kann man einen neuen Zustand anlegen und seine Eigenschaften (Name und Sensoren, siehe unten) festlegen. Mit „**Zustand bearbeiten**“



kann man die Eigenschaften eines markieren Zustandes verändern. Mit „**Zustand als Startzustand setzen**“ wird definiert, in welchem Zustand Kara startet; der gewünschte Zustand muss vorher markiert sein. Ohne Festlegung eines Startzustandes kann das



Kara-Programm nicht ausgeführt werden. Mit „**Zustand löschen**“ wird der markierte Zustand unwiderruflich gelöscht, eine Rückgängig-Funktion existiert nicht!

Der Stop-Zustand ist vorgegeben. Wenn Kara in diesem Zustand angekommen ist, endet der Programmablauf. Wenn es keinen Übergang (siehe unten) zum Stop-Zustand gibt, läuft Kara endlos. Der Stop-Zustand kann selbstverständlich nicht als Startzustand festgelegt werden.




Bei einem Zustand sind zwei Maus-Operationen möglich:

- ◆ Wenn der innere Bereich hervorgehoben ist, kann der Zustand im Zustände-Bereich verschoben werden, um z. B. die Übersichtlichkeit zu erhöhen.
- ◆ Wenn der äußere Bereich als Ring hervorgehoben ist, kann von diesem Zustand aus durch Ziehen mit der Maus ein Übergang zu einem anderen Zustand hergestellt werden. (Alternativ kann ein solcher Übergang im unteren Bereich definiert werden; siehe unten.) Ein solcher Übergang wird durch einen Verbindungspfeil symbolisiert.

Durch Klick mit der rechten Maustaste in den Zustände-Bereich können das Layout oder das Aussehen („Stil wählen ...“) verändert werden. Ein Zoomen bzw. ein Verschieben sind wie beim Welt-Fenster mithilfe der Alt-Taste und der linken Maustaste bzw. mit der linken Maustaste möglich.

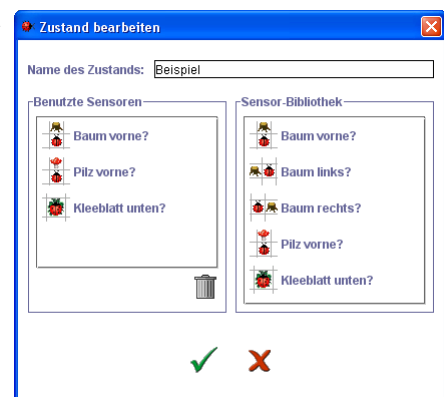
Auf die Speicherung des Programmierfensters als Grafik wird weiter unten eingegangen.

Jedem Zustand können bei dessen Anlegen oder Bearbeiten ein oder mehrere der folgenden Sensoren zugeordnet werden, mit denen Kara seine Umgebung inspizieren kann:

		
Befindet sich auf der rechten bzw. linken Seite bzw. vor mir ein Baumstumpf?	Befindet sich vor mir ein Pilz?	Befindet sich unter mir ein Kleeblatt?

Zum Einfügen eines Sensors in einen Zustand wird dieser mit der Maus aus der **Sensor-Bibliothek** in das Feld der **benutzten Sensoren** gezogen. Mithilfe von Drag-and-Drop kann auch die Reihenfolge der Sensoren geändert werden.

Zum Löschen wird der jeweilige Zustand mit der Maus aus dem Feld der benutzten Sensoren auf den Mülleimer gezogen.



## Übergänge programmieren

Eine fertige Programmierung von Übergängen könnte folgendermaßen aussehen:

Beispiel	Kara macht:	Nächster Zustand:
	yes or no	Beispiel
	yes no no	Stop
	no yes no	Stop
	no no no	Beispiel
* ...		

Jede Zeile stellt einen Übergang dar. Zum Einfügen eines neuen Übergangs klickt man auf den Stern unten, zum Löschen eines Übergangs auf das rote Kreuz.

Das Register oben (hier „Beispiel“) zeigt an, von welchem Zustand aus die angezeigten Übergänge starten.

Die drei dargestellten Sensoren sind die im Zustand „Beispiel“ benutzten Sensoren (siehe Screenshot oben).

Bei der Ausführung des Programms könne folgende drei Fälle auftreten:

1. Wenn kein Sensor anspricht (Zeile 4), geht Kara einen Schritt nach vorne und bleibt im Zustand „Beispiel“. Das Programm wird also in diesem Zustand fortgesetzt.
2. Wenn Kara auf einem Kleeblatt steht, wird – unabhängig vom Zustand der beiden anderen Sensoren – (Zeile 1) das Kleeblatt aufgehoben. Das Programm bleibt im Zustand „Beispiel“.

3. Wenn Kara vor einem Baum oder vor einem Pilz steht (Zeilen 2 bzw. 3), geht er sofort in den Stop-Zustand über. Das Programm hält dann an.  
Bei diesen Übergängen wird Folgendes deutlich: Der Sensor für das Kleeblatt kann „no“ sein, da ein Kleeblatt ggf. vorher mithilfe der Zeile 1 aufgehoben wurde. Der jeweils andere Sensor (Baum bzw. Kleeblatt) kann „no“ sein, da ein Zusammentreffen beider Fälle nicht möglich ist.

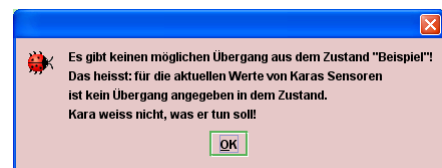
Eine – beim Programmieren übrigens häufig notwendige – Analyse der Sensor-Kombinationen zeigt, ob alle Möglichkeiten durchdacht wurden:

- ◆ In der ersten Zeile wird der Fall erfasst, dass unter Kara ein Kleeblatt liegt, die anderen Sensoren sind hierbei egal („yes or no“). Die anderen drei Zeilen enthalten jeweils das Gegenteil dieser Aussage, dass nämlich kein Kleeblatt unter Kara liegt. Alle Möglichkeiten mit „yes“ beim Kleeblatt-Sensor sind also erfasst.
- ◆ Von den weiteren vier Sensor-Kombinationen, bei denen der Kleeblatt-Sensor „no“ ist, fehlt nur die Möglichkeit „yes, yes, no“, die aber nie vorkommen kann, weil nicht gleichzeitig ein Baum und ein Pilz vor Kara sein können.

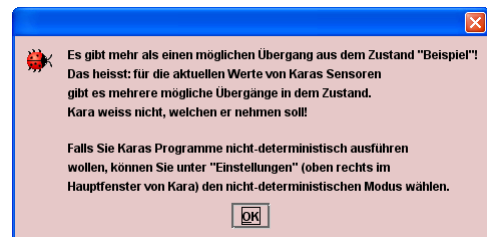
Für die Ermittlung der Anzahl der möglichen Sensor-Kombinationen hilft das Dual-System, indem die Sensoren als Stellenwertsystem mit den Ziffern „1“ für „yes“ und „0“ für „no“ verwendet werden. An dieser Stelle bietet sich im Unterricht ggf. ein Exkurs in das Dual-System an.

Manchmal kann es auch sinnvoll sein, die Reihenfolge der Sensoren durch Bearbeiten des Zustands (siehe oben) so zu verändern, dass der Überblick über die Möglichkeiten besser gewahrt wird.

Wenn nicht alle vorkommenden Möglichkeiten erfasst sind, kann es beim Programmablauf (siehe unten) zu nebenstehender Fehlermeldung kommen. Sie bedeutet, dass für diese Situation (= Sensorkombination) kein Übergang definiert wurde, so dass das Programm an dieser Stelle nicht weitergeführt werden kann.



Eine weitere mögliche Fehlermeldung ist nebenstehend zu sehen. In diesem Fall gibt es für die aktuelle Sensor-kombination mehr als eine Zeile mit einer Übergangsmöglichkeit, die zutreffen könnte. Dies ist jedoch bei einem deterministischen endlichen Automaten nicht zugelassen, so dass das Programm nicht weitergeführt werden kann. Es ist zwar möglich, über das Werkzeug-Symbol (siehe oben) Kara in einen nicht-deterministischen Modus zu versetzen; dies ist jedoch für den Unterricht in der Schule nicht empfehlenswert.



Das Programmieren selbst wird folgendermaßen durchgeführt:

- ◆ Zuerst wird durch Klicken auf den Stern einer neuer Übergang erstellt.
- ◆ Der Zielzustand ist zunächst der aktuelle Zustand, er kann jedoch rechts im Drop-down-Menü beliebig ausgewählt werden.
- ◆ Die Sensor-Einstellungen nimmt man durch Mausklick vor, wobei die Einstellung zyklisch zwischen „yes or no“, „yes“ und „no“ wechselt.
- ◆ Die gewünschten Aktionen können mithilfe von Drag-and-Drop aus dem Vorrat links an die gewünschte Stelle gezogen werden. Die Reihenfolge der Aktionen kann ebenfalls durch Drag-and-Drop verändert werden. Eine nicht mehr benötigte Aktion wird durch Ziehen auf den Mülleimer gelöscht.

Das Exportieren des Zustände-Bereichs sowie der Zustände mit ihren Übergängen als jpg-Datei ist zwar möglich, aufgrund eines Bugs im Programm werden jedoch beim ersten Zustand die Übergangszeilen mit der Kopfzeile vermischt.

Welt- und Programm-Datei sollten den gleichen Namen (mit verschiedener Dateieindung) tragen, damit die Zusammengehörigkeit sofort deutlich ist. Es können aber auch beliebige Welt-Dateien mit einer Programm-Datei verknüpft werden.

## Ablauf des Programms

Die Programmkontrolle geschieht mithilfe der Bereiche „**Ausführen**“ und „**Geschwindigkeit**“ im Welt-Fenster.

Die einzelnen Tasten im Ausführen-Bereich ermöglichen den schrittweisen Ablauf des Programms, den dauerhaften Ablauf des Programms, dessen Ablaufgeschwindigkeit durch den Regler im Bereich „Geschwindigkeit“ bestimmt wird, das Unterbrechen und das Abbrechen des Programmablaufs.



Während des Programmablaufs wird im Programmierfenster der schrittweise Ablauf dargestellt: Im Zustandsdiagramm wird der aktuelle Zustand farblich hervorgehoben. Bei dem jeweiligen Zustand ist die Übergangszeile hervorgehoben, die aufgrund der Sensorkombination aktuell abgearbeitet wird; dabei ist der nächste abzuarbeitende Aktion nochmals farblich unterlegt.

Wenn die Pause-Taste betätigt wird, muss zuerst die Übergangszeile vollständig abgearbeitet werden, ehe das Programm in der Ausführung pausiert.

Die Ablaufgeschwindigkeit kann während des Ablaufs verändert werden, um z. B. in einer bestimmten Situation die Geschwindigkeit soweit zu reduzieren, dass Einzelschritte genau beobachtet werden können.

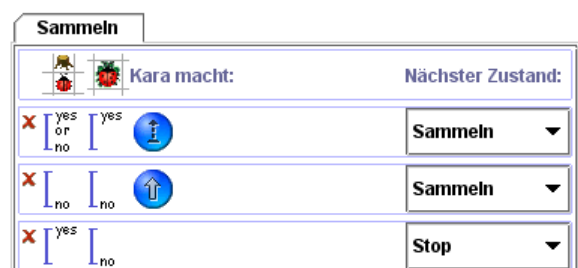
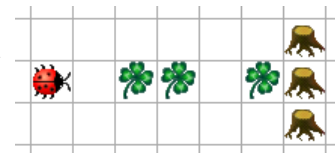
## Tipps

Im Unterricht sollte mit sehr einfachen Situationen begonnen werden, um den Schülern einen guten Einstieg in die Idee hinter der Programmierung von Kara zu geben. Insbesondere betrifft dies das Denken in Zuständen.

Beispiel:

Kara soll alle Blätter bis zu einem Baumstumpf aufnehmen und vor diesem anhalten.

Die Lösung kann sehr unterschiedlich aussehen, obwohl dieselbe Funktionalität erfüllt wird:



Noch gravierender können die Abweichungen sein, wenn mehrere Zustände verwendet werden. Hier muss häufig entschieden werden, ob das Programm eher übersichtlich sein soll, indem möglichst viele Zustände mit jeweils wenigen Aktionen pro Übergang gewählt wird, oder ob möglichst komplexe Kombinationen von Übergangsaaktionen in möglichst wenig Zuständen zusammengefasst werden.

Für den Unterrichtsablauf sind zwei weitere Aspekte von besonderer Bedeutung:

- ◆ Kommt das Programm mit Sondersituationen (z. B. Kara startet auf einem Kleeblatt bzw. vor einem Baum) zurecht?

Aus dieser Fragestellung kann für den Unterricht auch ein Spiel gemacht werden: Eine Gruppe kann eine Aufgabe entwerfen und einer anderen Gruppe zur Lösung vorlegen. Nachdem diese „gelöst“ wurde, wird an einer Beispiel-Welt der aufgabe-stellenden Gruppe getestet, ob das Programm der anderen Gruppe mit möglichen Sondersituationen zurechtkommt.

Als Beispiel für solche Sonderfälle können die mit dem Programm mitgelieferten Beispielaufgaben mit den verschiedenen Welten sein, insbesondere die Welt „einfacher Wald 3“ bei der Aufgabe „Kleeblattsuche im Wald I (einfach)“

- ◆ Die Idee, ein Feld (z. B. das Start-Feld) mit einem Kleeblatt zu markieren, sollte recht früh eingeführt werden, weil dies sehr häufig gebraucht wird.

Falls im Anschluss an Kara theoretische Informatik mit endlichen Automaten betrieben wird, ist darauf zu achten, dass es bei der Benutzung von Kara zu Fehlvorstellungen kommen kann:<sup>1</sup>

- ◆ Kara-Programme haben in der Regel Schleifen, bei denen ein Übergang den Zustand nicht verlässt. Bei endlichen Automaten ist dies häufig nicht der Fall.
- ◆ Endliche Automaten haben eventuell mehr als einen Stop-Zustand und lassen auch Übergänge aus dem Stop-Zustand heraus zu.

## Aufgaben

Das Kara-Programm liefert eine vielfältige Aufgabensammlung mit, die über den Button „**Aufgaben**“ zu erreichen ist. Die Aufgaben sind mit Schwierigkeitsgrad angegeben und enthalten Beispiel-Welten, an denen die Lösung getestet werden kann.

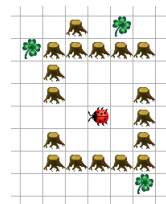
Sofern man beim Download eine Kara-Variante mit Lösungen gewählt hat, werden auch diese angeboten.

Im WWW gibt es sehr viele Aufgabensammlungen zu Kara. Stellvertretend seien hier folgende genannt:

- ◆ <http://www.cz.j.th.schule.de/neueseite/fachbereiche/informatik/kara/aufgaben.html> (einfachste Aufgaben, von SchülerInnen gestellt)
- ◆ <http://www.sn.schule.de/~infoms/kara/> (enthält vor allem einfachste Aufgaben)
- ◆ <http://www.swisseduc.ch/informatik/karatojava/kara/materialien.html>

Ein eigenes Beispiel zum Schluss:

Kara befindet sich in einem „Innenhof“, der von einigen Bäumen umgeben ist und guckt in Richtung eines Ausgangs. Er soll den Bereich außerhalb des „Innenhofs“ von Kleeblättern reinigen und anschließend wieder in den Innenhof zurückkehren. Der Innenhof hat eventuell mehrere Ausgänge (mit der Breite von einem Kästchen), enthält selbst aber keine Kleeblätter. Die Begrenzung des „Innenhofs“ darf auch „Ausstülpungen“ nach außen wie z. B. den Baum oben haben. Nicht zulässig sind Sackgassen. (Erweiterung: Kara guckt eventuell nicht in Richtung eines Ausgangs.)



<sup>1</sup> zitiert nach Kerstin Strecker vom Max-Planck-Gymnasium Göttingen