

TurtleStitch Projekttag – Lerninhalte

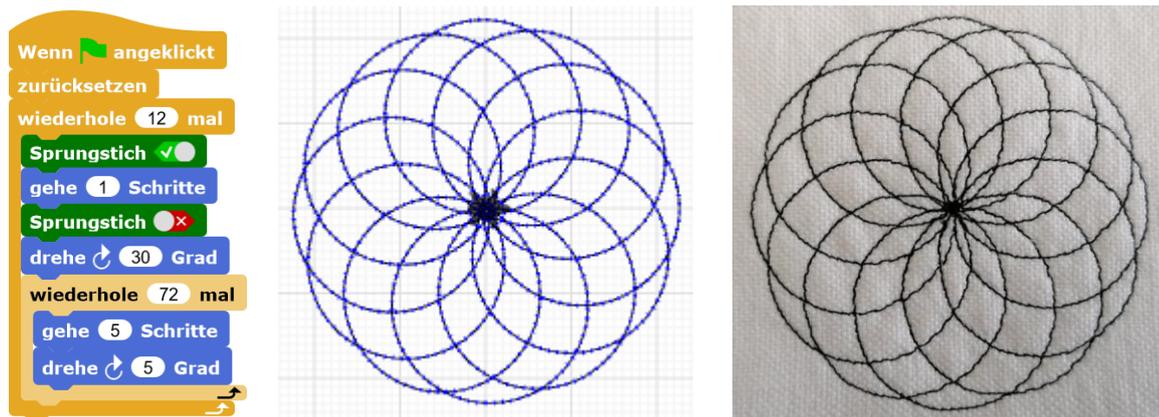
1 Einleitung

Diese Anleitung soll Schülerinnen und Schülern wie auch allen Lernkräften einen möglichst einfachen Einstieg bieten in TurtleStitch.



Bei TurtleStitch handelt es sich um eine Internetseite, auf der blockbasiert programmiert werden kann. Ihr müsst euch also nicht mit kompliziertem Code herumschlagen, sondern könnt direkt in den spannigen Teil des Programmierens übergehen!

Mit TurtleStitch könnt ihr allerlei Formen und Figuren entwerfen. Zum Beispiel kann man mit diesem Code (links) das entsprechende Muster auf dem Bildschirm entwerfen (Mitte). Mit einer Stickmaschine kann dieses Muster dann physisch auf Stoff genäht werden (rechts).



Bevor ihr aber zu sticken beginnt, stellt bitte sicher, dass ihr erst die Einleitung bezüglich der Stickmaschine gelesen und verstanden habt. Wenn ihr das auf später verschieben möchtet und einfach mal mit programmieren loslegen wollt, dürft ihr hier weitermachen und wir wünschen euch natürlich ganz viel Spass dabei 😊

2 Einführung in TurtleStitch

2.1 Programmstart

Ruft die Internetseite <https://www.turtlestitch.org/run> in eurem Browser auf. TurtleStitch kann direkt aus dem Browser bedient werden, ihr könnt auch eure Stickereien speichern und vieles mehr. Dies alles funktioniert, ohne dass ein Programm installiert werden muss.

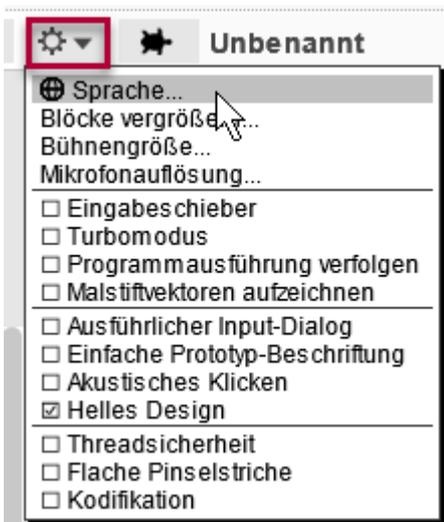
2.2 Sprache

Im oberen Bereich von TurtleStitch seht ihr einen Balken mit allerlei Einstellungen.



Wir gehen hier zu gegebener Zeit noch darauf ein. Da ihr sicher schon darauf wartet, endlich zu programmieren, besprechen wir hier die wichtigsten Einstellungen.

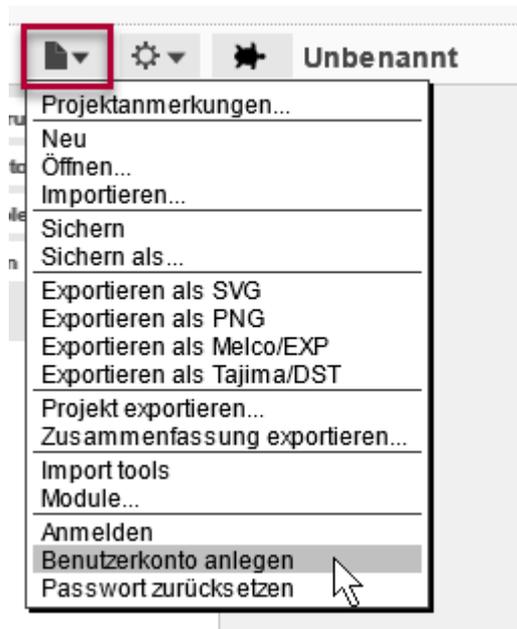
Als erstes solltet ihr die Sprache von TurtleStitch festlegen. Da unsere Beschreibungen und Beispiele in Deutsch formuliert sind, würden wir empfehlen, dass ihr ebenfalls Deutsch verwendet. Klickt dazu auf das Zahnradchen und wählt «Sprache».



Aus der Liste, die sich öffnet, wählt «Deutsch» aus. Die Oberfläche ist nun in Deutsch eingestellt und bleibt auch so, solange ihr denselben Browser weiterverwendet.

2.3 Account Anlegen

Um eure Arbeiten später speichern und wieder aufrufen zu können, solltet ihr euch ein Benutzerkonto anlegen. Klickt dazu auf das Dokument-Symbol (siehe folgender Screenshot) und wählt dann den Menüpunkt «Benutzerkonto anlegen». Wählt euch danach einen Benutzernamen aus und gebt eure E-Mail und ein Passwort an.



Die «Terms of Service» und «Privacy Policy» müsst ihr akzeptieren, um ein Konto anlegen zu können. Den Newsletter müsst ihr auch abonnieren, dieser wird aber nur selten versendet.

Ihr habt nun eine Bestätigungsmail bekommen, mit dem ihr das TurtleStitch Konto bestätigen und aktivieren könnt. Klickt dazu auf den Link, welcher in der Mail gesendet wurde.

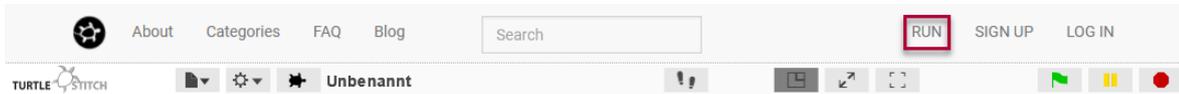
Das Benutzerkonto wurde nun aktiviert und ihr könnt euch einloggen.

User account activated.

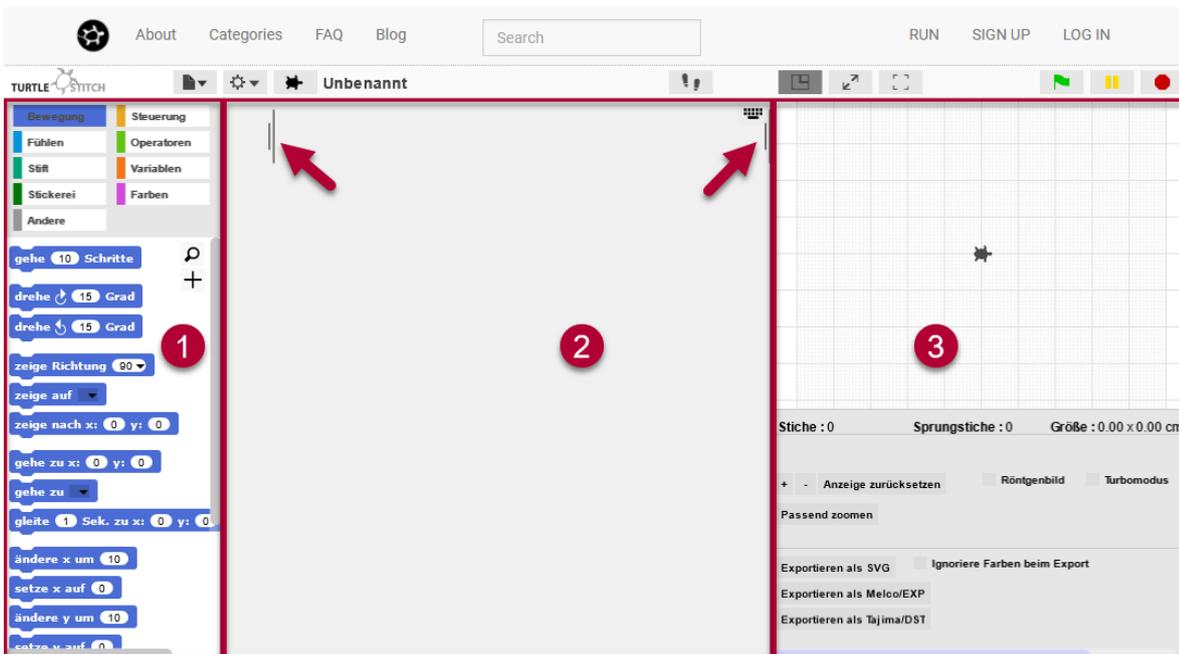
You can now [log in.](#)

2.4 Übersicht

Wenn ihr auf «RUN» klickt, gelangt ihr wieder zurück zum Hauptbildschirm:



Der Hauptbildschirm wird in TurtleStitch am meisten verwendet, wie der Name schon sagt. Die Ansicht ist in verschiedene Bereiche unterteilt:



- 1) Auf der linken Seite seht ihr die Palette mit den Codeblöcken. Von hier aus werdet ihr die Codeblöcke auswählen.
- 2) In der Mitte befindet sich der Programmierbereich. Um zu programmieren, zieht ihr die gewünschten Blöcke aus der Palette hierhin in den Programmierbereich.
- 3) Im oberen rechten Bereich ist die Bühne und ihr erkennt das Wappentier von TurtleStitch, die Schildkröte. Diese wird genau das tun, was ihr mit dem Code im Programmierbereich befohlen wird. Wir werden gleich ein Beispiel dazu machen. Unter der Bühne seht ihr noch einige Knöpfe und Funktionen, hier könnt ihr später zum Beispiel euer Programm auf einen USB-Stick exportieren und vieles mehr. Wir erklären diesen Teil zu gegebener Zeit.

Ihr könnt die Größe der Bereiche anpassen, indem ihr die Symbole mit den senkrechten Strichen klickt und zieht. Diese sind im Bild oben mit den Pfeilen markiert.

2.5 Codeblöcke

Aus der Palette können wir unsere Codeblöcke auswählen. Die Codeblöcke sind in verschiedene Kategorien unterteilt, die wir kennen sollten. Zur Übersicht sind die Kategorien eingefärbt.

Bewegung	Steuerung
Fühlen	Operatoren
Stift	Variablen
Stickerei	Farben
Andere	

Die Namen der Kategorien sind eigentlich selbsterklärend und zudem lernt ihr die Blöcke am besten während deren Anwendung kennen. Hier darum nur ein paar Beispiele:

- Bewegung: Wenn ihr die Kategorie aufruft und z.B. direkt auf diesen Block klickt, bewegt sich die Schildkröte bereits. Ihr könnt auch die weiteren Blöcke per Anklicken testen.

gehe 10 Schritte

- Fühlen: Eure Schildkröte kann auch Dinge ertasten oder abfragen. Beispielsweise fragt sie bei diesem Block, ob die Leertaste gedrückt ist. Diese Blöcke werden wir eher später benutzen.

Taste Leertaste gedrückt?

- Stift: Diese Blöcke werden vor allem zum Malen verwendet, also ob die Schildkröte beim Bewegen eine Linie zieht oder nicht. Stift hoch würde zum Beispiel heißen, dass im Anschluss keine Linie gemalt wird.

Stift hoch

- Stickerei: Da wir später Sticken wollen, ist dies für uns eine besonders wichtige Kategorie. Wenn wir z.B. einen Laufstich wählen, werden alle folgenden Bewegungen mit diesem Stich ausgeführt. Also nicht erschrecken, wenn der Laufstich noch nicht sofort etwas tut. Die Blöcke von «Stickerei» sind keine Bewegungen, sondern Einstellungen, wie die nächsten Bewegungen gemacht werden sollen.

Laufstich in 10 er Schritten

- Andere: Hier befinden sich spezielle Kommandos, momentan ist nur «passend zoomen» enthalten, welches eure Bühne auf den bereits gestickten Bereich zoomt.

passend zoomen

- Steuerung: Hier befinden sich eine Menge spannender Blöcke, die ihr häufig brauchen werdet. Als erstes werdet ihr sicher «Wenn 'Fahne' angeklickt» verwenden, der in TurtleStitch als Startkommando verwendet wird, um das Sticken zu beginnen. Auch weitere Blöcke die Entscheidungen treffen oder Kommandos wiederholen werden wir zu gegebener Zeit oft einsetzen.

Wenn angeklickt

- Operatoren: Die Operatoren nehmen euch die Rechenarbeit ab. Ihr könnt z.B. einfache Additionen ausführen und vieles mehr.



- Variablen: Diese Blöcke machen euch das Leben in TurtleStitch leichter. Ihr könnt zum Beispiel mit einer Variablen «Zähler» mitverfolgen, wie oft ein bestimmtes Ereignis eintritt.



- Farben: Hier könnt ihr zum Beispiel definieren, welche Farben eure Schildkröte auf dem Bildschirm benutzen soll.



3 Mein erstes TurtleStitch Programm

So, wir machen nun unser erstes TurtleStitch Programm zusammen 😊

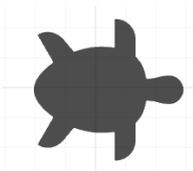
3.1 Vorbereiten des Hauptbildschirmes

Loggt euch immer erst mit eurem Konto ein, bevor ihr startet! Dies ist wichtig, damit ihr eure Programme später speichern könnt.

Klickt also erst auf «Log In» und gebt eure Kontodaten an falls ihr noch nicht angemeldet seid. Danach klickt auf «Run», um auf den Hauptbildschirm zurückzukehren.



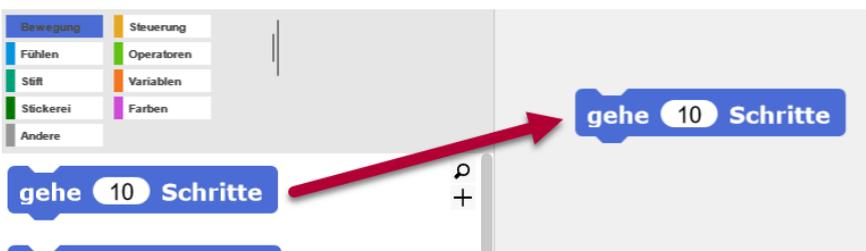
Die Schildkröte auf der Bühne schaut nach rechts und der Programmierbereich ist leer:



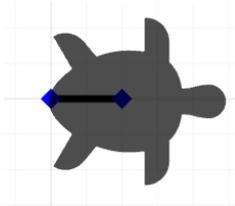
Nun sind wir startklar!

3.2 Bewegen

Wählt nun links aus der Palette in der dunkelblauen Kategorie «Bewegung» den Block «gehe 10 Schritte» und zieht diesen in die Mitte in den Programmierbereich hinein.



Ein Schritt ist dabei 0.2 mm, 10 Schritte sind also 2 mm. Wenn ihr den Block nun anklickt, bewegt sich die Schildkröte 2 mm geradeaus. **Denkt immer daran, dass die Schildkröte aus ihrer eigenen Sicht handelt.** 2 mm geradeaus sind also auf der Bühne 2 mm nach rechts für die Schildkröte, weil sie gerade dahin schaut.



Wenn ihr den Block nun mehrmals anklickt, bewegt sich die Schildkröte jedes Mal 2 mm geradeaus und macht alle 2 mm einen Stich. Irgendwann verlässt die Schildkröte die Bühne, sticht aber fröhlich weiter. Ihr erkennt die Grenze der Bühne daran, dass die Fläche innerhalb der Bühne kariert ist und die Fläche ausserhalb der Bühne weiss:



Ihr seht unterhalb der Bühne nun wichtige Informationen.

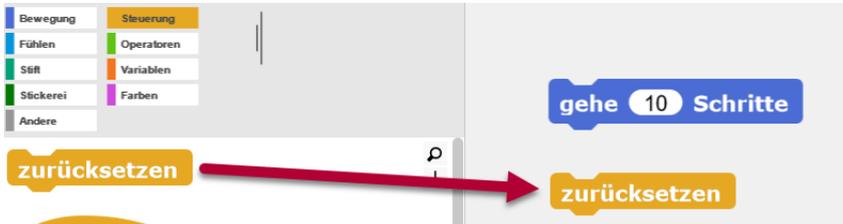
- Stiche: Die Anzahl Stiche die ein Muster benötigt gibt später Hinweise auf die Dauer, die man für das Sticken des Musters einrechnen muss.
- Grösse: Die Grösse ist sehr wichtig und kann darauf hindeuten, dass das Muster allenfalls zu gross ist für den Stickrahmen und verkleinert werden muss. Generell sollte noch gestickt werden können, was noch auf die karierte Bühne passt.

3.3 Zurücksetzen der Bühne

Wenn ihr ein Muster entwickelt, werdet ihr euren Code immer wieder von vorne ablaufen lassen. Dazu werdet ihr auch das bereits gezeichnete Muster entfernen, um von vorne beginnen zu können.

Zieht dazu den Block «zurücksetzen» aus der orangen Kategorie «Steuerung» in den Programmierbereich. Wenn ihr darauf klickt, werden die Stiche auf der Bühne gelöscht und die Schildkröte wieder auf die Startposition in der Mitte der Bühne gesetzt.

Wichtig dabei ist, dass der Programmierbereich mit «zurücksetzen» nicht geändert wird. Euer Code bleibt also erhalten:



3.4 Alles zusammen

Wie erwähnt ist der Startpunkt «Wenn 'Fahne' angeklickt» nützlich, um den Start eines Programms zu markieren. Zieht auch diesen aus der orangen Kategorie und setzt die bisherigen Blöcke darunter, also so:



Wenn ihr nun auf die grüne Fahne klickt, wird erst die Bühne zurückgesetzt und die Schildkröte geht 10 Schritte. Ändert die Anzahl Schritte von 10 auf z.B. 20, klickt nochmal auf die grüne Fahne und beobachtet was passiert. Probiert auch noch andere Zahlen aus.

Die Schildkröte fängt immer wieder von vorne an und macht das, was ihr der Schildkröte in eurem Programm aufgetragen habt.

Es ist übrigens normal, dass man ein Programm immer wieder laufen lässt. So kann man überprüfen, ob Änderungen oder Erweiterung auch tatsächlich den erwünschten Effekt erbringen. Auch Profiprogrammierer machen das dauernd so.

3.5 Wiederholschleife

Damit die Stichlänge nicht zu gross wird und doch eine etwas grössere Linie gestickt wird, könnt ihr eine Wiederholschleife verwenden. Verschachtelt den «gehe 10 Schritte» Block in der «wiederhole 12 mal» Schleife. Teste auch dies mit unterschiedlicher Anzahl Iterationen (Wiederholungen) und unterschiedlich grossen Schritten.



3.6 Stickeinstellungen festlegen

Es empfiehlt sich, gleich am Anfang eines Programmes die Stickeinstellungen festzulegen, also Stichtart und Stichtlänge. Dies verhindert, dass wir bei grossen Schritten zu lange Stiche machen, die dann als lose Fäden irgendwo festhängen können.

Unterhalb der Bühne würde eine Warnung angezeigt, wenn Stiche zu lange werden und somit nicht mehr straff gestickt werden können:

```
Stiche : 154
2 are too long! (will get clamped)
```

Hier in unserem Beispiel verwenden wir einen Laufstich, der alle 10 Schritte (2 mm) einen Stich macht, selbst wenn wir 100 Schritte auf einmal springen. 10 Schritte ist übrigens die unverbindliche Empfehlung für die Stichabstände beim Sticken.

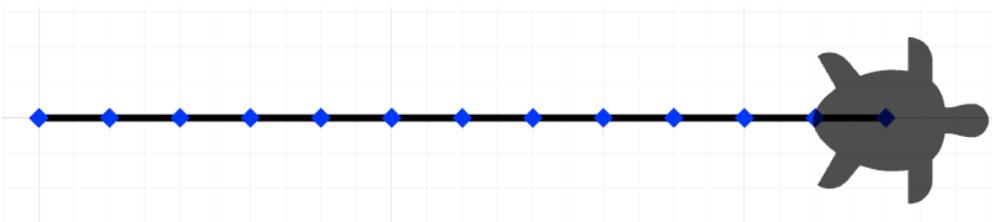
Stickeinstellungen zurücksetzen
Laufstich in 10 er Schritten

Weitere Stiche sind in Kapitel 4.1 «Stichtypen» beschrieben.

Der Block «Stickeinstellungen zurücksetzen» lohnt sich vor allem am Anfang eines Programmes, um alle Einstellungen aus der letzten Ausführung zu entfernen, die vielleicht am Anfang des Programmes stören könnten.

3.7 Eine Linie sticken

Repetieren und verbinden wir nun alles, was wir bisher gelernt haben. Wir werden dazu eine 24 mm lange Linie sticken. Befolgt hierfür die untenstehenden Schritte und erstellt eure eigene Kopie des Codeblocks.



Wenn angeklickt
zurücksetzen
Stickeinstellungen zurücksetzen
Laufstich in 10 er Schritten
wiederhole **12** mal
gehe **10** Schritte

Der Block «zurücksetzen» setzt alle Einstellungen zurück. Die Schildkröte wird auf den Ursprung (0, 0) gesetzt und schaut nach rechts, die Bühne wird geleert. Mit «Stickeinstellungen zurücksetzen» werden alle vorher getätigten Einstellungen aus der Stickerei-Kategorie zurückgesetzt.

Mit «Laufstich in _ er Schritten» wählt man die Stichtart und Stichtlänge: Ein Laufstich ist ein einfacher gerader Stich, ein Schritt entspricht dabei 0.2 mm Länge (10 Schritte entsprechen 2 mm, 20 Schritte entsprechen 4 mm usw.).

Der Block «wiederhole _ mal» sorgt dafür, dass die umschlossene Blocksequenz so oft wiederholt wird, wie durch die eingetragene Zahl angegeben.

Mit «gehe _ Schritte» bestimmt man schliesslich, wie lange die zu stickende Linie insgesamt wird. Auch hier entspricht ein Schritt 0.2 mm, sowohl am Anfang als auch Ende wird ein Stich gesetzt (markiert durch die blauen Punkte).

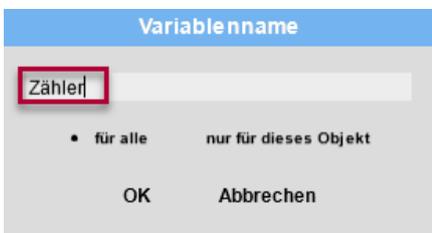
Experimentiert ein wenig herum!

3.8 Variablen

Ihr könnt das obige Beispiel noch vereinfachen mit einer Variablen. Drückt dazu in der Kategorie «Variablen» in der Palette ganz oben den grauen Knopf «Neue Variable»:



Gebt der Variablen einen Namen und drückt «OK»:



Wir können nun unser Programm ändern. Die Variable «Zähler» ziehen wir von der Palette in die Schleife «wiederhole 12 mal» hinein. Wir überdecken also die Zahl 12 mit unserer Variablen und nun zeigt die Schleife «wiederhole 'Zähler' mal» an.

Wir müssen dem Zähler noch einen Wert geben mit «setze 'Zähler' auf _ ». Setzen wir diesen auf 12, passiert genau das gleiche wie zuvor. Wenn wir eine Variable so festlegen, nennt sich dies übrigens «initialisieren».



Der Vorteil ist nun, dass wir nur an einer Stelle einen Wert für den Zähler festlegen müssen, aber wir können diesen immer wieder verwenden.

Z.B. können wir aus der Linie ein Quadrat machen. Dabei müssen wir die Kantenlänge nur einmal angeben, und das wird für alle vier Kanten gleichmässig übernommen. Wenn uns das Quadrat zu klein ist, ändern wir «setze 'Zähler' auf _» an der einen Stelle für alle vier Kanten auf einmal:

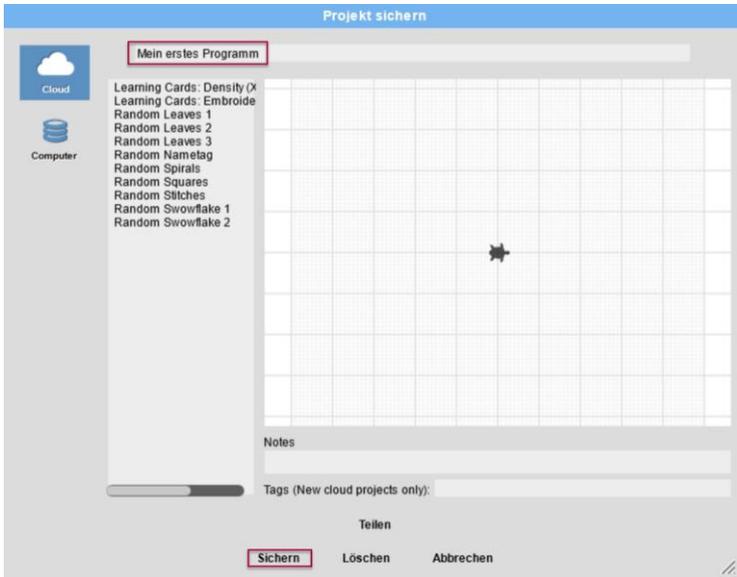


3.9 Speichern

Um speichern zu können müsst ihr eingeloggt sein. Klickt dazu auf das Dokument-Symbol und wählt dann den Menüpunkt «Sichern als»:



Gebt dem Programm anschliessend einen passenden Namen und klickt auf «Sichern»:



Wenn ihr weitere Änderungen an eurem Programm macht, könnt ihr das Dokument-Symbol anklicken und «Sichern» wählen, dann ist euer Fortschritt direkt gespeichert.



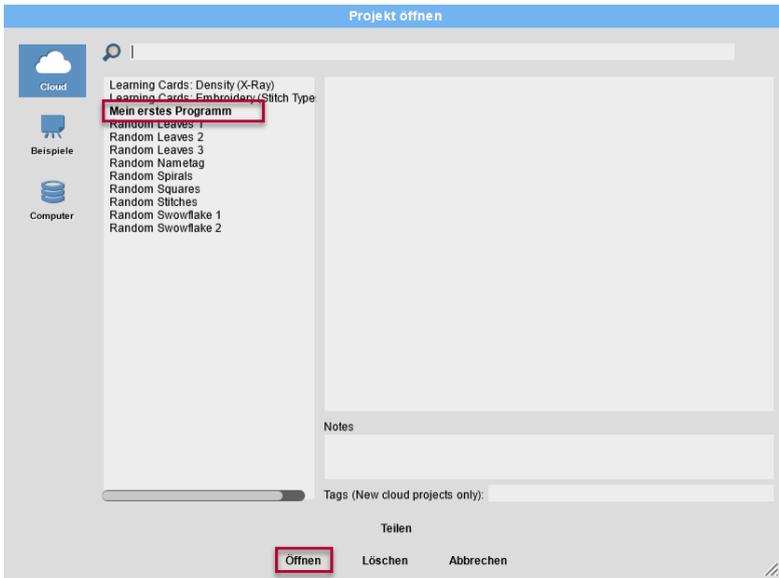
Achtung: Speichern heisst nur, dass ihr das Programm später in TurtleStitch wieder öffnen und weiterbearbeiten könnt. Um das entstandene Muster auf einer Stickmaschine zu sticken, seht auch das Kapitel 7 «Stickern mit der Stickmaschine» an.

3.10 Programme öffnen

Klickt nochmal auf das Dokument-Symbol und wählt diesmal «Öffnen»:



Wählt euer Programm aus und klickt auf «Öffnen» am unteren Bildrand:



3.11 Programme teilen

Euer Programm wird automatisch mit der Community geteilt, wenn ihr dieses zum ersten Mal speichert. Dadurch sind die Programme für alle Personen mit einem TurtleStitch Account sichtbar.

Die geteilten Programme sind in der Liste unter «Öffnen» in fetter Schrift dargestellt. Wenn ihr dies nicht möchtet, klickt darauf und wählt unten «Nicht mehr teilen». Umgekehrt könnt ihr Programme in normaler Schrift anklicken und «Teilen» wählen, um diese für alle sichtbar zu machen.

4 Weitere Tipps und Vereinfachungen

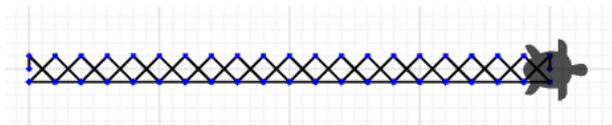
Hier erhält ihr weitere Tipps zu euren Stickmustern und wie ihr euren Code einfach und übersichtlich haltet.

4.1 Stichtypen

Wie in Kapitel 3.6 «Stickeinstellungen festlegen» beschrieben, setzt ihr die Stichtypen fest bevor ihr die Schildkröte bewegt. Das Festlegen der Stickeinstellung erzielt noch kein sichtbares Resultat. Erst wenn ihr die Schildkröte tatsächlich bewegt, wird der Stich angewendet.

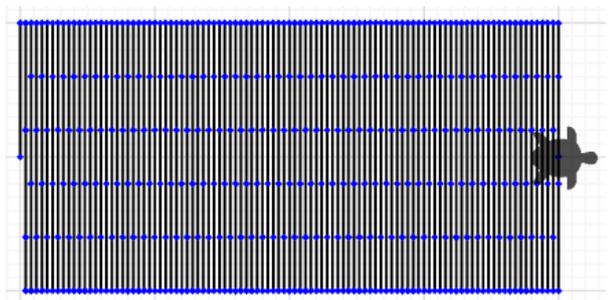
Die Stiche können sehr unterschiedlich sein. Hier ein Anschauungsbeispiel, wenn der Kreuzstich angewendet wird:

```
zurücksetzen
Stickeinstellungen zurücksetzen
Kreuzstich 10 mal 10 zentriert ✓
gehe 200 Schritte
```



Und hier ein Beispiel eines Tatami-Stiches:

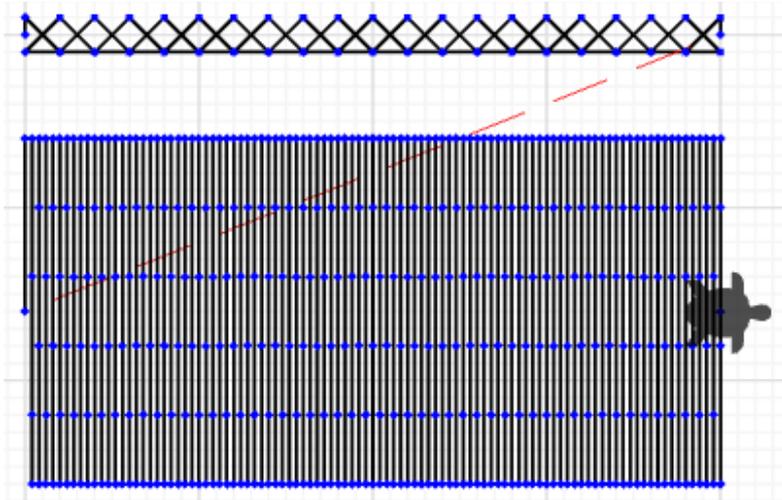
```
zurücksetzen
Stickeinstellungen zurücksetzen
Tatami-Stich mit Breite 100 Intervall 40 Versatz true zentriert ✗
gehe 200 Schritte
```



Sieht großartig aus, nicht wahr? Probiert die verschiedenen Stiche selbst aus und spielt auch mit den Einstellungen der einzelnen Stiche.

4.2 Sprungstich

Einen speziellen Stich schauen wir uns noch genauer an. Angenommen wir wollen beiden obigen Stiche untereinander sticken als Vergleich. Das sieht dann so aus:



Normalerweise stickt unsere Schildkröte immer etwas, wenn sie sich bewegt. Hier sehen wir aber eine rote Linie, die bedeutet, dass hier nicht gestickt wurde. Unsere Schildkröte macht dort also einen Sprung, darum Sprungstich. Der Faden wird beim Absprung verknotet und ebenfalls beim Landen, und der lose rote Teil dazwischen kann später abgeschnitten werden.

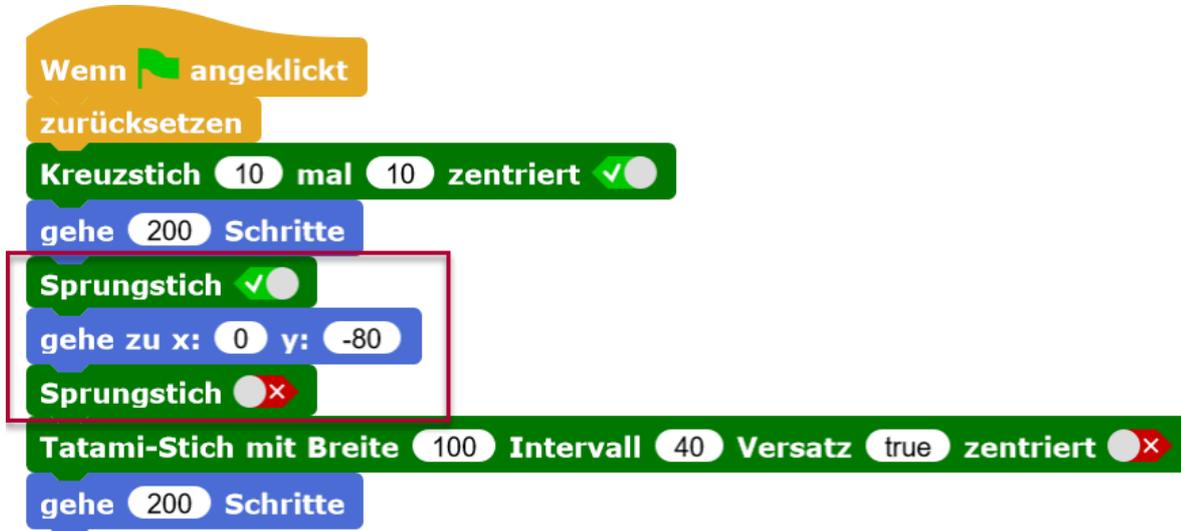
Die Stelle, wo der Absprung ist, wird mit dem Block «Sprungstich» programmiert. Der grüne Schalter bedeutet mit anderen Worten «Absprung = ja»:



Nachdem wir unsere Schildkröte zu der Landestelle bewegt haben, wird die Landung mit einem weiteren Block «Sprungstich» markiert. Dieses Mal stellen wir den Schalter um auf Rot, anders ausgedrückt ist dies «Absprung = nein» und bedeutet damit «Landung»:

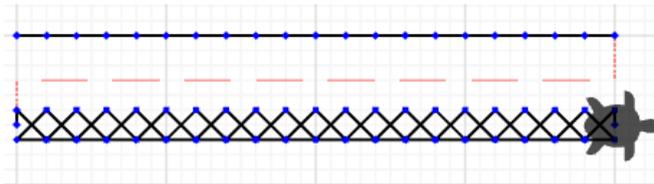


Wir müssen immer darauf achten, dass ein Absprung und eine Landung zusammenpassen. Hier ist das ganze Beispiel mit Absprung, Bewegung dazwischen und Landung eingerahmt:

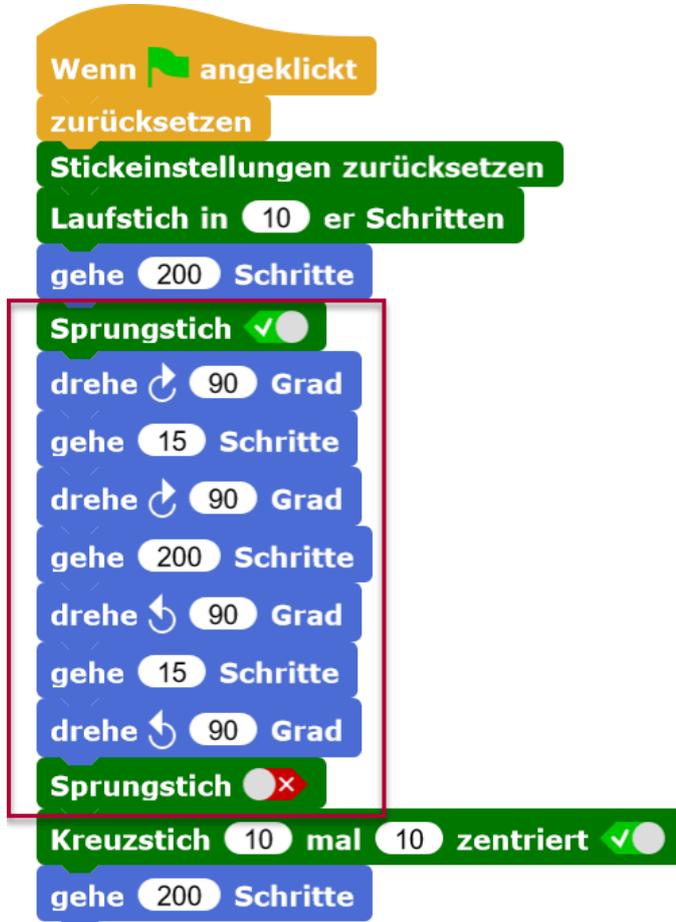


4.3 Eigene Blöcke erstellen

Stellt euch vor, wir wollen wie vorhin weitere Stichtypen vergleichen. Wir springen jeweils wieder an die gleiche Stelle auf der x-Achse, einfach etwas weiter runter.

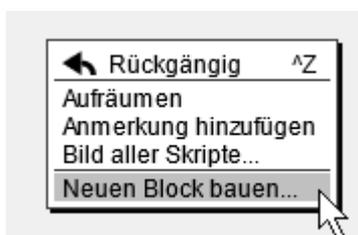


Der Code dazu könnte folgendermassen aussehen:

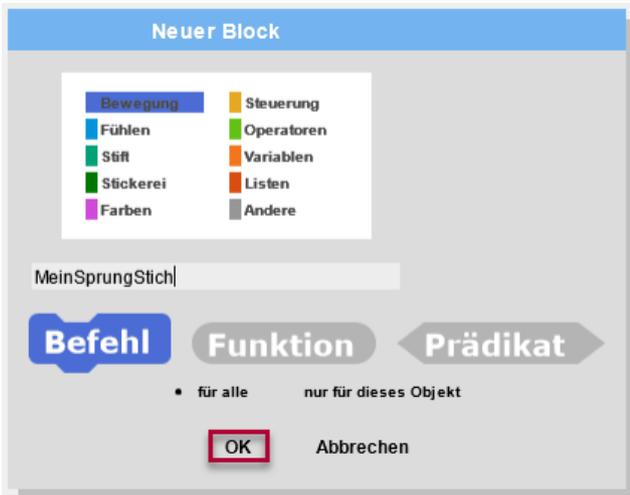


Der rot eingerahmte Teil müsste jetzt extra noch einmal wiederholt werden, bevor wir den nächsten Stichtyp unten anfügen können. Das alles nachzubauen wäre etwas mühselig, aber es geht zum Glück viel eleganter mit einem eigenen Block.

Macht einen Rechtsklick in den Programmierbereich und wählt aus dem Kontextmenü «Neuen Block bauen»:

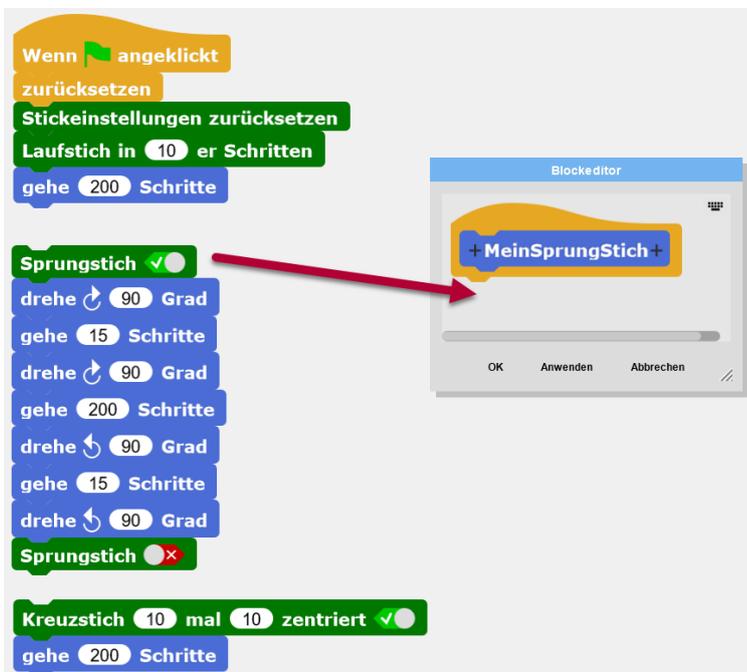


Klickt eine Kategorie an. Da unser Block vor allem Bewegungen ausführen soll, klicken wir entsprechend «Bewegungen» an. Gebt eurem Block noch einen Namen und Klick dann auf «OK»:



Dann trennt ihr den ursprünglichen Code so auf, dass der repetitive Teil einzeln bewegt werden kann. Wenn das erledigt ist, könnt ihr den Teil einfach in den gerade erstellten Block ziehen.

Ihr könntet auch direkt aus der Palette in den Block Editor neue Blöcke reinziehen und so neuen Code erstellen. Aber da wir den Code in unserem Beispiel schon gemacht haben, verwenden wir natürlich diesen. Funktionierenden Code wiederverwenden ist eine bewährte Technik, die ihr natürlich gerne anwenden sollt:

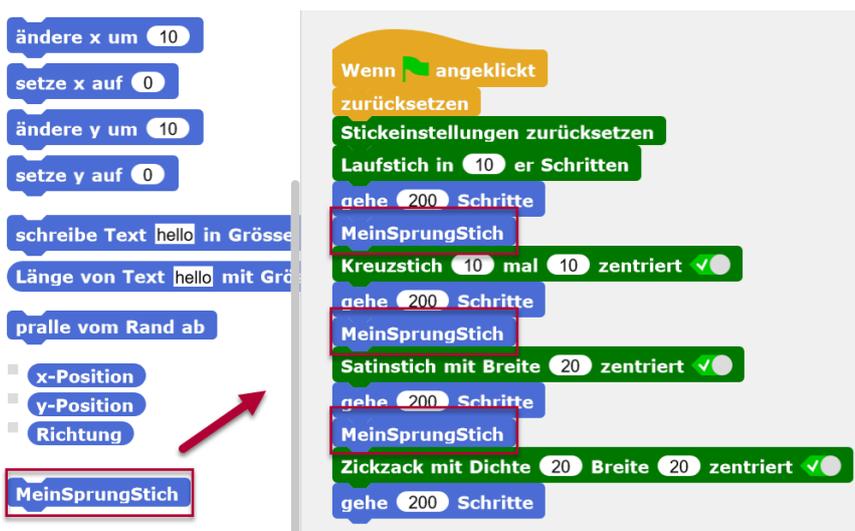


Ihr könnt den Blockeditor mit Ziehen am schraffierten Dreieck grösser machen, um alles darin sehen zu können. Wenn ihr zufrieden seid, könnt ihr auf «OK» klicken, um den Block zu speichern:



Ihr findet den Block nach dem Speichern in der jeweiligen Kategorie, für die ihr den Block erstellt habt. Hier haben wir einen blauen Block gemacht, der unter der Kategorie «Bewegung» ganz unten zu finden ist.

Wir können nun den ursprünglichen Code ersetzen mit dem Block, was viel kürzer und übersichtlicher ist. Ausserdem können wir sehr schnell den Block an zwei weiteren Stellen einsetzen, was natürlich ungemein praktisch ist:



4.4 Kommentare einfügen

Den gerade erstellten Block können wir noch mit einem Kommentar versehen. Dies erhöht die Verständlichkeit des Codes erheblich, sowie für uns wie auch für andere TurtleStitcher.

Klickt mit einem Rechtsklick in die Programmierumgebung und wählt aus dem Kontextmenü «Anmerkung hinzufügen»:



Zieht die Anmerkung auf den Block «MeinSprungStich» um diese zu verlinken, diese sind dann mit einer feinen Linie miteinander verbunden. Tippt noch eine Beschreibung in das Fenster der Anmerkung, um den Block zu beschreiben:



5 Testen und Debugging

Recht häufig kommt es vor, dass ein Computer-Programm beim ersten Testen nicht das macht, was es sollte. Meistens hat man kleine Denkfehler oder Unachtsamkeiten begangen und muss nun diese Fehler finden. Diesen Vorgang nennt man Debugging, also «Käfer finden» 😊

Die nachfolgende Checkliste zeigt die gängigsten Fehler und hilft euch, diese zu finden und zu korrigieren.

5.1 Vergessene Blöcke

Geht für euch in Gedanken die einzelnen Blöcke durch. Ergeben sie Sinn oder fehlt etwas? Falls ihr einen Code abschreibt: Zählt die Blöcke nach. Vielleicht könnt ihr den Code auch mit jemandem vergleichen.

5.2 Verwechselte Blöcke

Habt ihr die richtigen Blöcke verwendet? Besonders häufig werden Blöcke mit relativen und absoluten Anweisungen verwechselt. Bei diesen Beispielen bedeutet der Block links nicht dasselbe wie der Block rechts:

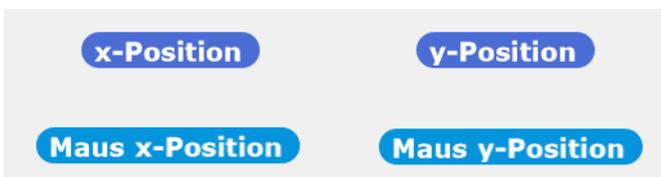


Auch die Farben der Blöcke geben Hinweise auf deren Funktion. Diese beiden Blöcke bedeuten nicht das gleiche, wie an den Farben erkennbar ist:



5.3 x-Koordinaten und y-Koordinaten verwechselt

Unter Umständen habt ihr einen Block für x-Koordinaten erwischt anstatt des Blocks für y-Koordinaten?



5.4 Falsche Zahlen

Wenn ihr zum Beispiel einen Block kopiert habt ist es oft notwendig, die Zahlen oder das Vorzeichen zu ändern:



5.5 Falsche Auswahl

Manchmal geht vergessen, dass man bei Dropdown-Menüs etwas auswählen muss:



5.6 Falsche Reihenfolge

Sind die Blöcke in der korrekten Reihenfolge?



5.7 Schleifen

Sind die Blöcke innerhalb von Schleifen, wenn sie ausserhalb sein sollten, oder umgekehrt?



Ihr seht, dass die beiden Programme ganz unterschiedliche Dinge tun, je nach dem wo wir «zeige Richtung 90» platzieren.

Ihr seht ausserdem, dass wir in beiden Programmen eine Endlosschleife gemacht haben, unsere Schildkröte ist also in ein fleissiges Bienchen mutiert und hört gar nicht mehr auf zu sticken!

Drückt oben rechts auf dem Balken für die Einstellungen auf das rote Stopp-Zeichen, um die Schildkröte anzuhalten:



Wir müssen unser Programm ändern, um die Endlosschleife zu verhindern. Die Schleife muss also eine sogenannte Schleifenbedingung haben, die aussagt, wie lange oder bis wann die Schleife laufen darf.

Wir können das zum Beispiel so machen wie im nächsten Code sichtbar. Die Schildkröte stickt nun also so lange hin und her, bis wir die Leertaste drücken:



Bei dieser Variante läuft die Schleife also so lange ab, bis ein Ereignis eintritt. Auch sehr beliebt ist, eine Schleife eine genaue Anzahl Iterationen durchlaufen zu lassen mit «wiederhole _ mal». Es hängt aber völlig von der Aufgabe ab, welcher Schleifentyp der beste ist. Spielt mit verschiedenen Schleifen herum und berätet euch gegenseitig, welches die beste Schleife ist für welche Aufgabe:



5.8 Initialisieren

Beim Starten eines Programms über das grüne Fähnchen müsst ihr eventuell Einstellungen zurücksetzen, also initialisieren.

Wenn ihr z.B. einen Zähler verwendet wie beim Programm links, zählt dieser bei jedem Programmstart um 1 hoch.

Wenn der Zähler aber bei jedem Start bei 0 anfangen soll und erst danach um 1 hochzählen soll, müsst ihr das dem Programm sagen wie beim Beispielprogramm rechts.



Wenn euer Programm immer etwas anderes macht bei jedem Durchlauf, habt ihr wahrscheinlich vergessen, am Anfang alles zu initialisieren.

5.9 Werte / Zustände anzeigen

Um besser verstehen zu können, was das Programm gerade macht, ist es oft hilfreich, die Variablenwerte auf der Bühne anzeigen zu lassen.

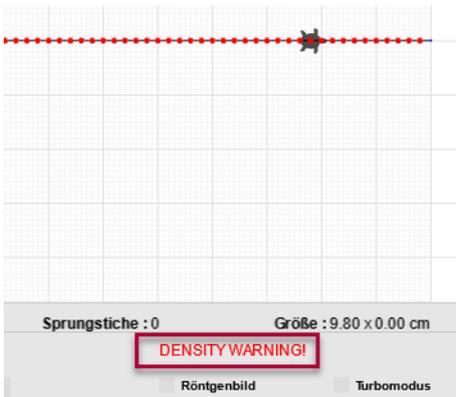
Wenn das Häkchen neben einer Variablen gesetzt ist in der Palette, dann wird deren aktueller Wert in der Bühne angezeigt. Nehmt das Häkchen raus, wenn auch das stören sollte.



5.10 Dichte der Stiche

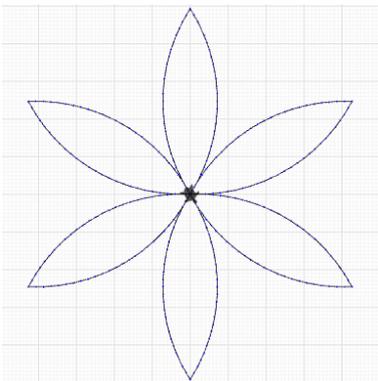
Wenn ihr zu viele Stiche an der gleichen Stelle macht, wird eine «Density Warning!» angezeigt und die betroffenen Stellen sind rot markiert. Dies bedeutet, dass die Stellen beim Sticken allenfalls nicht mehr mit der Nadel durchstochen werden können. **Die Nadel kann sich verbiegen oder gar abbrechen. Dies kann zu Verletzungen führen!**

Ihr dürft solche Muster also auf keinen Fall auf der Stickmaschine sticken und müsst erst den Code und damit das Muster korrigieren.



Gleich unter der «Density Warning!» kann jederzeit das Röntgenbild eingeschaltet werden. Dieses zeigt den Faden in einem dunklen weiss, und je mehr eine Stelle bestickt ist desto satter wird das weiss. Ist die Dichte zu hoch, wird die Stelle wiederum rot wie oben im normalen Modus.

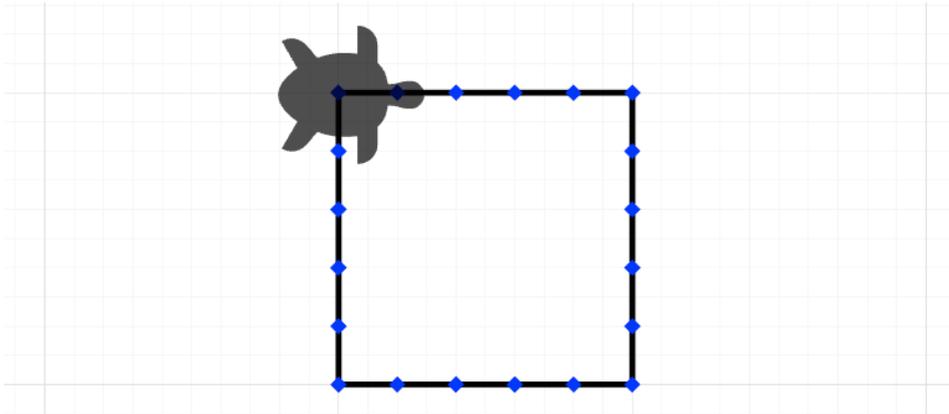
Hier ein Beispiel einer Blume mit sechs Blättern links. Reingezoomt und mit eingeschaltetem Röntgenmodus sieht man, dass im mittleren Teil einige Strecken doppelt bestickt sind und darum heller erscheinen. Das Zentrum ganz in der Mitte ist am hellsten, da Start- und Endpunkt von jedem Blatt im Zentrum liegen, also kommen dort zwölf Stiche hin.



6 Beispiele

6.1 Quadrat

Nun werden wir ein Quadrat sticken. Befolgt hierfür die untenstehenden Schritte und erstellt eine eigene Kopie des Codeblocks.



Verwendete Blöcke:



Dieser Block sorgt dafür, dass die umschlossene Blocksequenz so oft wiederholt wird, wie durch die eingetragene Zahl angegeben.

Mit diesem Block wählt man die Stichart und -länge (hier: einfacher gerader Stich, ein Schritt entspricht dabei 0.2 mm Länge).

Mit diesem Block bestimmt man, wie lang die zu stickende Linie wird. Auch hier entspricht ein Schritt 0.2 mm.

Mit diesem Block wird die Figur um die angegebenen Grad im Uhrzeigersinn gedreht (Achtung: es gibt verschiedene Blöcke für die unterschiedlichen Drehrichtungen).

Klickt die Blöcke zusammen und führt sie aus, indem ihr einmal mit der linken Maustaste auf das Skript klickt.

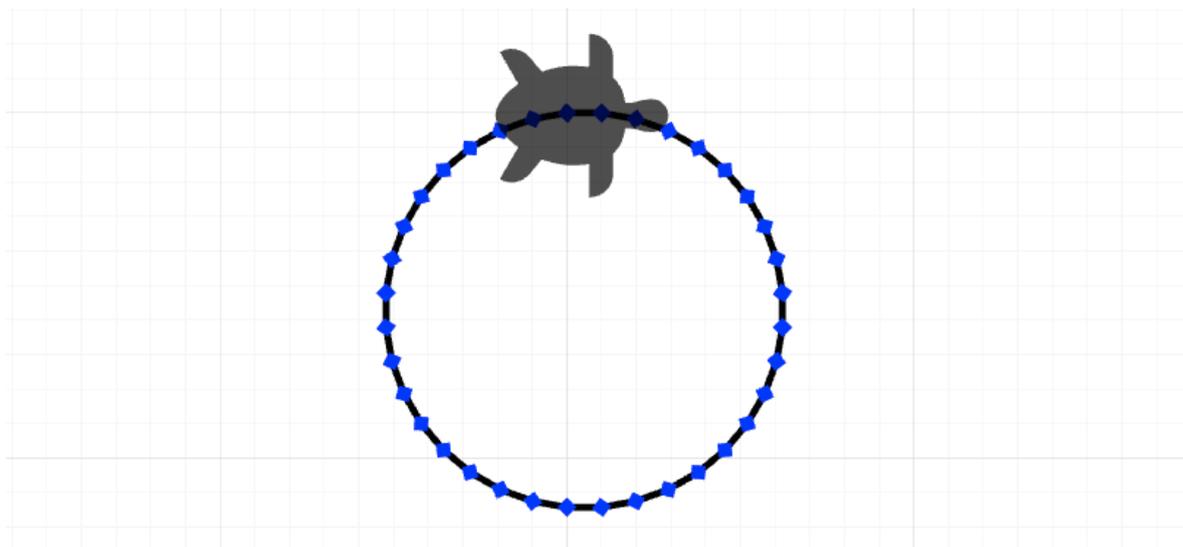
```
Laufstich in 10 er Schritten
wiederhole 4 mal
  wiederhole 5 mal
    gehe 10 Schritte
  drehe 90 Grad
```

Nun hat die Schildkröte ein Quadrat gestickt! Könnt ihr berechnen, wie groß es geworden ist und wie viele Stiche es enthält?

Lösung: Jede Kante ist $5 * 10 * 0.2$ mm lang, also hat das Quadrat eine Größe von $1\text{ cm} * 1\text{ cm}$. Es wurden $4 * 5$, also 20 Stiche gemacht.

6.2 Kreis und n-Eck

Nun werden wir einen Kreis sticken. Befolgt hierfür die untenstehenden Schritte und erstellt eine eigene Kopie des Codeblocks.



Verwendete Blöcke:



Dieser Block sorgt dafür, dass die umschlossene Blocksequenz so oft wiederholt wird, wie durch die eingetragene Zahl angegeben.



Mit diesem Block wählt man die Stichart und -länge (hier: einfacher gerader Stich, ein Schritt entspricht dabei 0.2 mm Länge).



Mit diesem Block bestimmt man, wie lang die zu stickende Linie wird. Auch hier entspricht ein Schritt 0.2 mm.



Mit diesem Block wird die Figur um die angegebenen Grad im Uhrzeigersinn gedreht (Achtung: es gibt verschiedene Blöcke für die unterschiedlichen Drehrichtungen).

Klickt die Blöcke zusammen und führt sie aus, indem ihr einmal mit der linken Maustaste auf das Skript klickt.

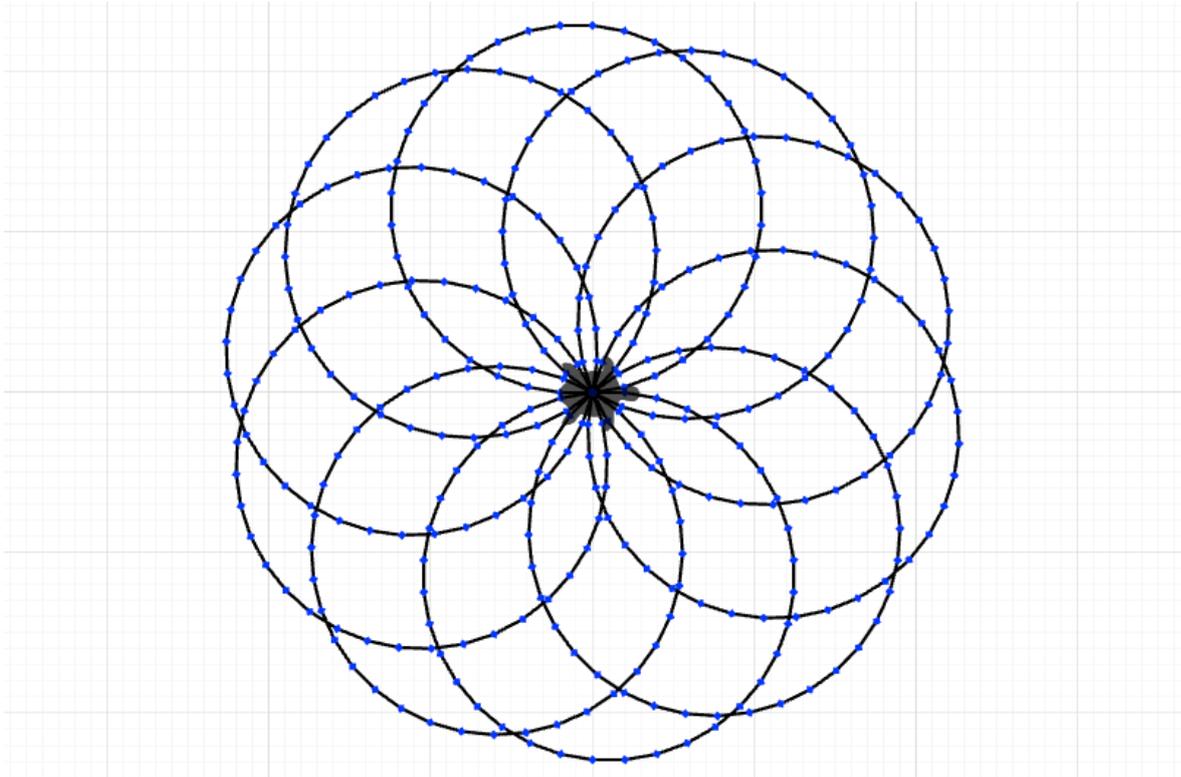


Nun hat die Schildkröte einen Kreis gestickt, genauer genommen: ein 36-Eck! Experimentiert ein wenig mit den Werten. Könnt ihr nur durch Verändern der Zahlen schaffen, dass die Schildkröte ein Sechseck stickt?

Lösung: Bei einem Sechseck müssen sechs Kanten gestickt, also sechs Wiederholungen durchgeführt werden. Damit die Schildkröte wieder am Beginn ankommt, muss sie sich insgesamt um 360° drehen. Bei sechs Drehungen entspricht dies also jeweils einer Drehung von 60°.

6.3 Blüte

Nun wollen wir eine einfache Blüte sticken, die aus Kreisen zusammengesetzt ist. Befolgt hierfür die untenstehenden Schritte und erstell eine eigene Kopie des Codeblocks.



Verwendete Blöcke:



Dieser Block sorgt dafür, dass die umschlossene Blocksequenz so oft wiederholt wird, wie durch die eingetragene Zahl angegeben.



Mit diesem Block wählt man die Stichart und -länge (hier: einfacher gerader Stich, ein Schritt entspricht dabei 0.2 mm Länge).



Dies ist ein selbst erstellter Block, der das Programm ausführt, das in Kapitel 6.2 «Kreis und n-Eck» erstellt wurde, allerdings ohne den Block der Stickkategorie. Wie man eigene Blöcke erstellt, wird in Kapitel 4.3 «Eigene Blöcke erstellen» erklärt.



Mit diesem Block wird die Figur um die angegebenen Grad im Uhrzeigersinn gedreht (Achtung: es gibt verschiedene Blöcke für die unterschiedlichen Drehrichtungen).

Klickt die Blöcke zusammen und führt sie aus, indem ihr einmal mit der linken Maustaste auf das Skript klickt:



Nun hat die Schildkröte eine einfache Blüte mit zehn Blütenblättern gestickt. Wie müsst ihr das Programm verändern, damit die Blüte aus 20 Blütenblättern besteht? Achtung: Hierbei werdet ihr auf ein «Density-Problem» stossen, dies könnt ihr umgehen, in dem ihr die Schildkröte nach jeder Drehung ein wenig weitergehen lasst. Erklärungen hierzu finden sich in Kapitel 5.10 «Dichte der Stiche».

Lösung: Die Anzahl der Blütenblätter bestimmt sich durch die Anzahl der Wiederholungen, denn mit jeder Wiederholung wird genau ein Kreis gestickt. Um 20 Blütenblätter zu sticken, müssen also 20 Wiederholungen durchgeführt werden. Damit auch die Überlappungen entsprechend angepasst werden, muss der Drehwinkel neu berechnet werden. Damit die Schildkröte wieder am Beginn ankommt, muss sie sich insgesamt um 360° drehen. Bei 20 Drehungen entspricht dies also jeweils einer Drehung von 18°.

Nun wollen wir das Programm so erweitern, dass es selbständig berechnet, um wieviel Grad die Schildkröte sich drehen muss, nachdem der Benutzer angegeben hat, wie viele Blütenblätter die Blüte haben soll.

Hierzu werden zusätzlich folgende Blöcke benötigt:

frage Wie heißt Du? **und warte**

Mit diesem Block wird eine Frage an den Benutzer gestellt. Auf der Bühne erscheinen die Frage sowie ein Bereich zum Eintippen der Antwort. Das Programm pausiert, bis der Nutzer seine Eingabe mit Enter bestätigt hat.

Antwort

Dieser Block ist eine Variable, in der die Antwort auf die gestellte Frage gespeichert wird. Sie bleibt so lange erhalten, bis eine neue Frage gestellt und die alte mit der neuen Antwort überschrieben wird.

/

Mit diesem Operator-Block können zwei Zahlen dividiert werden, das Ergebnis der Operation kann per Linksklick auf den Block angezeigt und im Programm direkt verarbeitet werden.

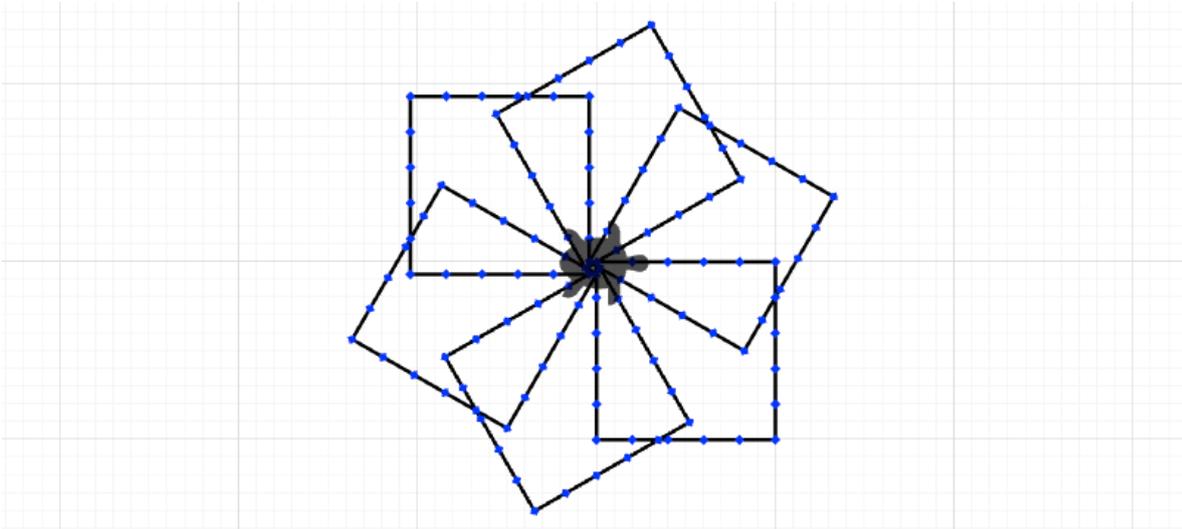
Klickt die Blöcke zusammen und führt sie aus, indem ihr einmal mit der linken Maustaste auf das Skript klickt.

frage Wie viele Blütenblätter soll die Blüte haben? **und warte**
Laufstich in 10 **er Schritten**
wiederhole **Antwort** **mal**
 gehe einen Kreis
 drehe 360 / **Antwort** **Grad**
 gehe 2 **Schritte**

Nun könnt ihr der Schildkröte über das Eingabefenster mitteilen, was für eine Blüte sie stecken soll!

6.4 Windrad

Wir werden nun ein Windrad sticken, das aus Quadraten besteht. Befolgt hierfür die untenstehenden Schritte und erstellt eine eigene Kopie des Codeblocks.



Verwendete Blöcke:



Dieser Block sorgt dafür, dass die umschlossene Blocksequenz so oft wiederholt wird, wie durch die eingetragene Zahl angegeben.



Mit diesem Block wählt man die Stichtart und -länge (hier: einfacher gerader Stich, ein Schritt entspricht dabei 0.2 mm Länge).

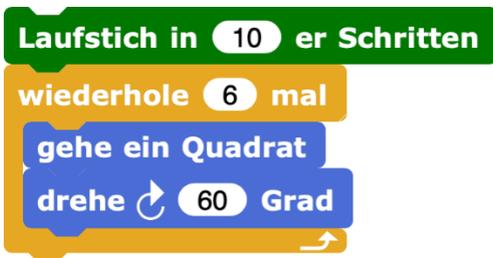
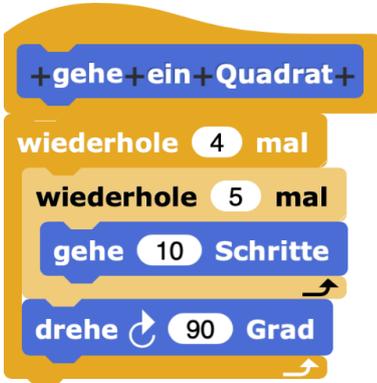


Dies ist ein selbst erstellter Block, der das Programm ausführt, das in Kapitel 6.1 «Quadrat» erstellt wurde, allerdings ohne den Block der Stickkategorie. Wie man eigene Blöcke erstellt, wird in Kapitel 4.3 «Eigene Blöcke erstellen» erklärt.



Mit diesem Block wird die Figur um die angegebenen Grad im Uhrzeigersinn gedreht (Achtung: es gibt verschiedene Blöcke für die unterschiedlichen Drehrichtungen).

Klickt die Blöcke zusammen und führt sie aus, indem ihr einmal mit der linken Maustaste auf das Skript klickt.



Nun hat die Schildkröte ein einfaches Windrad mit sechs Rotorblättern gestickt. Wie müsst ihr das Programm verändern, damit das Windrad aus 20 Rotorblättern besteht? Achtung: Hierbei werdet ihr auf ein «Density-Problem» stossen, dies könnt ihr umgehen, in dem ihr die Schildkröte nach jeder Drehung ein wenig weitergehen lasst. Erklärungen hierzu finden sich in Kapitel 5.10 «Dichte der Stiche».

Lösung: Die Anzahl der Rotorblätter bestimmt sich durch die Anzahl der Wiederholungen, denn mit jeder Wiederholung wird genau ein Quadrat gestickt. Um 20 Rotorblätter zu sticken, müssen also 20 Wiederholungen durchgeführt werden. Damit auch die Überlappungen entsprechend angepasst werden, muss der Drehwinkel neu berechnet werden. Damit die Schildkröte wieder am Beginn ankommt, muss sie sich insgesamt um 360° drehen. Bei 20 Drehungen entspricht dies also jeweils einer Drehung von 18°.

Nun wollen wir das Programm so erweitern, dass es selbständig berechnet, um wieviel Grad die Schildkröte sich drehen muss, nachdem der Benutzer angegeben hat, wie viele Rotorblätter das Windrad haben soll.

Hierzu werden zusätzlich folgende Blöcke benötigt:

frage Wie heißt Du? **und warte**

Mit diesem Block wird eine Frage an den Benutzer gestellt. Auf der Bühne erscheinen die Frage sowie ein Bereich zum Eintippen der Antwort. Das Programm pausiert, bis der Nutzer seine Eingabe mit Enter bestätigt hat.

Antwort

Dieser Block ist eine Variable, in der die Antwort auf die gestellte Frage gespeichert wird. Sie bleibt so lange erhalten, bis eine neue Frage gestellt und die alte mit der neuen Antwort überschrieben wird.

/

Mit diesem Operator-Block können zwei Zahlen dividiert werden, das Ergebnis der Operation kann per Linksklick auf den Block angezeigt und im Programm direkt verarbeitet werden.

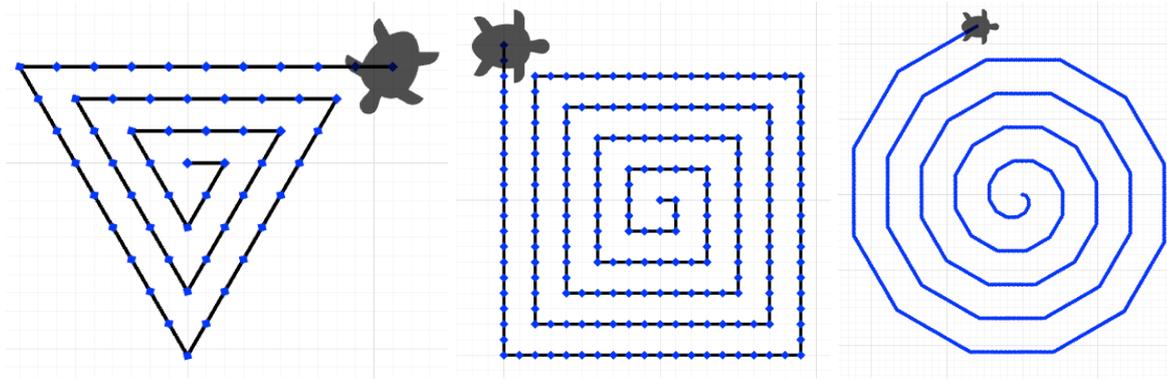
Klickt die Blöcke zusammen und führt sie aus, indem ihr einmal mit der linken Maustaste auf das Skript klickt.

frage Wie viele Rotorblätter soll das Windrad haben? **und warte**
Laufstich in 10 **er Schritten**
wiederhole **Antwort** **mal**
gehe ein Quadrat
drehe 360 / **Antwort** **Grad**

Nun könnt ihr der Schildkröte über das Eingabefenster mitteilen, was für ein Windrad sie stecken soll!

6.5 Spiralen

Jetzt möchten wir verschiedene Spiralformen sticken! Befolgt hierfür die untenstehenden Schritte und erstellt eine eigene Kopie des Codeblocks.



Verwendete Blöcke:



Dieser Block sorgt dafür, dass die umschlossene Blocksequenz so oft wiederholt wird, wie durch die eingetragene Zahl angegeben.

Mit diesem Block wählt man die Stichart und -länge (hier: einfacher gerader Stich, ein Schritt entspricht dabei 0.2 mm Länge).

Mit diesem Block wird eine vorab erstellte Variable mit einem Wert belegt.

Mit diesem Block wird der Wert einer vorab erstellten Variablen um einen Wert geändert (Addition). Subtraktion lässt sich durch Addition negativer Zahlen bewerkstelligen.

Mit diesem Block wird die Figur um die angegebenen Grad im Uhrzeigersinn gedreht (Achtung: es gibt verschiedene Blöcke für die unterschiedlichen Drehrichtungen).

Klickt die Blöcke zusammen und führt sie aus, indem ihr einmal mit der linken Maustaste auf das Skript klickt.

Dreiecksspiralen



Quadratspiralen



Kreisspiralen



Nun kann die Schildkröte verschiedene Spiralformen sticken! Aber warum wirkt die Kreisspirale nach aussen immer kantiger?

Lösung: Je weiter innen man in die Spirale schaut, desto kürzer sind die Schritte, die zwischen zwei Drehungen erfolgen (erst 1, dann 2, dann 3 usw.). Dadurch erscheinen die Verbindungen zwischen zwei Einstichstellen noch eher als Punkte und weniger als Linien. Je weiter außen man schaut, desto größer werden die Abstände zwischen den Drehungen und umso kantiger wirkt auch die Spirale. Genau genommen ist es keine Kreisspirale, sondern eine n-Eck-Spirale. Je nach Drehwinkel lässt sich die Anzahl der Ecken der Grundform (n-Eck) wie folgt berechnen: $n = 360 / \text{Drehwinkel}$.

Experimentiert ein wenig mit den Werten herum (Anzahl Wiederholungen und Drehwinkel), um verschiedene Spiralformen zu erzeugen.

7 Sticken mit der Stickmaschine

7.1 Mustergösse

Die Stickmuster sollten nicht grösser als 18 cm Höhe und 13 cm Breite haben. Der verwendete Stickrahmen ist sonst zu klein (jedoch abhängig von der verwendeten Stickmaschine und Zubehör).

7.2 Übertragung mit USB-Stick

Um ein Muster auf die Stickmaschine zu übertragen, braucht man einen USB-Stick. Am besten ihr verwendet dafür einen leeren USB-Stick, auf dem keine anderen Daten gespeichert sind.

Ihr könnt auf dem USB-Stick eigene Ordner machen und die Muster darin speichern, falls die Stickmaschine dies lesen kann. Die Stickmaschine «Brother Innov-is 750E» kann aber **keine weiteren Unterordner** lesen, also macht bitte maximal eine Ebene Ordner.

Benutzt ausserdem **keine Sonderzeichen** für den Namen der Ordner oder der Muster.

Zum Exportieren eures Musters müsst ihr euer Projekt mit dem Code geladen haben und es ist im Programmierbereich sichtbar. Dann wählt ihr «Exportieren als Tajima/DST» bei den Funktionen unterhalb der Bühne:



Wählt dann den Speicherort aus und speichert die DST Datei mit einem Namen wie z.B. «Mein erstes Programm.dst».

Wie man die Datei mit der Stickmaschine vom USB-Stick einliest und das Muster stickt, zeigen wir euch in der entsprechenden Anleitung.

Wir wünschen euch viel Spass beim Programmieren und Sticken!



Goldau, 29. Juli 2021

Beat Horat (beat.horat@phsz.ch)

Mareen Przybylla (mareen.przybylla@phsz.ch)