

Ein Zeichenprogramm mit Snap!¹

In den folgenden Aufgaben werden wir nach und nach ein kleines Zeichenprogramm mit Snap! erstellen.

Figuren zeichnen

Aufgabe 1: Das erste Objekt (Sprite), das wir benötigen ist ein Stift.

Den können Sie selbst zeichnen, indem Sie im Bereich für die Sprites auf die mittlere Schaltfläche klicken.

Erstellen Sie für den Stift Skripte, so dass er

- a) ein Quadrat zeichnet, wenn die Taste **q** gedrückt wird.
- b) einen Kreis zeichnet, wenn die Taste **k** gedrückt wird.
- c) ein Dreieck zeichnet, wenn die Taste **d** gedrückt wird.



Abbildung 1: Objekte (Sprites) in Snap! erzeugen

Neben den Bausteinen aus der Rubrik **Pen**, könnten folgende Bausteine dabei hilfreich sein.



Abbildung 2: Hilfreiche Bausteine zu Aufgabe 1

Eine Auswahl mithilfe von Buttons treffen²

Nun können wir vom Anwender nicht erwarten, dass er sich für jede Funktion eine Taste merkt. Deshalb wollen wir ihm die Arbeit erleichtern, indem wir zur Steuerung Buttons zur Verfügung stellen. Für jede Figur soll es also einen Button geben. Je nachdem welcher Button gerade aktiv ist, zeichnet der Stift die entsprechende Figur. Damit wir erkennen, welcher Button gerade aktiv ist, benötigt jeder Button zwei verschiedene Kostüme. In Abbildung 3 ist z. B. der aktive Button für das Dreieck grün, die übrigen sind grau.

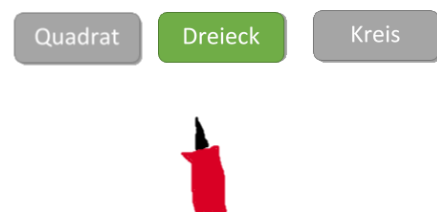


Abbildung 3: Zeichenprogramm mit Buttons

¹ Snap! wird von der University of California, Berkeley zur Verfügung gestellt: <https://snap.berkeley.edu>

² Angelehnt an eine Idee von E. Modrow (2013). Informatik mit BYOB/Snap! <http://www.emu-online.de>

Aufgabe 2: Erstellen Sie drei entsprechende Buttons als Objekte. Als Kostüme können Sie die Bilder im Ordner *Buttons* verwenden. Ziehen Sie die entsprechende Bilddatei dazu einfach in den Bereich für die Kostüme (s. Abbildung 4).



Abbildung 4: Kostüme verwalten

Jetzt müssen wir dafür sorgen, dass immer nur ein Button aktiv ist. Betrachten Sie dazu die folgenden Skripte. Zu welchem der Buttons könnten diese Skripte gehören?



Abbildung 5: Skripte für einen Button

Wir haben hier das Problem, dass auch die Buttons für das Quadrat und das Dreieck reagieren müssen, wenn der Button *Kreis* angeklickt wird. Dieses Problem können wir mithilfe von Nachrichten lösen. Um andere Objekte über ein Ereignis zu informieren, können Nachrichten ausgesandt werden. Die anderen Objekte empfangen diese Nachricht und können darauf reagieren.

Aufgabe 3: Ergänzen Sie die Skripte für die übrigen Buttons entsprechend. Wenn Sie schnell fertig sind, können Sie sich am Ende dieses Arbeitsblattes im Abschnitt *Für Fortgeschrittene* noch eine weitere Möglichkeit für die Kommunikation der Objekte untereinander anschauen.

Nun müssen wir das Stift-Objekt noch dazu bringen, die ausgewählte Figur zu zeichnen. Eine Möglichkeit wäre auch hierfür Nachrichten oder den *tell*-Baustein zu verwenden. Haben Sie schon eine Idee?

Wenn wir die Position, an der die Figur gezeichnet werden soll, jeweils mit der Maus auswählen möchten, kommen wir mit den Nachrichten bzw. dem *tell*-Baustein allein jedoch nicht weiter. Denn die Figur darf dann nicht sofort beim Anklicken des Buttons, sondern erst später als Reaktion auf

einen Mausklick gezeichnet werden. Eine Möglichkeit, wie das Skript des Stiftes passend erweitert werden könnte, erörtern wir in Aufgabe 4.

Aufgabe 4:

a) Betrachten Sie das Skript in Abbildung 6.

Was bewirken die einzelnen Bausteine und warum sind sie wichtig? Probieren Sie es aus! Untersuchen Sie dabei insbesondere

(1) was sich verändert, wenn Sie den *forever*-Baustein weglassen.

(2) welcher Wert bei



in Ihrem

Programm sinnvoll ist.

Tipp: Wenn Sie vor dem Baustein



im Auswahlbereich links den Haken setzen, wird Ihnen der aktuelle Wert, den der Baustein liefert, auf der Bühne angezeigt.

(3) Was passiert, wenn Sie den Baustein *pen up* weglassen.

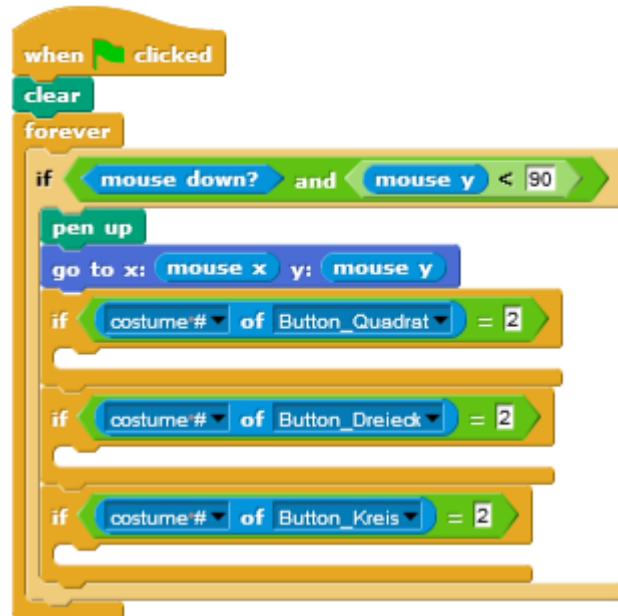


Abbildung 6: Skript des Stift-Objektes zu Aufgabe 4

b) Vervollständigen Sie das Skript so, dass die Figur gezeichnet wird, die mithilfe der Buttons ausgewählt wurde.

c) Alternativ kann die Bühne mithilfe des Bausteins in Abbildung 7 auf einen Mausklick reagieren und an den Stift je nach ausgewähltem Button eine geeignete Nachricht oder die passenden Anweisungen mithilfe des *tell*-Bausteins senden. Probieren Sie es aus!



Abbildung 7: Baustein zur Reaktion auf Mausklick

Erweiterung des Zeichenprogramms

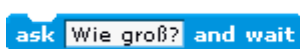
Aufgabe 5: Ergänzen Sie Ihr Zeichenprogramm um weitere Funktionen.

- Über weitere Buttons soll man z. B. die Stiftfarbe oder die Stiftdicke auswählen können.
- Wenn kein Button ausgewählt wurde, wäre vielleicht ein Hinweis an den Anwender nützlich.

Verwenden Sie dazu z. B. den Baustein:



- Mithilfe der folgenden Bausteine können auch Benutzereingaben abgefragt werden. So kann der Anwender z. B. die Größe des Quadrats vorgeben.



- Oder Sie können das freie Zeichnen mithilfe des Stiftes ermöglichen. Sie haben bereits die

Baustein-Kombination



kennengelernt. Die kann hierbei hilfreich sein.

Für Fortgeschrittene

Beim Auswählen der Buttons gibt es statt dem Versenden der Nachrichten noch eine weitere Möglichkeit, um die anderen Buttons zu deaktivieren. Mithilfe des *tell ... to ...* - Bausteins kann ein Sprite einem anderen Sprite direkt sagen, was es tun soll. Unser Skript für den Button *Kreis* sähe damit so aus:

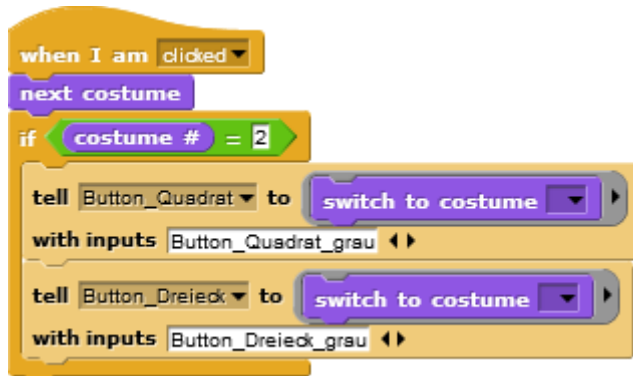


Abbildung 8: Das Skript hinter *to* wird für das angegebene Sprite ausgeführt

Das Kreis-Button-Sprite sagt also dem Quadrat-Button Sprite, dass es den Baustein



ausführen soll, und zwar mit der Eingabe *Button_Quadrat_grau*, denn so heißt das Kostüm, zu dem gewechselt werden soll. Eine entsprechende Anweisung erhält auch das Dreieck-Button-Sprite. Die Option *with inputs* erscheint, wenn man auf den zweiten schwarzen Pfeil klickt.

Aufgabe: Erstellen Sie entsprechende Skripte für Ihre drei Buttons. Legen Sie dazu ggf. eine Kopie Ihres bisherigen Programms an.

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Sie erlaubt Bearbeitungen und Weiterverteilung des Werks unter Nennung meines Namens und unter gleichen Bedingungen, jedoch keinerlei kommerzielle Nutzung.

Für die korrekte Ausführbarkeit der Quelltexte in diesem Leitfaden wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.

