

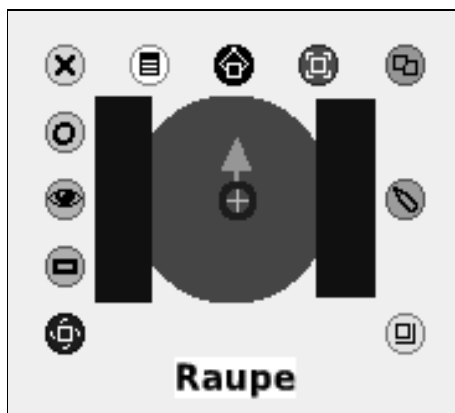
# Propädeutische Algorithmik und Objektorientierung mit Etoys

von Rüdiger Baumann

Die anfassbare, begreifbare Welt  
besteht aus vielfältigen Objekten.  
*Koerber/Peters (2003)*

Der „Hundert-Dollar-Laptop“ XO-1 (bzw. die als Ersatz in LOG IN angebotene LiveCD, siehe Rohrmoser, 2009, S.70ff.) bietet einen reichen Schatz an Werkzeugen zum kreativen Umgang mit dem Computer. Angenommen, dieser Schatz (oder sollte man ihn nicht eher ein „Sammelsurium“ nennen?) wird mit dem Ziel informatischer Bildung verwendet – wie findet man dann das am besten geeignete Werkzeug?

Geht es beispielsweise um die „objektorientierte Sichtweise“, die in den *Bildungsstandards für die Sekundarstufe I* „als durchgängiges, grundlegendes Prinzip“ empfohlen wird (AKBSI, 2008, S.27), so stehen mit SCRATCH und SQUEAK-Etoys mindestens zwei geeignete Werkzeuge zur Wahl. Nachdem für SCRATCH schon vielfältige Unterrichtserfahrungen vorliegen, soll im Folgenden für das anspruchsvollere Werkzeug Etoys an einigen Beispielen gezeigt werden, wie es sich im Anfangsunterricht einsetzen lässt. Es wird die These vertreten, dass propädeutische Objektorientierung und Algorithmik bereits in der Grundschule vermittelt werden können und sich somit die Diskussion „OOP zuerst oder später?“ als überflüssig erweist. Ferner wird behauptet und an Beispielen belegt, dass alle wesentlichen Kompetenzen der Bildungsstandards mit Etoys erreichbar sind.



**Bild 1:**  
Ein Objekt  
mit seinem  
„Heiligenschein“  
(Halo) von  
Symbolen.

Eine sprachliche Anmerkung vorweg: In Analogie zu „E-Mail“ müsste man eigentlich „E-Toys“ schreiben (wörtl.: „elektronische Spielzeuge“, vielleicht übersetzbar mit „E-Zeugs“); bei Freudenberg/Hancl/Mietzsch (2007) liest man „eToys“. Im Englischen ist „eToys“ als Plural zu verstehen („electronic“, „educational“, „exciting“, „exploratory“; siehe Allen-Conn/Rose, 2003); er bezieht sich auf die Modelle, Simulationen und Spiele, die sich mit dem „Ideenprozessor“ erstellen lassen. Der Singular „eToy“ bezeichnet dann ein einzelnes, von Schülern entwickeltes, Spiel(zeug) bzw. Projekt.

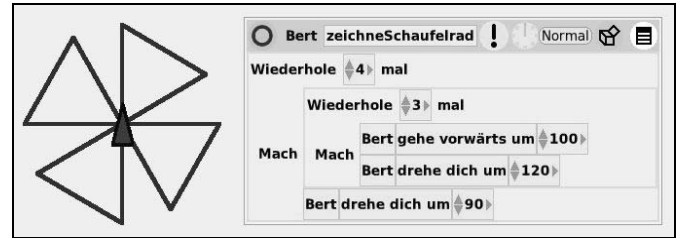
## Überblick und Bezug zu den Bildungsstandards

Gemäß Bildungsstandards beschäftigen sich die Schülerinnen und Schüler zwar schon ab Jahrgangsstufe 5 mit Algorithmen, die Umsetzung in Programme ist aber erst ab Stufe 8 vorgesehen (AKBSI, 2008, S.32). In den Lehrbüchern „für bayerische Gymnasien“ dürfen die Schüler sogar frühestens in Klasse 9 oder 10 das erste (JAVA-)Programm schreiben. Demgegenüber sieht das neue Schweizer Lehrwerk vor, dass bereits am Ende der Grundschulzeit (LOGO-)Programme verfasst und ausprobiert werden (Hromkovic, 2008). Ein Unterrichtsversuch in Baden-Württemberg zeigt, dass im Fachunterricht (Mathematik, Musik, Englisch) der Klassenstufe 5 informatische Grundkonzepte (inklusive Algorithmik) mittels LOGO gut vermittelbar sind (Wursthorn, 2008).

Im ersten Abschnitt des vorliegenden Beitrags werden einfache Algorithmen (Skripte) der „Schildkröten-Geometrie“, wie sie seit LOGO bekannt und beliebt sind, erarbeitet. Während hier im Wesentlichen imperativ bzw. prozedural gedacht wird, treten die Objekte im zweiten Abschnitt als (virtuelle) Wesen auf, die Sensoren besitzen und interagieren können. Der dritte Abschnitt befasst sich mit dem Thema „Animationen und Spiele gestalten“ und zeigt außerdem, wie vorgegebene (etwa dem Internet entnommene) Etoys-Projekte analysiert und gegebenenfalls verbessert werden können. Schließlich wird ein Beispiel für „gestaltendes Lernen“ gegeben. (vgl. Zahn, 2009)



**Bild 2:**  
Ein Skript  
(Anweisungs-  
folge) und  
sein Ergebnis  
(Beispiel 1).



**Bild 3:**  
Skript mit zwei geschachtelten Schleifen (Beispiel 2).

## Geometrische Algorithmen

In Etoys ist jedes Objekt zugleich eine Papert'sche Zeichenschildkröte und somit, geeignet programmiert, zu ästhetisch ansprechenden und interessanten Zeichnungen fähig. Mit ihr können sowohl die „algorithmischen Grundbausteine“ als auch ein naiver Objektbegriff erarbeitet werden.

### Objekte und Skripte

#### Beispiel 1: Der Informatik-Biber als Raupe

Eine Raupe (Kettenfahrzeug) wird durch ein Programm gesteuert, und zwar hört sie auf die Kommandos *links*, *rechts*, *vor*, *zurück*. Sie hinterlässt dabei eine Spur wie auf Bild 2 zu sehen (Pohl/Hein/Bastisch, 2008, S.13). Diese Raupenspur soll nun in Etoys erzeugt, d.h. die Raupe so gesteuert werden, dass die Spur auf dem Bildschirm entsteht.

Zuerst wird der Malkasten aus dem „Lager“ auf die Arbeitsfläche gezogen und die Raupe gemalt. Damit ist ein Objekt geschaffen, das man anfassen und über den Bildschirm schieben kann (Bild 1, vorige Seite). Um die Raupe herum ist – wie eine Art Heiligenschein oder Lichthof (Halo) – ein Kranz farbiger Symbole (runder Knöpfe) angeordnet, mit denen sich das Objekt handhaben (drehen, verschieben, verdoppeln, vergrößern, verkleinern, löschen) lässt.

Um die Eigenschaften und Fähigkeiten eines Objekts zu erfahren, wird (durch Anklicken des türkisfarbenen Knopfes mit dem Auge) der sogenannte *Betrachter* (engl.: *viewer*) geöffnet. Er zeigt in Gestalt von „Fliesen“ (oder Kacheln), die sich auf die Arbeitsfläche ziehen und aneinanderkoppeln lassen, die Anweisungen, denen das Objekt gehorcht.

Wir wählen zunächst den Block *Skripte* und ziehen ein leeres Skript auf die Arbeitsfläche. Aus dem Block *Stifte* holen wir die Kachel *lösche alle Stiftspuren* und koppeln sie an *Skript1* an, nachdem wir das Prädikat *Stift unten* auf *wahr* gesetzt haben. Sodann öffnen wir

den Block *Bewegung*, fügen die Kachel *gehe vorwärts um 5* an, ersetzen die 5 durch 100 – und so weiter (wie in Bild 2 gezeigt). Klicken auf das gelbe Ausrufezeichen veranlasst die Raupe, das gewünschte Bild zu zeichnen.

Das folgende Beispiel, angelehnt an Aufgabe 3.06 der Bildungsstandards (AKBSI, 2008, S.33), dient zur Einführung von Wiederholungsanweisungen (Zähl-schleifen).

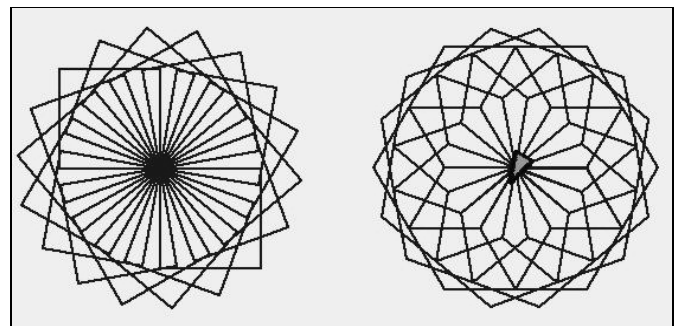
#### Beispiel 2: Das Schaufelrad

Der Informatik-Biber (er nennt sich übrigens Bert) soll eine Grafik wie in Bild 3 erstellen.

Bert zeichnet (in Gestalt eines gleichseitigen Dreiecks) zuerst eine „Schaufel“ und wiederholt diesen Vorgang dann viermal. Den „algorithmischen Grundbaustein“ *Wiederhole <Anzahl> mal* verschaffen wir uns durch Anklicken des gelben Piktogramms im Kopf des Skripts, das ein „Lager“ (Kasten mit offenem Deckel) symbolisiert (siehe Bild 3).

#### Aufgabe 1.1:

- Der Biber soll eine Quadratrossette (Bild 4) zeichnen, d.h. ein Quadrat mehrfach um einen bestimmten Winkel drehen.
- Dasselbe für ein beliebiges regelmäßiges Vieleck (z. B. Sechseck, siehe Bild 4, rechts).



**Bild 4:** Quadratrossette (links) und Sechseckrossette (Aufgabe 1.1).

**Bild 5:**  
Variable und  
Wertzuweisung  
(Beispiel 3).



## Strukturierung als „mächtige Idee“

Zur Gliederung eines Skripts (Algorithmus) in überschaubare Teile kann man mehrere Anweisungen zu einem neuen Skript zusammenfassen, mit einem Namen versehen und unter diesem Namen „aufrufen“, d.h. aktivieren. Eine solcher *Teilalgorithmus* lässt sich immer da verwenden, wo in unterschiedlichem Kontext gleichartige Tätigkeiten zu verrichten sind. Es handelt sich um die „mächtige“ oder „schlagkräftige“ Idee (*powerful idea*) der *Strukturierung* bzw. *Modularisierung*. Das folgende Beispiel führt zugleich den Begriff der Variablen ein.

### Beispiel 3: Schachteldreiecke

In ein gleichseitiges Dreieck soll ein Dreieck gleicher Art, aber halber Größe eingefügt werden – und dies insgesamt dreimal.

Nachdem das Zeichengerät erstellt ist, legen wir ein Skript zum Zeichnen eines gleichseitigen Dreiecks an; die Länge der Seiten halten wir in einer Variablen *seitenlänge* fest. Wir erschaffen sie durch Klicken auf den rosa Knopf mit dem V-Symbol. Eine solche Variable kann man sich als Schachtel vorstellen, in die sich verschiedene Dinge (Zahlen, aber auch Farben oder Objekte) legen lassen. Der nach links gerichtete Pfeil ← bezeichnet die

sogenannte *Wertzuweisung* (kurz: Zuweisung). In Bild 5 (rechts) wird beispielsweise der Variablen *seitenlänge* der Wert 150 zugewiesen. Diese Variable gehört, wie die Kachel ausdrückt, zum Objekt „Rita“; es handelt sich um ein sogenanntes *Attribut* (Ritas).

Zum Zeichnen des Schachteldreiecks legen wir ein neues Skript an und bauen eine Zählschleife ein. Nach dem Zeichnen des Dreiecks wird Ritas Seitenlänge durch die Anweisung *seitenlängeMalnehmen* mit 0.5 halbiert, und jene rückt um die neue Seitenlänge vor, dreht sich um 60 Grad nach links und wiederholt das Ganze. Die beiden Schaltflächen (Bild 6 links unter der Zeichnung) erhalten wir, wenn wir auf den Knopf mit dem schwarzen Menü-Symbol rechts klicken („Knopf, um dieses Skript auszuführen“). Es handelt sich um Objekte, deren Eigenschaften (Farbe, Rand usw.) beliebig veränderbar sind; auch die Beschriftung der Schaltfläche ist ein Objekt, das sich nach Wunsch gestalten lässt (Schrifttyp, Größe usw.).

**Aufgabe 1.2:** Die Schüler und Schülerinnen, die sich an der polnischen *Olimpiada informatyczna* beteiligen, tun dies unter einem geometrisch interessanten Emblem (Bild 7). Seine Umrisse sollen gezeichnet werden.

**Bild 6 (rechts):**  
Die Skripte zum Zeichnen des Schachteldreiecks (Beispiel 3).

**Bild 7 (unten):**  
Emblem der polnischen Informatik-Olympiade (Aufgabe 1.2).

