Punkt vor Strich

In diesem Abschnitt soll es darum gehen, eine Zeichnung als Bewegung umzusetzen, also nicht als das Bild bzw. Striche selbst, sondern als eine Abfolge von Punkten, die ein Malstift durchläuft, um etwas zu zeichnen. Eine solche Abfolge von Punkten nennt man “Pfad[[1]](#footnote-1)”.

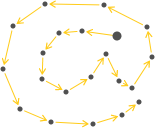
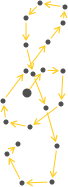
Yellow text on a black background

Description automatically generated with low confidence

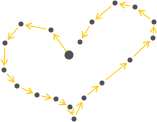
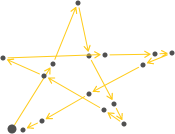
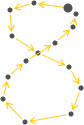
Beispiel für einen “Zickzack”-Pfad, der ein Z-förmiges Zeichen malt.

Weitere Beispiele für Pfade, die verschiedene Zeichen darstellen:

A picture containing watch

Description automatically generated  

die Zahl 3 das E-Mail-Zeichen @ Ein Sopran-Notenschlüssel



Die Zahl 8 ein Stern ein Herz

Allen diese Zeichen ist gemeinsam, dass du sie mit deinem Programm in einem einzigen Zug zeichnen kannst, also ohne den Stift abzusetzen.

# Einen Pfad aufzeichnen

Um einen Pfad zu speichern, benötigst du eine Variable. Am besten gibst du ihr einen aussagekräftigen Namen, also z.B. “Zeichen”:

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

In dieser neuen Variablen soll nun jedes Mal, wenn dein Programm ein Bild malt, die Abfolge der durchlaufenen Punkte aufgezeichnet werden. Dafür benötigst du die folgenden Blöcke bzw. Anweisungen:

Graphical user interface, application

Description automatically generated

Die erste Anweisung initialisiert die Variable als Liste. “Initialisieren” heißt hier, dass die Variable zu einer neuen, leeren Liste wird. Das soll immer dann passieren, wenn du ein neues Zeichen malst, also am Anfang deines Zeichenprogramms.

Eine Liste speichert Daten der Reihe nach. Elemente einer Liste können Daten aller Art sein, also Zahlen, Text, Figuren, Kostüme, Klänge, Blöcke und natürlich auch andere Listen.

Wie also kannst du eine Position, einen geometrischen Punkt, in einer Liste speichern? Die Position deines Objekts und damit des Malstiftes hat zwei Werte, eine x-Position und eine y-Position. Du kannst also einen Punkt als Liste mit zwei Werten beschreiben. Genau das macht die zweite Anweisung.

Füge die beiden Anweisungen in dein Malprogramm ein, damit die Liste

* am Anfang einer neuen Geste auf eine leere Liste gesetzt wird.
* die aktuelle x- und y-Position nach jedem Positionswechsel zur Liste hinzugefügt wird.

# Einen Pfad abspielen

Graphical user interface, website

Description automatically generatedWas man aufgezeichnet hat, kann man natürlich auch wieder abspielen. Anstatt zum “Mauszeiger” zu gehen, wie in diesem Block,

kann eine Figur auch an einem bestimmten Punkt gehen, der eine x-Position und eine y-Position hat:

Graphical user interface, text, application, chat or text message

Description automatically generated

Diese Blöcke machen beide dasselbe, sie bewegen die Figur zur Koordinate [7, 12].

Um eine Liste der Reihe nach von vorn bis hinten “abzuspielen”, kannst du Schleifenblöcke verwenden. Einer davon ist dafür ganz besonders praktisch, du findest ihn bei den anderen Listenblöcken:

Graphical user interface

Description automatically generated with low confidence

Dieser “für jedes Element von \_” Block nimmt der Reihe nach jedes Element aus einer Liste und führt damit diejenigen Blöcke aus, die eingerückt in seinem C-förmigen Bereich stecken. Der Block hat eine eingebaute Variable namens “Element”. Von dieser Variablen kannst du ganz einfach Kopien “abziehen” und in anderen Blöcken innerhalb der Schleife verwenden.

A picture containing text, sign, green, vector graphics

Description automatically generatedDamit kannst du jetzt ein kleines Wiedergabeprogramm bauen. Diese drei Blöcke helfen dir dabei:

Teste dein Programm! Male ein Zeichen, danach klicke auf das Abspielprogramm. Wenn du alles richtig gemacht hast, dann gibt das Programm dein Zeichen exakt so wieder, wie du es zuvor gemalt hast.

1. Wusstest du, dass Vektorgrafiken auch als Pfade von Punkten (Vektoren) gespeichert werden? Das Tolle daran ist, dass man die Grafik beliebig vergrößern kann, ohne dass die Auflösung schlechter wird. [↑](#footnote-ref-1)