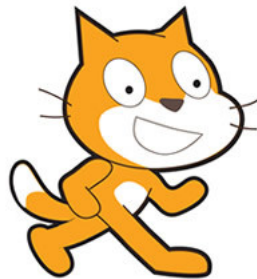


Scratch

Hr. Walter, Hr. Ehret und Hr. Ribeaud

16. Juli 2017

SCRATCH



Inhaltsverzeichnis

1	Projekt 1: Formel 1	2
1.1	Die Rennstrecke	2
1.2	Das Auto	2
1.3	Die Skripte	3
1.4	Fragen zum Autorennen	3
1.5	Erweiterungen	3
1.5.1	Gas geben und bremsen	3
1.5.2	Autorennen mit mehreren Szenen	4
2	Projekt 2: Ein gespielter Witz	4
2.1	Eine Comicfigur zeichnen	5
2.2	Die Skripte	5
2.2.1	Synchronisation durch Warten	5
2.2.2	Synchronisation durch Nachrichten	5

1 Projekt 1: Formel 1

Dieses Projekt ist ein Autosimulator. Die Bühne zeigt eine Rennstrecke, über die ein Auto fährt. Du steuerst das Auto mit den Pfeiltasten. Immer wenn du auf der rechten Pfeiltaste drückst, dreht es sich um ein Stückchen nach rechts. Wenn du auf der linken Pfeiltaste drückst, dreht es sich nach links. Das Auto fährt mit konstanter Geschwindigkeit. Wenn es die Strasse verlässt, hält es an.

1.1 Die Rennstrecke

Lege ein neues Projekt an und speichere es in deinem Projektordner z.B. unter dem Name *auto* ab. Als erstes zeichnest du die Rennstrecke.

Klicke auf das Symbol der Bühne und wähle den Reiter *Hintergründe*. Klicke dann auf die Schaltfläche *Bearbeiten*.

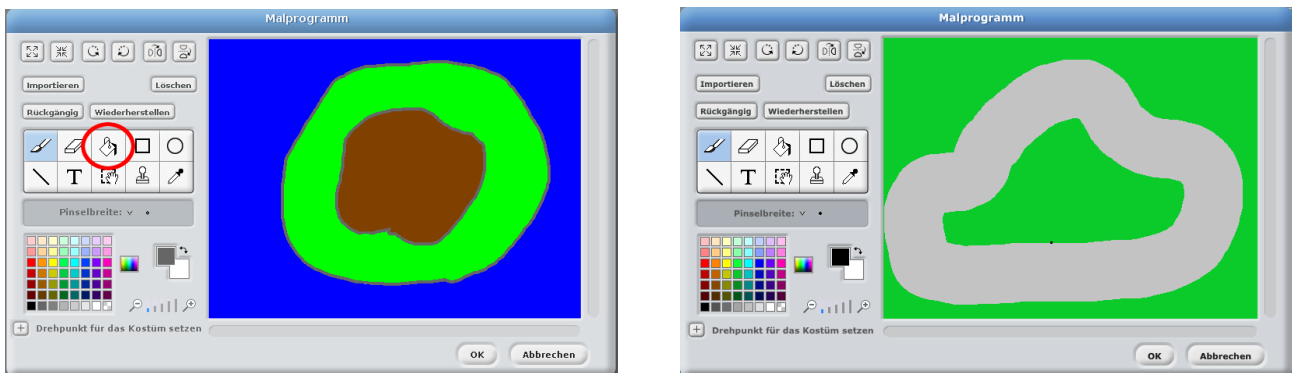



Abbildung 1: Rennbahn als Bühnenbild

1.2 Das Auto

Jedes neue Scratch-Projekt enthält bereits ein Objekt - die Katze. Aber jetzt brauchst du keine Katze, du brauchst ein Auto.

Das Malprogramm, um ein neues Sprite zu zeichnen, kann sich auf eine der folgenden Weisen öffnen:

- Doppelklicken auf der Katze, dann Schaltfläche *Malen* neben *Neues Kostüm*: anklicken.
- Klicke dann auf die Schaltfläche *Neues Objekt Malen* 

Halte die Form klar und einfach. Denn das Auto wird auf dem Bildschirm nur zu sehen sein. Vergiss nicht, dem neuen Objekt einen sinnvollen Namen zu geben, z.B. **Auto**.

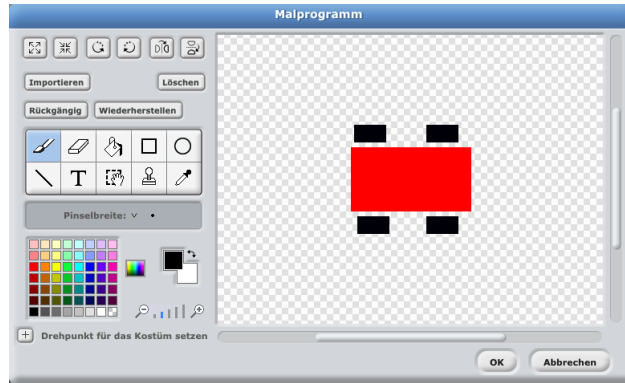


Abbildung 2: Ein Auto. Du kannst auch ein realistischeres Bild eines Formel-1-Wagens von oben zeichnen. Tipp: Zeichne zuerst ein rotes Rechteck und entferne dann mit dem Radiergummi kleinere Stücke.

1.3 Die Skripte

Das Auto-Objekt erhält für die Steuerung **drei** Skripte. Das erste Skript wird zu Beginn einmal gestartet und läuft dann so lange, bis der Wagen von der Bahn abkommt und die grüne Umgebung berührt.

Die beiden anderen Skripte werden immer ausgeführt, und zwar immer dann, wenn eine Pfeiltaste gedrückt wird.

1.4 Fragen zum Autorennen


- Wie kannst du das Auto schneller machen? Es gibt zwei Möglichkeiten!
- Was ändert sich, wenn in *Zeige Richtung...*-Anweisung statt der Zahl 90 die Zahl 0 steht?

1.5 Erweiterungen

1. Lass einen Klang lauten, wenn das Auto die Strasse verlässt.
2. Das Auto sollte weiterfahren, wenn du die Leertaste drückst.
3. Anzeige einer Stoppuhr.

1.5.1 Gas geben und bremsen

Wenn du in deinem Rennwagen Gas gibst, wird die Geschwindigkeit höher. Wenn du bremst, wird sie niedriger. Das heisst: Die Schrittlänge ändert sich. Das Problem ist also, die Zahl in dem *Gehe*-Baustein. Wenn die Geschwindigkeit geändert werden kann, darf hier überhaupt keine feste Zahl stehen.

Was tun? Die Lösung ist eine **Variable**. An Stelle einer Zahl setzen wir in das Zahlfenster eine Variable, z.B.  (engl. *speed*=Geschwindigkeit).

1.5.2 Autorennen mit mehreren Szenen

Erweitere das Projekt auf folgende Weise. Die Rennstrecke erstreckt sich über *zwei* Hintergrundbilder wie in Abb. 3. Wenn der Rennwagen den rechten Rand des ersten Bildes berührt, erscheint ein neues Hintergrundbild mit dem zweiten Teil der Rennstrecke. Der Wagen springt an den linken Rand der Bühne und fährt von links über das Bild weiter. Anders herum funktioniert es genauso: Wenn der Wagen sich im rechten Teil der Rennstrecke befindet und gegen den linken Rand fährt, ändert sich wieder das Hintergrundbild.

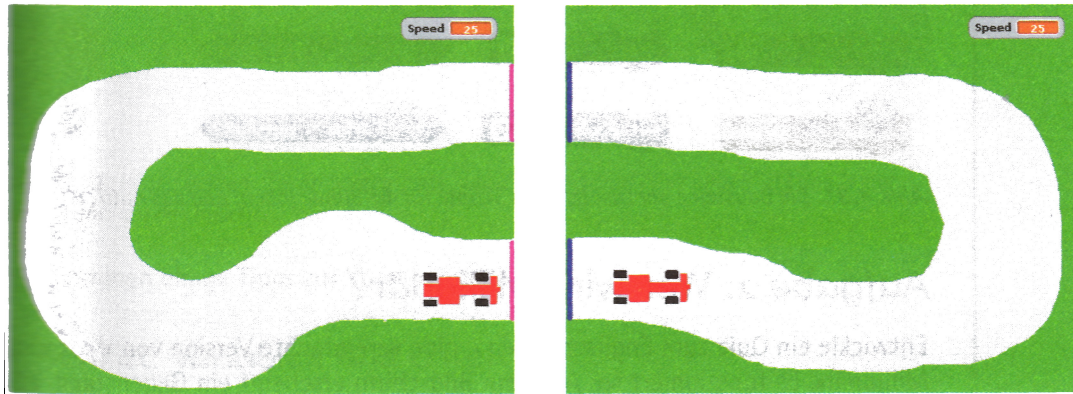


Abbildung 3: Eine Rennstrecke, die sich über zwei Hintergrundbilder erstreckt.

2 Projekt 2: Ein gespielter Witz

In diesem Projekt geht es um Animationen mit mehreren Figuren. Du entwickelst einen Dialog zwischen zwei Figuren.

Abb. 4 zeigt, was passiert. Da kommt ein Hund auf die Bühne und sagt etwas. Dabei öffnet und schließt sich der Mund. Der zweite Hund antwortet. Es ergibt sich ein Gespräch mit einem Lacher am Ende.

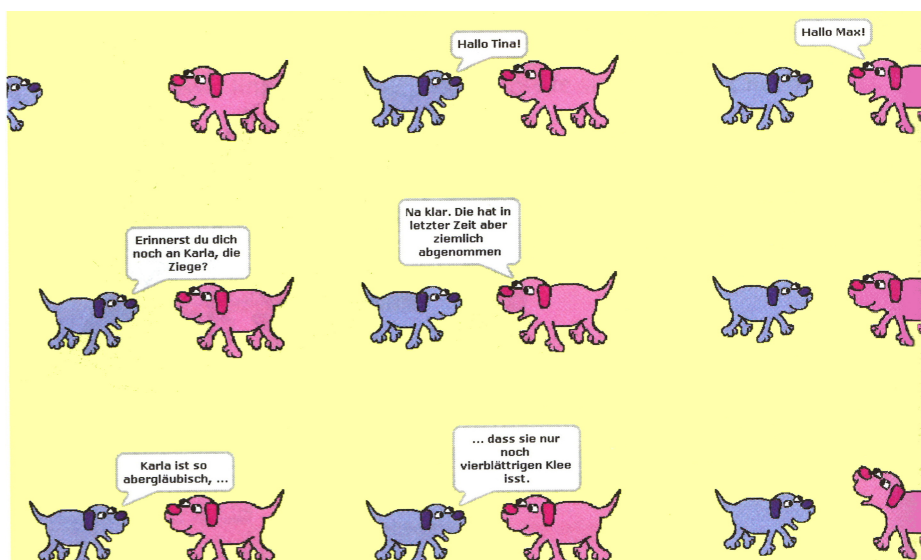


Abbildung 4: Ein animiertes Gespräch zwischen zwei Comicfiguren

Was ist die Besonderheit einer solchen Animation? Wir haben hier zwei Objekte, die aufeinander reagieren. Niemand fällt dem anderen ins Wort. Jeder wartet, bis er an der Reihe ist. Die Antwort kommt zur richtigen Zeit - nämlich erst dann, wenn eine Frage gestellt worden ist. Das nennt man Synchronisation.

2.1 Eine Comicfigur zeichnen

Natürlich kannst du eine Figur mit dem Malprogramm selbst zeichnen. Aber es ist auch interessant, eine bereits fertige Zeichnung zu übernehmen und abzuwandeln.

Auf dem Raspberry Pi findest du schon ein Angebot von Figuren. Wenn dir hier nichts gefällt, suche im Internet nach schönen Bildern. Anschließend kannst du in Scratch ein neues Objekt erzeugen, das das gefundene Bild als Kostüm trägt.



Das neue Objekt erscheint sofort auf der Bühne. Es hat ein Kostüm, das du nun abwandelst. Gib dem Objekt einen Namen.

Klicke auf den Reiter *Kostüme*. Dann ist die Registrierkarte mit den Kostümen sichtbar. Klicke dann auf *Bearbeiten*. Dann öffnet sich das Malprogramm und du kannst das Bild ändern.

Das Bild ist bestimmt viel zu gross. Klick auf Schrumpfen um es zu verkleinern. Als nächstes musst du die Umrisse deines Objektes ausschneiden. Für den kleineren Hund (**Max**) brauchst du mindestens zwei Kostüme, eins mit geschlossenem und eines mit geöffneten Mund. Kopiere das erste Kostüm und bearbeite es.

Tina, der zweite Hund, schaut in eine andere Richtung als Max. Deshalb wirst du im ersten Schritt das Bild von Max spiegeln.

Färbe das Kostüm anders ein und mach es ein bisschen größer. Dann machst du zwei Kopien, so dass du insgesamt drei Kostüme für das Objekt **Tina** hast. Denke daran, den Kostümen sinnvolle Namen zu geben, z.B. *Mund auf*, *Mund zu*, *Lachen*.

In den Skripten verwendest du diese Namen. Da sollte man schon am Namen erkennen, welches Kostüm gemeint ist.

2.2 Die Skripte

2.2.1 Synchronisation durch Warten

2.2.2 Synchronisation durch Nachrichten