Für die Einführung in Scratch haben wir ein Projekt aus der sehr wertvollen Broschüre von Urs Frischherz ausgewählt. Die ganze Broschüre und weitere spannende Scratch-Ideen finden Sie unter: https://www.zebis.ch/unterrichtsmaterial/scratch-30-kennenlernen

SCRATCH 3.0 kennenlernen

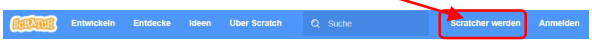
Teil I (ohne Variablen)

Urs Frischherz

Scratch-Konto einrichten

0

Um bei Scratch ein Konto zu eröffnen, klickst du auf der Startseite von Scratch (https://scratch.mit.edu/) den Reiter «Scratcher werden». Die Anmeldung ist kostenlos und ohne jede Verpflichtung.



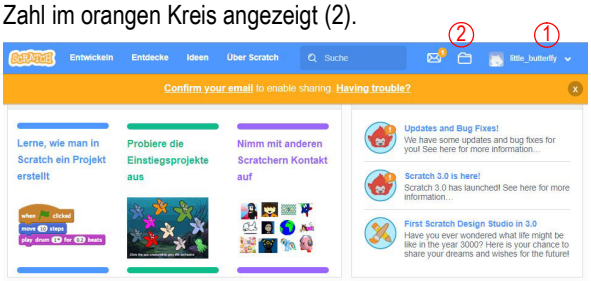
Danach wirst du durch das Anmelde-Prozedere geführt:

Für die Wahl des Scratch-Benutzernamens lohnt es sich, einen Moment zu investieren. Andere Scratcher werden ihn sehen und sich gegebenenfalls daran erinnern.
Tabu ist es, den eigenen Namen zu nehmen.

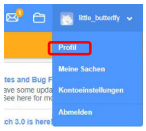
Nachdem noch Geburtsjahr und Geburtsmonat, das Geschlecht und das Herkunftsland angegeben werden müssen, bleibt als Letztes noch die Mailadresse, welche zur Bestätigung des Kontos dient.

Danach öffnest du am besten dein Mailkonto und bestätigst durch Klicken des bereitgestellten Links dein Konto.

Jetzt wird auf der Scratch-Webseite dein Scratcher-Name oben rechts angezeigt (1), zum Zeichen, dass du angemeldet bist.
Weiter kannst du erkennen, dass du eine Nachricht erhalten hast. Nachrichten, aber auch andere Aktivitäten, im Zusammenhang mit deinem Konto, werden durch eine Zahl im orangen Kreis angezeigt (2).



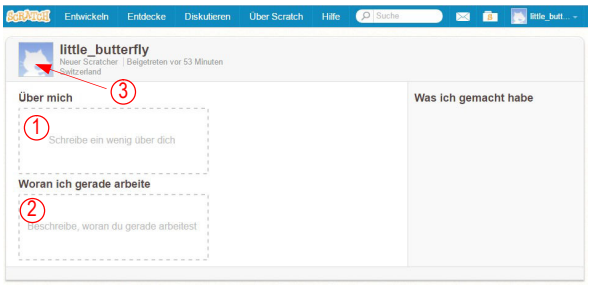
Falls du magst, kannst du dein Konto noch etwas persönlicher gestalten. Dazu klickst du auf deinen Benutzernamen oben rechts und wählst *Profil*. Danach hast du folgende drei Optionen.



1) Über mich: Ist eine Gratwanderung. Personaldaten, Name, Alter, Geschlecht sollten nicht erwähnt werden. Passen würden Hobbies und Interessen.

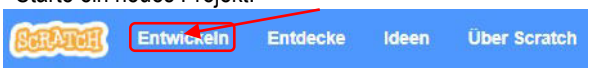
2) Woran ich arbeite: Gemeint sind da Scratch-Projekte.

3) Profilbild: Im Gegensatz zu den Texten, welche viele Scratcher offen lassen, kreiert eigentlich fast jeder sein eigenes Profilbild. Auch hier gilt aber: kein persönliches Foto. Ansonsten sind der Fantasie keine Grenzen gesetzt. Nebst Bildern sind übrigens auch bewegte GIFs möglich.
Um sein persönliches Bild einzusetzen auf den Platzhalter im Profil doppelklicken und zum gewünschten Bild navigieren.

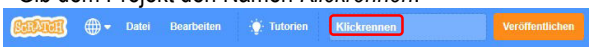


Projekt starten

• Starte ein neues Projekt:

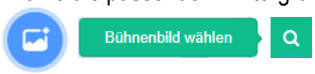


• Gib dem Projekt den Namen *Klickrennen*.

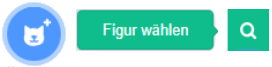


Eine Figur auswählen und verändern

• Wähle als passenden Hintergrund *Night City*.

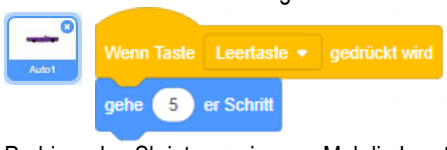


• Lösche die Katzen-Figur mit Rechtsklick und wähle stattdessen die Figur *Convertible*.



• Ändere den Namen in *Auto1*, indem du das Wort *Convertible* bei den Figuren-Infos überschreibst.

• Schreibe folgendes Skript, damit sich das Auto mittels Klick auf die Leertaste bewegen lässt:

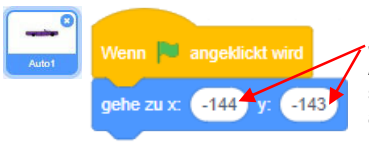


• Probiere das Skript aus ein paar Mal die Leertaste klickst.

Auf die Plätze

• Ziehe das Auto in die linke untere Ecke.

• Wechsle in den *Skripte*-Register und schreibe ein weiteres Skript für das Auto, damit es nach jedem Klick auf die grüne Flagge automatisch wieder am richtigen Ort startet:

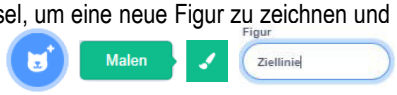


Je nachdem, wo du dein Auto hingezogen hast, sehen deine Werte etwas anders aus.

• Teste dein Skript, indem du das Auto in die Mitte der Bühne ziehst und danach die grüne Flagge klickst.

Eine Ziellinie einrichten

• Klicke auf den Pinsel, um eine neue Figur zu zeichnen und nenne sie *Ziellinie*:



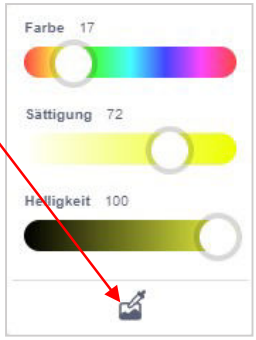
• Wähle das Linienwerkzeug aus und ziehe damit eine gelbe senkrechte Linie deren Dicke du auf 10 eingestellt hast. . Halte, während du die Linie ziehst, die Umschalttaste gedrückt, damit die Linie senkrecht bleibt. Klicke auf die Delete-Taste, wenn du die Linie nochmals zeichnen möchtest.

• Die Linie sollte etwa einen Drittel der Bühnenhöhe haben. Ziehe sie anschliessend auf der Bühne an den richtigen Ort.

• Klicke auf die Auto-Figur und wechsle anschliessend ins *Skripte*-Register.

• Ergänze das Leertasten-Skript des Autos wie folgt:

Um die gewünschte Farbe in den Sensor zu bekommen, klickst du zuerst auf das Farbfeld des Sensors und danach auf die Farbpipette im unteren Teil des erscheinenden Farbfensters. Jetzt kannst du auf der Bühne auf diejenige Farbe klicken, welche du im Farbfeld des Sensors haben möchtest.



• Teste das Skript: Klicke solange auf die Leertaste, bis das Auto die Ziellinie erreicht.

Einen Gegner wählen

• Wähle eine zusätzliche Figur: Convertible 2

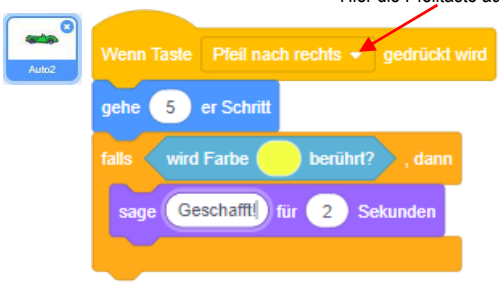
• Benenne sie *Auto2*

• Ziehe sie auf der Bühne auf gleiche Höhe wie das *Auto1* und benütze deren Positionsangaben (Koordinaten) in einem Skript, um die Startposition festzulegen:

Je nachdem, wo du das *Auto2* hingezogen hast, sehen deine Werte etwas anders aus.

Pfeiltaste bewegt werden kann.

Hier die Pfeiltaste auswählen.



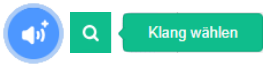
- Teste dein Skript: Du kannst mit der Leertaste, bzw. der rechten Pfeiltaste die Figuren zum Bewegen bringen.

Einen Klang hinzufügen

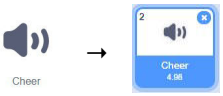
Wenn die Ziellinie überschritten wird, soll ein Klang ertönen.

- Klicke auf *Auto1* anschliessend auf das Klänge-Register.

- Klicke auf *Klang wählen*.



- Wähle den Klang *cheer* aus.

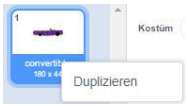


- Wechsle ins *Skripte*-Register und ergänze das bereits bestehende Skript:

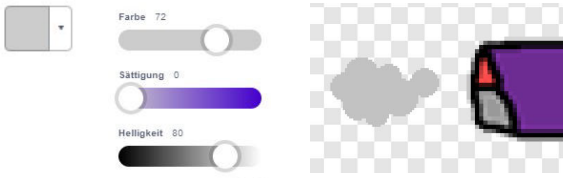
- Teste das Skript, indem du solange auf die Leertaste klickst, bis das Auto die Ziellinie überschreitet.

Das Auto qualmen lassen

- Wechsle ins Register *Kostüme*. Durch Rechtsklicken auf das Kostüm links oben kannst du es duplizieren.



- Wähle danach das Kreiswerkzeug aus.
- Bei der Füllfarbe mischt du dir ein helles Grau zusammen. Anschliessend zeichnest du mit einigen kleinen grauen Kreisen hinter dem *Auto1* etwas Qualm.



- Wechsle ins *Skripte*-Register und ergänze das Skript:

- Der Auspuff sollte nun qualmen, wenn du das Skript testest.

Gegen den Computer antreten

- Das zweite Auto soll vom Computer nun automatisch bewegt werden. Wähle deshalb die Figur *Auto2* aus.
- Lösche anschliessend das Pfeiltasten-Skript, indem du es auf die Block-Palette zurückziehst.
- Ergänze das verbleibende Skript wie folgt:

Du kannst die Geschwindigkeit verändern, wenn du hier eine grössere oder kleinere Zahl einsetzest.

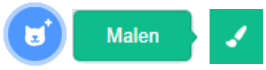
Legen die Zielposition fest. Die Zahlen können in deinem Projekt leicht abweichen (Je nachdem, wo du die Ziellinie hingesetzt hast.

- Teste das Skript, indem du gegen den Computer antrittst (grüne Flagge und danach die Leertaste möglichst schnell klicken). Probiere andere Geschwindigkeiten für das grüne Auto aus, bis für dich passt.

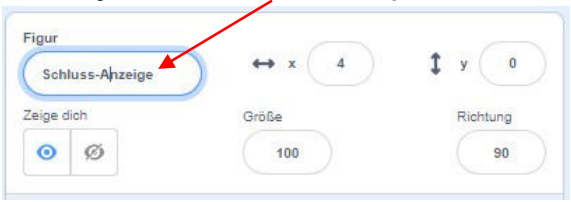
Gewonnen oder verloren?

Du kannst dir vom Programm anzeigen lassen, ob du gewonnen oder verloren hast:

- Dazu wählst du eine neue, leere Figur, indem du *Malen* auswählst.

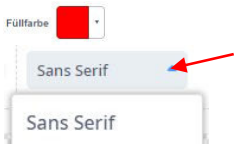


- Gib der Figur den Namen *Schluss-Anzeige*.

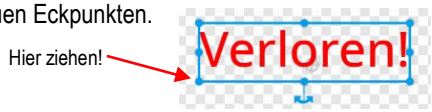


- Klicke auf das Textwerkzeug T und schreibe *Verloren!*

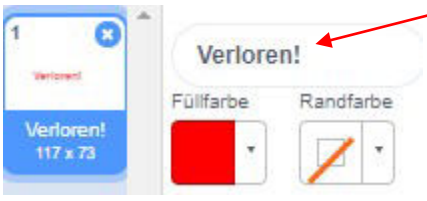
- Du kannst bei diesem Schriftzug…
  - die Farbe ändern:
  - die Schriftart ändern:



  - die Grösse ändern: Klicke dafür auf das *Auswählen*-Werkzeug und ziehe danach an einem der erscheinenden blauen Eckpunkten.



- Verändere nun den Schriftzug nach deinem Geschmack und ziehe ihn danach in die Mitte der Zeichnungsfläche, solange das *Auswählen*-Werkzeug noch angewählt ist.
- Ändere links oben den Namen des *Kostüm1* in *Verloren!*



- Dupliziere das Kostüm *Verloren!* durch Rechtsklick und ändere links oben seinen Namen in *Gewonnen!*

- Wähle das Textwerkzeug T aus und klicke auf der Arbeitsfläche auf das Wort *Verloren!* Ändere es ab in *Gewonnen!*

Deine Figur *Schlussanzeige* hat nun zwei Kostüme: *Verloren* und *Gewonnen*.

Um die Gewinn-, beziehungsweise die Verloren-Anzeige im richtigen Moment erscheinen zu lassen, muss noch einiges programmiert werden:

- Ändere folgende beiden Skripte ab:

Sendet eine Nachricht an alle Figuren. Durch Klicken auf den weissen Pfeil und danach auf *Neue Nachricht* können weitere Nachrichten kreiert werden.

- Damit die die Figur *Schluss-Anzeige* die Nachrichten empfangen kann, benötigt sie noch weitere Skripts:

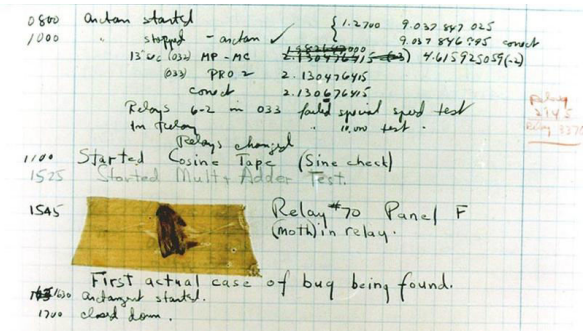
Versteckt die Anzeige, bis sie gebraucht wird.

Sobald die Figur diese Meldung erhält, wechselt sie zum richtigen Kostüm, zeigt sich und stoppt danach das Spiel.

- Teste nun die Skripts. Bei Spielende sollten nun die passende Anzeige erscheinen.

Debugging

Recht häufig kommt es vor, dass ein Computer-Programm beim ersten Test alles andere macht, als was es sollte. In den seltensten Fällen ist dann der Computer schuld. Meistens hat der Scratcher einen Denkfehler oder eine Unaufmerksamkeit begangen und muss nun den Fehler suchen.
Programmierer nennen das *Debugging*. Das Wort kommt vom Wort *Bug*. In grauer Computer-Vorzeit, im Jahre 1947, hatte Computer-Pionierin Grace Hopper eine Motte als Ursache für eine Computerpanne ausgemacht. Sie hat den *Bug* in ihrem Logbuch verewigt und seither hält sich der Name Bug (Ungeziefer, Wanze) für Programmfehler.

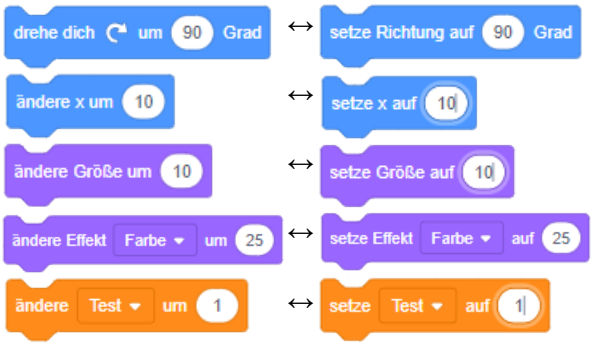


Debuggen ist nicht besonders spannend, aber manchmal kommt man nicht darum herum. Die nachfolgende Checkliste kann dir beim Suchen helfen, wenn dein Programm nicht so läuft, wie du dir das vorgestellt hast:

1. Skript am falschen Ort
Hast du das Skript bei der richtigen Figur geschrieben?

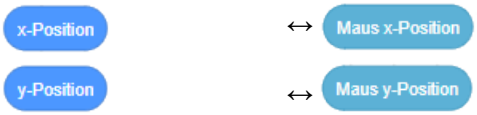
2. Vergessene Blocks
Gehe für dich in Gedanken die einzelnen Blocks durch. Machen sie Sinn oder fehlt etwas? Falls du einen Code abschreibst: Zähle die Blocks nach. Vielleicht kannst du deinen Code auch mit deiner Kollegin, deinem Kollegen vergleichen.

3. Verwechselte Blocks
Hast du die richtigen Blocks verwendet? Besonders häufig werden Blocks mit relativen und absoluten Anweisungen verwechselt:

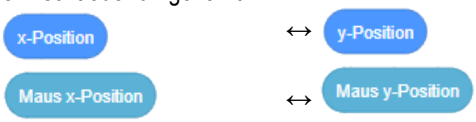


Auch Blocks mit ähnlichen Anweisungen sind gefährdet:

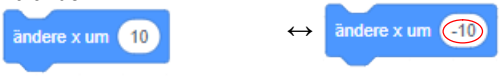
Vielleicht haben die Blocks aber auch nicht die richtige Farbe und damit die falsche Funktion:



4. x und y verwechselt
Unter Umständen hast du einen X-Block statt einen Y-Block erwischt oder umgekehrt?

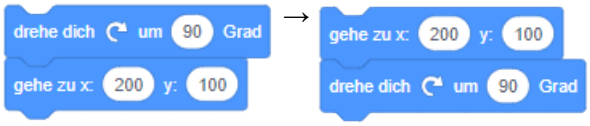


5. Falsche Zahlen
Gerade wenn du ein Skript oder ein Skript-Teil kopiert hast, ist es oft notwendig, noch die Zahlen oder das Vorzeichen zu ändern:

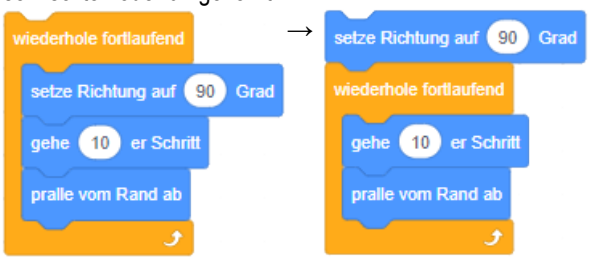


6. Falsche Auswahl
Ebenfalls geht das Ändern der Dropdown-Menüs schnell vergessen:

7. Falsche Reihenfolge
Sind die Blocks in der korrekten Reihenfolge?



8. Schleifen
Sind Blocks innerhalb von Schleifen, wenn sie ausserhalb sein sollten oder umgekehrt?

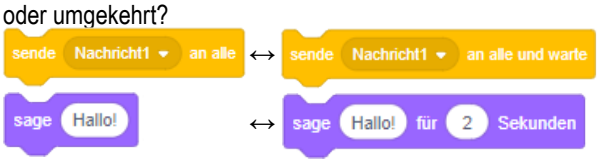


Oder fehlt gar eine Schleife? Statt so,

sollte das Skript so aussehen:

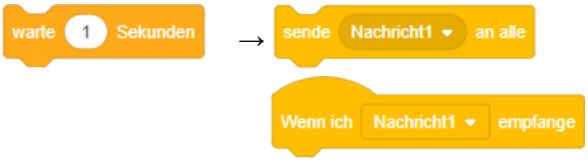
9. Zeitliche Begrenzung

Sind Aktionen zeitlich begrenzt, wo sie es nicht sein sollten oder umgekehrt?



10 Falsches Timing

Um einen genauen zeitlichen Ablauf mit mehreren Figuren festzulegen, sind die *sende-empfange*-Blocks den *warte*-Blocks vorzuziehen.



Tipp: Verschwundene Figur

Wenn plötzlich eine Figur verschwunden ist, hilft oft das folgende Skript. Schreibe es in den Skriptbereich der vermissten Figur und klicke darauf: