

## Arbeitsteilige Projekte in Snap!<sup>1</sup>

Wenn ihr in einer Gruppe gemeinsam an einem Projekt arbeitet, kann es hilfreich sein, einzelne Objekte (Sprites) getrennt von einander zu erstellen. Anschließend kann jeder sein Objekt exportieren, um es in das Gesamtprojekt einzufügen. Abbildung 1 zeigt ein Beispiel. Die Objekte Steuerung, A und N wurden von drei verschiedenen Personen erstellt und anschließend zu einem Gesamtprojekt zusammengefügt.

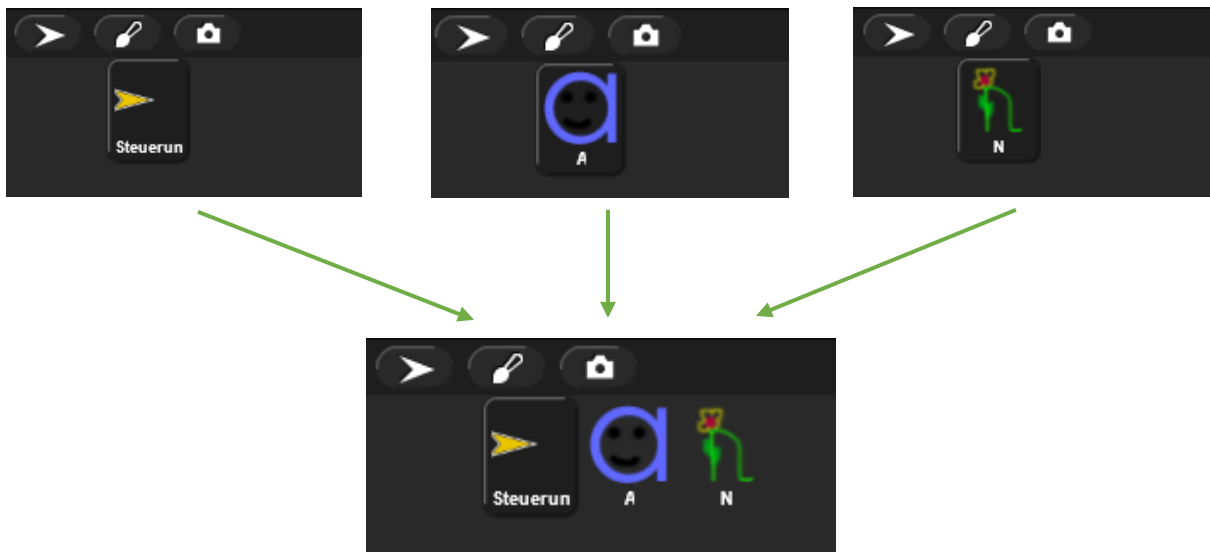


Abbildung 1: Einzelne Objekte zu einem Gesamtprojekt zusammenfügen

Wie ihr ein Objekt exportieren und in ein anderes Projekt einfügen könnt und worauf ihr achten müsst, damit keine Blöcke oder Variablen, die ihr für euer Objekt erstellt habt, verloren gehen, schauen wir uns im Folgenden genauer an.

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

## Exportieren von Objekten

Zum Exportieren wird das Objekt mit der rechten Maustaste angeklickt. Es öffnet sich ein kleines Menü. Wähle den Punkt *Exportieren ...* aus (s. Abbildung 2).

Für das Objekt wird dann eine *xml*-Datei erstellt, die den Namen des Objektes als Dateinamen erhält (s. Abbildung 3). Diese Datei speicherst du. Am besten legst du dir einen Ordner an, in dem du alle Dateien zu eurem Projekt sammelst, damit du die Dateien schnell wiederfindest.

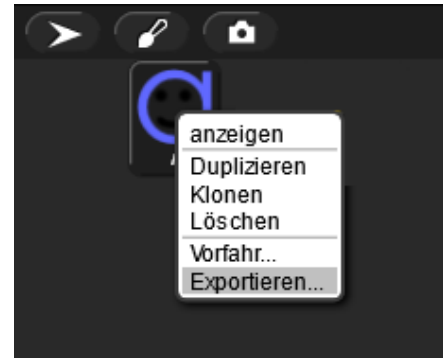


Abbildung 2: Exportieren eines Objekts

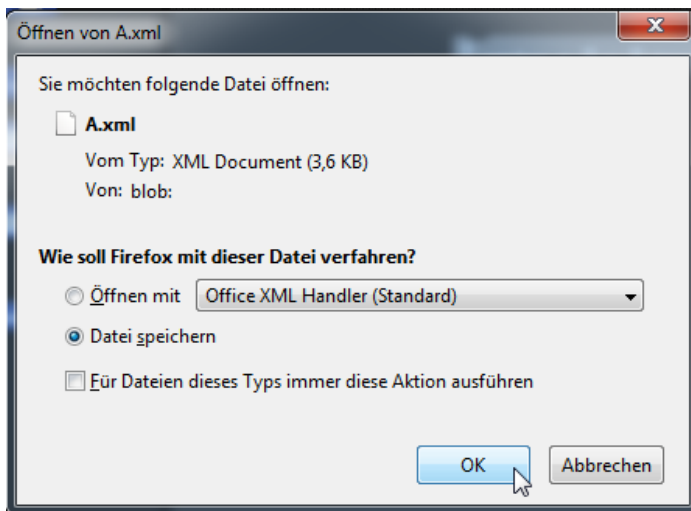


Abbildung 3: Speichern der Datei Objektname.xml

## Einfügen von Objekten

Wenn das Hauptprojekt in Snap! geöffnet ist und du dein Objekt A hinzufügen möchtest, wählst du im Menü oben links den Punkt *Import...* aus (s. Abbildung 4).



Abbildung 4: Importieren eines Objekts

Es öffnet sich der Dateibrowser. Hier öffnest du den Ordner mit deinen Projektdateien und wählst die *xml*-Datei zu deinem Objekt aus (s. Abbildung 5). Klicke anschließend unten rechts auf *Öffnen*.

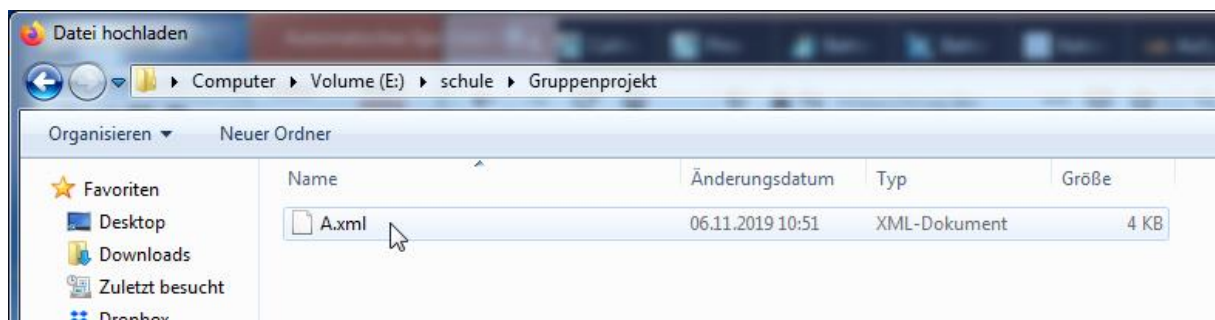


Abbildung 5: Objekt-Datei im Dateibrowser auswählen

Dein Objekt erscheint nun unten rechts in der Übersicht der Objekte zusammen mit allen anderen Objekten, die sich bereits in dem Projekt befinden. Wenn du das Objekt anklickst, erscheinen die Skripte und Kostüme, die du für das Objekt erstellt hast.

## Variablen an ein Objekt binden

Wenn wir eine neue Variable erzeugen, haben wir zwei Optionen. Wir können die Variable für alle Objekte erzeugen oder nur für das Objekt, das gerade ausgewählt ist. Die Variablen, die allen Objekten zur Verfügung stehen, nennt man auch **globale Variable**. Die Variablen, die zu einem bestimmten Objekt gehören, bezeichnet man hingegen als **lokale Variable**.



Abbildung 6: Optionen beim Erzeugen einer Variablen (deutsch und englisch)

Nur die lokalen Variablen, die mit der Option *nur für dieses Objekt* erzeugt wurden, werden beim Exportieren eines Objektes mitgenommen. Deshalb müsst ihr euch bei der Planung eures Projektes gut absprechen. Eine globale Variable, mit der alle Objekte arbeiten, erzeugt ihr mit der Option *für alle*. Diese ist dann im Hauptprogramm, in das du dein Objekt später einfügst, schon vorhanden, so dass es nicht schlimm ist, dass dein Objekt die Variable nicht mitnimmt. Wichtig ist nur, dass ihr euch auf einen einheitlichen Namen für diese Variable einigt, damit dein Objekt die Variable im Hauptprogramm unter dem gleichen Namen wiederfindet.

Eine Variable, die außer dir bzw. deinem Objekt, keiner benötigt, erzeugst du mit der Option *nur für dieses Objekt*. Diese Variable wird dann beim Exportieren und Einfügen des Objektes in ein anderes Projekt automatisch mittransportiert.


Die lokalen Variablen sind in dem Bereich für die Variablen auch nur dann sichtbar, wenn das entsprechende Objekt aktuell ausgewählt ist. Man erkennt die lokalen Variablen an dem kleinen Symbol  vor dem Variablennamen. In Abbildung 7 können wir deshalb erkennen, dass `xPosition` eine globale Variable ist, die alle Objekte verwenden können, während `zaehler` eine lokale Variable ist, die zu dem aktuell ausgewählten Objekt gehört.



Abbildung 7: Unterscheidung zwischen globalen und lokalen Variablen

## Eigene Blöcke an ein Objekt binden

Auch bei den Blöcken, die wir selbst erstellen, unterscheiden wir zwischen **lokalen** und **globalen** Blöcken. Beim Erzeugen eines eigenen Blocks stehen uns daher ebenfalls die Optionen *für alle* und *nur für dieses Objekt* zur Verfügung (s. Abbildung 8). Ein Block, den wir für alle definieren, wird für alle Objekte angezeigt und kann in jedes Skript eingebaut werden. Diese Blöcke werden aber beim Exportieren eines Objektes nicht mitgenommen. Sie müssten vorher über die Option *Export Blocks...* separat exportiert und in das neue Projekt eingefügt werden (s. Abbildung 9).

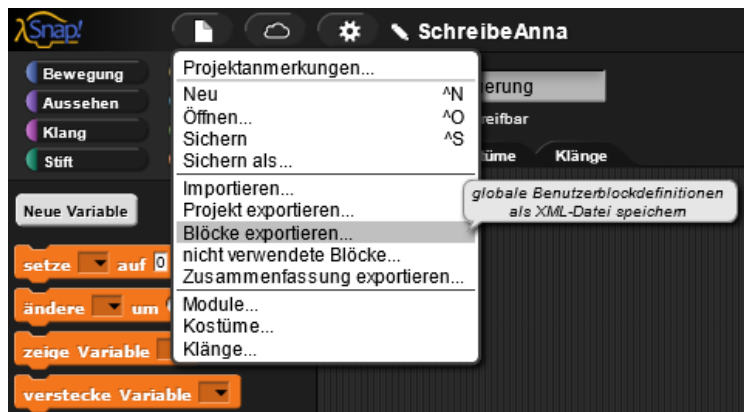


Abbildung 9: Exportieren von globalen Blöcken



Abbildung 8: Optionen beim Erzeugen eines Blocks

Wenn ein Block nur von deinem Objekt verwendet wird, ist es daher einfacher den Block mit der Option *nur für dieses Objekt* zu erzeugen. Dann wird der Block beim Exportieren des Objektes gleich mitgenommen.

## Aufruf lokaler Blöcke von außen

Wenn wir die Arbeit in einem Projekt aufteilen und die Skripte der einzelnen Objekte in lokale Blöcke verpacken, hat dies einen weiteren Vorteil. Neben der Möglichkeit über Nachrichten miteinander zu kommunizieren, kann ein Objekt mithilfe des *lasse ... tun*-Blocks (engl. *tell... to...*) einem anderen Objekt sagen, dass es seinen lokalen Block ausführen soll. Im Gegensatz zu dem Versenden von Nachrichten haben wir dabei die Möglichkeit auch Parameter zu übergeben. Schauen wir uns das an einem Beispiel an. Unser Projekt besteht aus zwei Buchstaben-Objekten A und N. Beide Objekte besitzen jeweils einen lokalen Block *zeichne*. Als Parameter erhalten die lokalen Blöcke die *x-Position*, an der gezeichnet werden soll, und *wahr* oder *falsch*, je nachdem, ob ein Großbuchstabe oder ein Kleinbuchstabe gezeichnet werden soll. Ein drittes Objekt übernimmt die Steuerung und ruft die lokalen Blöcke der Objekte A und N nacheinander mit geeigneten Parametern auf, so dass das Wort *Anna* entsteht. Abbildung 10 zeigt das Skript des Steuerprogramms.



Abbildung 11: Steuerprogramm zum Zeichnen des Namens Anna



Abbildung 10: Ausgabe des Skriptes aus Abbildung 10

Der erste *lasse ... tun* Block, lässt also Objekt A den lokalen Block *zeichne* mit den Eingaben 0 für die x-Position und *wahr* ausführen. Dementsprechend zeichnet Objekt A ein großes A an die x-Position 0. Anschließend soll Objekt N ein kleines N an Position 40 und ein weiteres an Position 80 zeichnen. Zum Schluss soll Objekt A noch ein kleines A an Position 120 zeichnen. Fertig ist der Name Anna. Das vollständige Programm findest du in der Datei *SchreibeAnna.xml*

Den blauen *... von ...* Block findest du im Bereich *Fühlen* als *Kostüm Nr. von ...* Wähle in dem Block zunächst rechts das Objekt aus, für das du einen lokalen Block aufrufen möchtest. Anschließend werden dir in der Auswahl links unter *Kostüm Nr.* die lokalen Blöcke des Objektes angezeigt.



Abbildung 12: Auswahl eines lokalen Blocks



## 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 dieser Anleitung und der beigefügten Beispiele 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.