

Gruppenprojekt: Bunte Namensschilder

Zielgruppe

Dieses Projekt eignet sich für Schülerinnen und Schüler der Sek I oder auch der Einführungsphase, die erste Kenntnisse in der Algorithmik in einer grafischen Programmiersprache erworben haben. Dieses Projekt dient der Einführung von Arbeitsweisen, die notwendig sind, um eine Implementierung arbeitsteilig zu gestalten.

Projektidée

Es soll ein Programm entstehen, das bunte Namensschilder erstellt. Der Anwender kann einen beliebigen Namen eingeben. Das Programm erstellt dann das passende Namensschild. Die Darstellung der einzelnen Buchstaben soll dabei arbeitsteilig mit der gesamten Lerngruppe erfolgen.

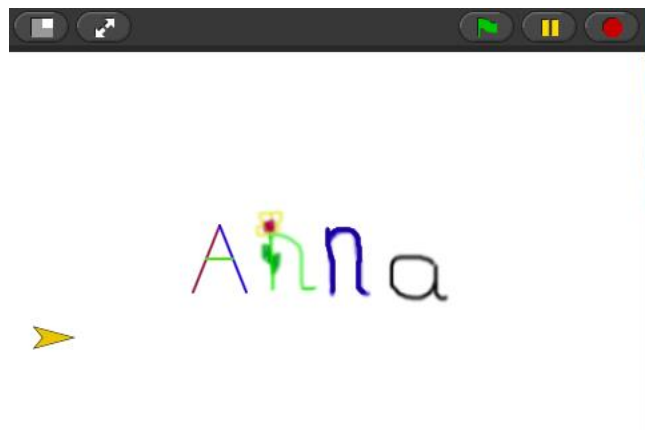


Abbildung 1: Exemplarische Ausgabe für den Namen Anna

Lernziele

Bei der Durchführung des Projektes sollten die Schülerinnen und Schüler so viel

Programmiererfahrung haben, dass Sie die algorithmischen Anforderungen zur Darstellung der Buchstaben ohne große Probleme bewältigen können. Ggf. sollten entsprechende Vereinfachungen vorgenommen werden (s. Abschnitt *Differenzierungsmöglichkeiten*). Im Vordergrund steht hier die Zusammenarbeit mit der gesamten Lerngruppe. Dies erfordert das Einhalten von Absprachen. Jeder übernimmt einen Teil der Verantwortung für das Gelingen des gesamten Projektes. Es stärkt sowohl das Kompetenzerleben jedes einzelnen Schülers und jeder einzelnen Schülerin als auch das Zusammengehörigkeitsgefühl der Gruppe, wenn jeder individuelle Beitrag das Funktionieren des Gesamtproduktes erst möglich macht. Im Vordergrund steht somit die Förderung von Kompetenzen aus den Bereichen Kommunikation und Kooperation.

Grundidee

Jeder Schüler bzw. jede Schülerin übernimmt die Implementierung eines Buchstabens. Alternativ können natürlich auch Zweierteams ein bis zwei Buchstaben übernehmen. Pro Buchstabe wird ein Objekt erstellt, das diesen Buchstaben als Groß- und Kleinbuchstaben darstellen kann. Noch abwechslungsreicher wird das Namensschild, wenn die Kleinbuchstaben mehrere Darstellungen besitzen.

Ein weiteres Objekt übernimmt die Steuerung. Nachdem der Anwender einen Namen eingegeben hat, wird einmal durch die Antwort iteriert. Für jedes Zeichen wird das entsprechende Objekt aktiviert, um an der passenden Position einen Buchstaben zu erzeugen.

Werkzeuge

Da es sich um ein Projekt für Programmieranfänger handelt, die an das Arbeiten in größeren Gruppen herangeführt werden sollen und eine grafische Ausgabe erzeugt werden soll, bietet sich die

Verwendung einer grafischen Programmiersprache an. Eine Umsetzung ist z. B. mit *Scratch 3.0*¹ oder *Snap!*² möglich. Die Besonderheiten dieser beiden Werkzeuge in Bezug auf das hier vorgestellte Projekt werden im Folgenden näher erörtert.

Kommunikation der Objekte

In Scratch 3.0 können alle Objekte die gleichen globalen Variablen verwenden, um beispielsweise die Position abzustimmen. Die globalen Variablen bleiben beim Exportieren und Importieren der Objekte erhalten. Eine Aktivierung der Objekte kann daher einfach über Nachrichten erfolgen. Hier wäre dann nach dem Importieren zu überprüfen, ob tatsächlich alle Objekte die gleichen Variablenbezeichnungen verwenden.

Auch in Snap! kann die Kommunikation des Steuerobjektes mit den Buchstabenobjekten über globale Variablen erfolgen. Dabei ist jedoch zu beachten, dass global definierte Variablen beim Exportieren nicht erhalten bleiben. Wenn es in dem Programm, in welches das Objekt importiert wird, bereits entsprechend benannte globale Variablen gibt, können diese aber ohne Änderung der Skripte genutzt werden. Für leistungsstärkere Lerngruppen bietet es sich allerdings an, hier algorithmische Konzepte der objektorientierten Programmierung einzuführen, die für arbeitsteilige Implementierungen auch langfristig tragfähig sind. In der exemplarischen Implementierung in Snap! (*namensschild_tellMitVariable.xml*) erhalten die Buchstabenobjekte daher lokale eigene Blöcke, die von dem Steuerobjekt mithilfe des *tell*-Bausteins aufgerufen werden.

Darstellung der Buchstaben

Möchte man die Buchstaben erzeugen, indem ein passendes Kostüm einen Abdruck hinterlässt, ist es wichtig, das Zentrum des Objektes bei jedem Kostüm passend einzustellen. In Snap! gibt es im Zeichenfenster für die Objekte ein eigenes Werkzeug, um das Zentrum der Objekte festzulegen. In Scratch 3.0 wird das Zentrum im Zeichenfenster durch ein kleines Kreuz markiert. Dieses ist jedoch sehr blass, so dass man genau hinschauen muss. Scratch 3.0 verfügt allerdings bereits über zwei verschiedene Kostüme für jeden Großbuchstaben.

Um die Größe der Buchstaben einheitlich zu gestalten, kann es hilfreich sein eine Bühne mit einem Referenzrahmen oder einen Referenzbuchstaben vorzugeben, an den die übrigen Buchstaben angepasst werden. Denn die Zeichenfenster von Scratch und Snap! enthalten keine Größenangaben.

Unterscheidung von Klein- und Großbuchstaben

Weiterhin unterscheiden sich Scratch 3 und Snap! bei der Erkennung von Klein- und Großbuchstaben. In beiden Werkzeugen unterscheidet der Vergleichsoperator nicht zwischen Klein- und Großbuchstaben. Der Vergleich `a = A` ergibt somit `wahr`. In Snap! kann man die Unterscheidung relativ leicht mithilfe des Unicodes vornehmen. In Scratch muss man einen etwas größeren Umweg gehen, der im Scratch Wiki beschrieben wird: https://de.scratch-wiki.info/wiki/Case_Sensing [Datum des Zugriffs: 05.11.2019].

Insgesamt erscheint die Umsetzung mit Snap! daher etwas komfortabler, insbesondere wenn man das Programm etwas umfangreicher gestalten möchte.

¹ Scratch ist ein Projekt der Scratch Foundation in Zusammenarbeit mit der Lifelong Kindergarten Group des MIT Media Lab. Es ist kostenlos unter <https://scratch.mit.edu> erhältlich.

² Snap! wird von der University of California, Berkeley zur Verfügung gestellt: <https://snap.berkeley.edu>

Notwendige Absprachen

Zunächst muss die Höhe und Breite der Buchstaben festgelegt werden. Großbuchstaben sind vermutlich höher als Kleinbuchstaben, die Breite sollte jedoch einheitlich gewählt werden, um den Algorithmus nicht unnötig kompliziert zu machen.

Die Position der Buchstaben lässt sich über eine x- und eine y-Koordinate festlegen. Wird jeder Name in einer Zeile dargestellt, ändert sich die y-Koordinate nicht, so dass nur die x-Koordinate an die Buchstabenobjekte übergeben werden muss. Es muss jedoch festgelegt werden, welchen Punkt des Buchstabens die Koordinaten angeben. Es bietet sich z. B. die linke untere Ecke an.

Je nach den in der Lerngruppe auftretenden Namen werden eventuell noch weitere Zeichen wie ein Bindestrich, Umlaute etc. benötigt.

Weiterhin ist zu klären, ob die Kommunikation der Objekte über Nachrichten und globale Variablen oder über das Aufrufen von lokalen eigenen Blöcken erfolgen soll. Bei der Kommunikation über global definierte Variable sind deren Namen festzulegen. Bei der Verwendung lokaler eigener Blöcke sollten die Parameter einheitlich gewählt werden. Wird das Steuerobjekt wie in dem Beispiel *namensschild_tellMitVariable.xml* implementiert, wären auch die Datentypen der Parameter abzusprechen. Auch ein Muster für die Bezeichnung der Objekte und Blöcke wäre in diesem Fall notwendig. Sinnvoll und hilfreich ist eine solche Absprache natürlich auch bei anderen Implementierungen.

In diesem Zusammenhang wäre dann auch zu entscheiden, wie zwischen Groß- und Kleinbuchstaben unterschieden werden soll. Ist dabei z. B. die Eingabe des Anwenders entscheidend oder wird immer der erste Buchstabe großgeschrieben. Wie ist das bei Doppelnamen? Wird immer nur der erste Buchstabe großgeschrieben, könnte das Objekt anhand der Position des Zeichens im Wort selbst entscheiden. Oder es erhält *klein* bzw. *groß* über einen Parameter mitgeteilt. Zu beachten sind dabei auch die Hinweise im Abschnitt *Werkzeuge*.

Die Absprachen sollten in jedem Fall gemeinsam mit der Lerngruppe ausgehandelt und getroffen werden, damit sie für alle transparent und nachvollziehbar sind.

Differenzierungsmöglichkeiten

Besonders kreativ können Schülerinnen und Schüler sein, wenn zur Darstellung der Buchstaben möglichst wenig Vorgaben gemacht werden. Neben der eigentlichen Darstellung kann auch der Vorgang des Zeichnens algorithmisch, kreativ ergänzt werden, z. B. durch das Abspielen von Tönen oder visuelle Effekte.

Im Idealfall wählen die Schülerinnen und Schüler selbst einen Weg, der ihnen geeignet scheint und ihrem Leistungsniveau entspricht. Erfordert die Konstellation der Lerngruppe hier steuernd einzugreifen, bieten sich zum Beispiel folgende Möglichkeiten an:

Eine Vereinfachung kann darin bestehen, einen Namen ausschließlich in Großbuchstaben zu schreiben, so dass nur eine Darstellung pro Buchstabe benötigt wird.

Besonders bunt lassen sich Buchstaben darstellen, wenn man den Buchstaben als Kostüm vorbereitet. Das kann entweder mithilfe der Zeichenfunktionen des Programmierwerkzeugs erfolgen oder in einem separaten Bildbearbeitungsprogramm. Scratch 3.0 bringt sogar zwei verschiedene Sätze an Kostümen für Großbuchstaben mit, die als Grundlage verwendet werden können. Das

Objekt muss dann lediglich das passende Kostüm anziehen und an der richtigen Stelle einen Abdruck hinterlassen.

Algorithmisch anspruchsvoller ist es, die Buchstaben mithilfe des Stiftes und dem *go to (x, y)* – Baustein oder den Bausteinen *move* und *turn* direkt im Programm zu zeichnen. Für Großbuchstaben, die aus geraden Linien bestehen, geht das relativ gut. Bei Rundungen wird es schwieriger. Von leistungsstarken Schülern könnte diese Variante z. B. für die Großbuchstaben eingefordert werden. Auch bei der Zuteilung der Buchstaben kann hier eine Differenzierung vorgenommen werden. Allerdings ist dabei zu beachten, dass es für die Schülerinnen und Schüler ggf. motivierender ist, wenn sie einen Buchstaben darstellen, der in ihrem Namen enthalten ist.

Das Steuerprogramm kann je nach Leistungsstärke und zur Verfügung stehender Zeit von der Lehrkraft zur Verfügung gestellt oder von den Schülerinnen und Schülern selbst programmiert werden. Es kann hilfreich sein, dass Steuerprogramm bereits im Vorfeld zur Verfügung zu stellen, damit die Schülerinnen und Schüler ihre Objekte darin testen können.

Wie bereits erläutert, kann auch die Kommunikation der Objekte unterschiedlich komplex gestaltet werden. Eine einfache Möglichkeit ist die Verwendung gemeinsamer globaler Variablen und das Senden einer Nachricht der Form „Schreibe A“, auf die das Objekt für Buchstabe A reagiert. Anspruchsvoller, aber langfristig tragfähiger ist die Verwendung eigener Blöcke, die an das jeweilige Buchstabenobjekt gebunden sind. Diese Blöcke erhalten die notwendigen Informationen, wie z. B. die Position, über Parameter. Die Aktivierung erfolgt in diesem Fall mithilfe des *tell*-Bausteins.

Beispiele

Die Datei *namensschild_Nachrichten.sb3* enthält ein Scratch-Programm, welches die Kommunikation zwischen Steuerobjekt und Buchstabenobjekten über Nachrichten und globale Variable löst. Um das Steuerprogramm übersichtlich zu halten, werden die Nachrichten hier mithilfe von Variablen erzeugt. Die Unterscheidung zwischen Klein- und Großbuchstaben erfolgt mithilfe der Auswahl der passenden Kostüme wie in https://de.scratch-wiki.info/wiki/Case_Sensing [Datum des Zugriffs: 05.11.2019] beschrieben.

Die Datei *namensschild_tellMitVariable.xml* enthält ein Snap!-Programm, welches die Kommunikation zwischen Steuerobjekt und Buchstabenobjekten über lokale Blöcke und den *tell*-Baustein löst. Die Auswahl der Objekte im *tell*-Baustein erfolgt über Variablen, so dass das Steuerprogramm übersichtlich bleibt und keine Fallunterscheidung für alle 26 Buchstaben notwendig ist. Diese Art der Implementierung ist hilfreich, wenn das Steuerprogramm von der Lehrkraft gleich zu Beginn zur Verfügung gestellt wird. Für Schülerinnen und Schüler in der Sek I wird diese Variante aber in der Regel zu komplex sein, um sie selbständig zu implementieren. In der Anleitung für die Schülerinnen und Schüler wird der *tell*-Baustein daher ohne die Verwendung von Variablen vorgestellt (s. Schülermaterial)

Die Beispiele enthalten nur exemplarisch die Buchstaben A und N.

Schülermaterial

Scratch

Für die Umsetzung eines Gruppenprojektes mit Scratch kann die Anleitung *Gruppenprojekte_Scratch* verwendet werden. In dieser Anleitung wird das Vorgehen zum Exportieren und Importieren von Figuren beschrieben. Außerdem wird die Unterscheidung von globalen und lokalen Variablen in

diesem Zusammenhang thematisiert. Ob dieser Abschnitt benötigt wird, muss je nach Projekt und Lerngruppe entschieden werden. Die Kommunikation zwischen Objekten mithilfe von Nachrichten wird als bekannt vorausgesetzt.

Snap!

Für die Umsetzung eines Gruppenprojektes mit Snap! kann die Anleitung *Gruppenprojekte_Snap* verwendet werden. Das grundsätzliche Vorgehen zum Erstellen eines Projektes in Snap! und die Kommunikation zwischen Objekten mithilfe von Nachrichten wird dabei als bekannt vorausgesetzt. Die für das Exportieren und Importieren benötigten Konzepte werden weitestgehend unabhängig von dem konkreten hier vorgestellten Projekt erläutert, so dass die Anleitung auch im Zusammenhang mit anderen Gruppenprojekten verwendet werden kann. Neben dem Vorgehen zum Exportieren und Importieren wird auch die Unterscheidung zwischen globalen und lokalen Variablen sowie globalen und lokalen Blöcken thematisiert. Bis zu welchem Abschnitt die vorgestellten Konzepte benötigt werden, muss vor dem Hintergrund der jeweiligen Lerngruppe entschieden werden. Die Verwendung von Variablen im Zusammenhang mit dem *tell*-Baustein wird nicht thematisiert, da dies für die Sek I relativ komplex ist. Die Schülerinnen und Schüler werden sich an die Verwendung dieses Bausteins daher eher durch die explizite Implementierung eines Namens wie in dem Beispielprogramm *SchreibeAnna.xml* oder eine explizite Fallunterscheidung der gefunden Zeichen in der Eingabe herantasten.

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#). Sie erlaubt Download und Weiterverteilung des vollständigen Werkes unter Nennung meines Namens, jedoch keinerlei Bearbeitung oder kommerzielle Nutzung.

Für die korrekte Ausführbarkeit 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.