
GamesLab Handbuch

KidsLab.de



2024-12-18

Übersicht

Willkommen im GamesLab	4
Deine Reise ins Spieleentwickeln	4
Level up: Achievements	6
Was ist Scratch?	8
So funktioniert's	8
Das kannst du damit machen	8
Das Beste an Scratch	8
Deine Zugangsdaten	9
Movie Director 🎬 [★]	11
Die Bühne	12
Deine Figuren	12
Die Blöcke	12
Dein Programmierbereich	12
Leg los: lass die Katze laufen!	12
Katzen Freund 😺 [★]	16
Kater Einstein vs. Evil-Staubsauger	16
Sound Master 🎵 [★★]	21
Sound-Effects	21
Cookie King 🍪 [★★★]	23
Mmh, Kekse!	23
Der Keks!	23
Die Variablen!	24
.. und so zählt dein Cookie hoch	25
Das Upgrade-System	25
Dino Runner 🦕 [★★★★]	28
Dino-Runner - der Plan!	28
... das war jetzt aber sehr schnell!	31
High-Score Master 🏆 [★★★★]	33
Teil 1: Die Basics	33

Teil 2: Pimp my Highscore!	36
Effect Wizard ⚡ [★★★★★]	37
Effekte-Spotlight: Paralax-Effekt!	37
Regenbogen-Power ☀	40
Sparkle! Lass die Funken fliegen!	40
Intro Designer 🎬 [★★★★★]	44
Der erste Eindruck zählt!	44
Teil 1: Das Intro	45
Teil 2: Game Over mit Stil	45
Extension King 🐍 [★★★★★]	46
TurboWarp (https://turbowarp.org/)	46
Scratch Addons (https://scratchaddons.com/de/)	47
BlockLive - Gemeinsam ist besser!	47
PengiunMod (https://penguinmod.com/)	47
Game Master 🎮 [★★★★★]	48
Mein Spiel ☰ Spielinformationen	48
Der GamesPreis!	51
Der GamesPreis 2024	52
Die 3 Hauptgewinner:	54
Debugging-Tipps für dein Scratch-Projekt	56
Bugs? No Problem! ☰	56
1. Laut vorlesen	56
2. In Teile zerlegen	56
3. Verlangsamten	56
4. Sound-Checkpoints	56
5. Block-Reihenfolge checken	56
6. Schleifen prüfen	57
7. Timing beachten	57
8. Mach Pausen!	57
Nutze den Telefon-Joker!	57
Danke! 🎮	59

Lizenz & Weitergabe	59
Danke an unsere Unterstützer! 	59
Klaus Tschira Stiftung gGmbH	59
Stiftung AUFWIND	60
XITASO GmbH	60

|

Willkommen im GamesLab

Deine Reise ins Spieleentwickeln

Hey!

Willkommen im GamesLab Handbuch! Hast du schon mal davon geträumt, dein eigenes Spiel zu entwickeln? Hier lernst du, wie das geht – mit Scratch, einer coolen Plattform, die Programmieren super einfach macht. Scratch funktioniert wie ein Baukasten: Du kombinierst Blöcke und bringst Figuren in Bewegung, für deine ganz eigenen Spiele, Animationen oder Geschichten.

Was dich erwartet:

- **Dein erstes Spiel:** Wir starten mit einfachen Klick-Spielen und arbeiten uns zu spannenden Jump'n'Runs vor.
- **Tolle Effekte:** Von coolen Animationen bis hin zu Sounds und Partikel-Effekten.
- **Schritt-für-Schritt-Anleitungen:** Du wirst sehen, wie einfach es ist, deine Ideen in die Tat umzusetzen.

Egal ob du schon Erfahrung hast oder ganz neu bist – hier wirst du zum Spiele-Entwickler!

Lass uns loslegen und gemeinsam richtig tolle Spiele bauen!

Gregor, Regine, Irmel, MatzE, Daniel, Johannes... und alle anderen Mentorinnen und Mentoren



Abbildung 1: Diese Mentorinnen und Mentoren begleiten euch!

Level up: Achievements

Level Up! Hier kannst du deine Game-Developer-Skills tracken und sehen, was du schon alles drauf hast! 🎮

Level 1 Einführung

- Zugangsdaten aufgeschrieben
- Gecheckt, warum das nur manchmal ne gute Idee ist
- Überlegt, ob ich schon mal Passwörter aufgeschrieben habe

Level 2 Movie Director

- Du kannst deinem Mitschüler erklären, was sich wo im Arbeitsbereich befindet
- Die Katze dreht sich fleißig im Kreis, wenn du die grüne Flagge klickst
- Sie ist animiert, wenn man auf sie klickt
- Du hast der Katze einen Bart, rote Augen oder Eselsohren gemalt
- Die Katze ist verschwunden! Und du hast sie wiedergefunden (außerhalb der Bühne)

Level 3 Katzenfreund

- Du kannst die Katze bewegen
- Die Maus landet im Magen des Katers und verschwindet
- Der Roboter ist viel zu schnell und du schaffst es trotzdem 5 Mäuse zu fressen
- Die Katze verschwindet im Off und du kannst sie nicht mehr sehen, wie krieg ich die wieder her?

Level 4 Sound Master

- Einstein macht lustige Geräusche
- Sie ändern sich, wenn er viele Mäuse gefressen hat
- Die Töne sind total nervig und dein Lehrer hat gesagt "Um Gottes Willen, macht mal den Ton aus"
- Not-Aus-Knopf für den Lehrer eingebaut

Level 5 Cookie King

- Klick-Feedback ist satisfying
- Upgrades sind balanced
- Spieler bleiben motiviert

Level 6 Dino Runner

- Dino reagiert sofort auf Sprung-Befehle
- Sprung fühlt sich natürlich und flüssig an
- Hindernisse verschwinden am linken Rand

- Game Over funktioniert
- Score wird korrekt gezählt
- Der Dino ist so frustrierend schwer, dass dein Lehrer es nicht schafft
- Du hast versehentlich einen Bug eingebaut, bei dem der Dino fliegen kann - und findest es eigentlich ganz cool

Level 7 HighScorer

- Im Rang zu "Scratcher" aufgestiegen
- Scores werden korrekt gespeichert
- Cloud-Variablen sind easy peasy für dich!
- Dein Freund hat es geschafft, deinen Highscore zu überbieten!

Level 8 Effect Wizard

- Baue die Wolken in Dino-Runner ein
- Mache eine Regenbogen-Power-Animation beim Laufen
- Erstelle eine Siegertanz wenn 20 Kakteen geschafft sind!
- Erstelle zu viele Klone, dein Computer stürzt ab!

Level 9 Intro Designer

- Läuft flüssig ohne Ruckeln
- Skip funktioniert
- Sieht auch nach 10x noch gut aus
- Füge Easter Eggs ein

Level 10 Extension King

- Mindestens eine Extension mit TurboWarp erfolgreich eingebunden
- Spiel als Programm exportiert
- Offline-Version funktioniert
- Dein Spiel läuft so smooth, dass die anderen denken, es wäre gekauft
- Du hast versehentlich so viele Extensions aktiviert, dass dein Browser streikt

Was ist Scratch?

Scratch ist wie ein digitaler **Baukasten** zum **Programmieren**. Du kannst damit ganz einfach eigene **Spiele**, **Animationen** und **Geschichten** erstellen - auch ohne **Vorkenntnisse**!



NerdY Fun-Fact Bis her wurden unglaubliche **1 Milliarde Projekte** veröffentlicht! Und jeden Tag kommen **500.000** neue Projekte dazu! Es gibt **Scratches** aus **jedem** Land unserer Welt.

So funktioniert's

- Du ziehst bunte **Blöcke** wie **Legosteine** auf deine **Arbeitsfläche**
- Die **Blöcke** passen automatisch zusammen und sagen deinen **Figuren**, was sie tun sollen
- Mit ein paar Klicks können deine **Figuren** sich **bewegen**, **Geräusche** machen oder die **Farbe ändern**
- Alles funktioniert durch einfaches **Ziehen und Ablegen** mit der **Maus**

Das kannst du damit machen

- Eigene **Spiele** entwickeln
- **Figuren** zum **Leben erwecken**
- **Geschichten** animieren
- Mit **Farben** und **Effekten** experimentieren
- **Musik** und **Töne** hinzufügen

Das Beste an Scratch

- **Scratch** ist komplett **kostenlos** und du kannst es unter scratch.mit.edu kostenlos nutzen.
- Du brauchst keine **Programmiererfahrung**
- Es macht richtig viel **Spaß**
- Du kannst sofort **loslegen** und **kreativ** werden
- Wenn etwas nicht klappt, kannst du es einfach noch mal probieren

Deine Zugangsdaten

Deine Scratch-Zugangsdaten 



NerdY Erkläromat Im Normalfall solltest du **Passwörter** nie aufschreiben. Für deinen **Scratch-Account** machen wir eine Ausnahme, weil: 1. Dies ein **Lern-Account** ist 2. Du in **Scratch** keine **persönlichen** oder **geheimen** Sachen speicherst 3. Du ansonsten das **Passwort** vergisst und nicht weiter an deinem **Spiel** arbeiten kannst! **Daher diese wichtigen Regeln:** 1. Verwende hier ein **eigenes, neues Passwort** 2. Nutze **NICHT** dein **Standard-Passwort** von anderen Accounts

Diese pragmatische Lösung hilft dir, dich auf das **Programmieren** zu konzentrieren, ohne dich um vergessene **Zugangsdaten** sorgen zu müssen! 

Meine Scratch-Zugangsdaten:

- Benutzername: _____
- Passwort: _____

Movie Director 🎬 [⭐]



NerdY Fun-Fact Wusstest du? Der erste **Animationsfilm** der Welt wurde **1908** von **Émile Cohl** erstellt und hieß “**Fantasmagorie**”. Er bestand aus über **700** einzelnen **Zeichnungen**, die nach-einander **fotografiert** wurden! 📸

Was du hier lernst

- Die **Scratch-Oberfläche** kennenlernen
- **Figuren** und **Sprites** steuern
- Erste **Animationen** erstellen
- Mit der **Bühne** und **Kostümen** arbeiten

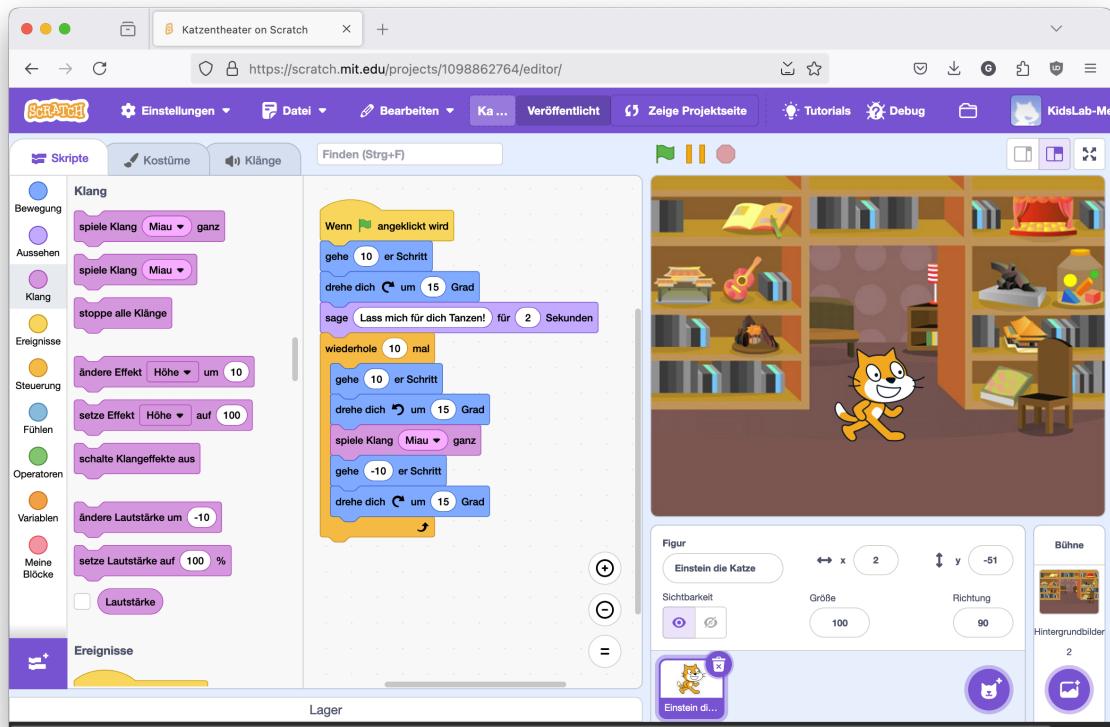


Abbildung 2: 02-Übersicht Scratch IDE

Die Bühne

- Rechts siehst du die “**Bühne**” - das ist dein **Spielfeld**
- Hier bewegen sich deine **Figuren** und hier passiert die ganze **Action**
- Die **Bühne** kannst du auch als **Vollbild** anzeigen lassen

Deine Figuren

- Rechts unten siehst du alle **Figuren** in deinem **Projekt**
- Am Anfang ist nur die **Scratch-Katze** da
- Mit dem “+”-Button kannst du neue **Figuren** hinzufügen:
 - Fertige **Figuren** aus der **Bibliothek** wählen
 - Eigene **Figuren** malen
 - **Bilder** von deinem **Computer** hochladen

Die Blöcke

- Links findest du alle **Befehlsblöcke**
- Die **Blöcke** sind nach **Farben** sortiert, so findest du sie leichter:
 - Blaue **Blöcke für Bewegung**
 - Violette **Blöcke für Aussehen**
 - Gelbe **Blöcke für Events** (was wann passieren soll)
 - Orange **Blöcke für Steuerung**

Dein Programmierbereich

- In der Mitte ist dein **Programmierbereich**
- Hierhin ziehst du die **Blöcke**
- Die **Blöcke** rasten wie **Puzzleteile** ineinander ein
- Alles was du hier zusammenbaust, bestimmt was deine **Figur** macht

Leg los: lass die Katze laufen!

1. Klicke auf die **Scratch-Katze**
2. Ziehe einen blauen “gehe 10er-Schritt”-**Block** in den **Programmierbereich**
3. Klicke auf den **Block**

4. Die **Katze** bewegt sich!



Abbildung 3: alt text

Und jetzt mach noch ein ganzes **Programm** draus!

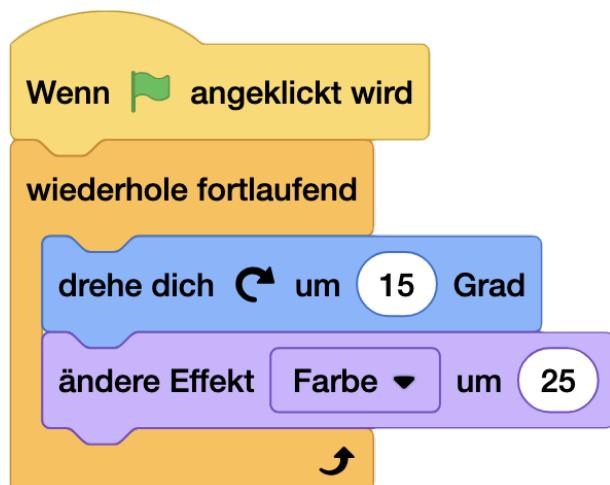


Abbildung 4: Programm für die TanzeKatze

Klicke auf die **Grüne Flagge**:



Nerdy-Erkläromat

1. **Figuren** schnell duplizieren
 - Rechtsklick auf **Figur** → **Duplizieren**
 - Spart Zeit beim Erstellen ähnlicher **Figuren**
2. **Kostüme** clever nutzen
 - **Kostüme** für verschiedene Zustände
 - **Animation** durch **Kostümwechsel**
 - **Kostüme** aus der Bibliothek mixen
3. **Ordnung** im Projekt
 - **Figuren** sinnvoll benennen
 - **Blöcke** kommentieren
 - Ähnliche **Blöcke** gruppieren



Nerdy-Side-Quest

Mach aus der **Katze** einen **Filmstar**!

1. füge den **Block** wenn auf figur geklickt und wechsle zum nächsten kostüm



Abbildung 5: alt text

.

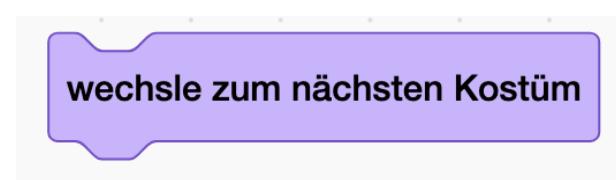


Abbildung 6: alt text

.

2. Klicke auf die **Katze**, was passiert?
3. Wechsle im **Arbeitsbereich** auf Kostüme und male etwas auf der **Katze** rum

Katzen Freund 😺 [★]



NerdY Fun-Fact

Wusstest du? Der erste digitale **Staubsauger-Roboter** wurde **2002** eingeführt. **Katzen** auf der ganzen Welt haben seitdem einen neuen **Erzfeind** - oder ein neues **Transportmittel!** ☺

Was du hier lernst

- **Steuerung mit Pfeiltasten** programmieren
- **Kollisionen** erkennen und darauf reagieren
- **Punkte** sammeln und zählen
- Mit **Zufallspositionen** arbeiten

Kater Einstein vs. Evil-Staubsauger

Kater Einstein, das klügste Haustier der Stadt, muss vor einem durchgedrehten **Saugroboter** fliehen. **Sammle Wollknäule** für **Extra-Power!**



Abbildung 7: 03-Figuren

So programmierst du es:

Figuren einrichten:

1. **Kater Einstein** (deine Pixel-Katze)
2. **Saugroboter** (runde Form, böse Augen)
3. **Quietsche-Maus** (Power-Ups)

Für Kater Einstein:

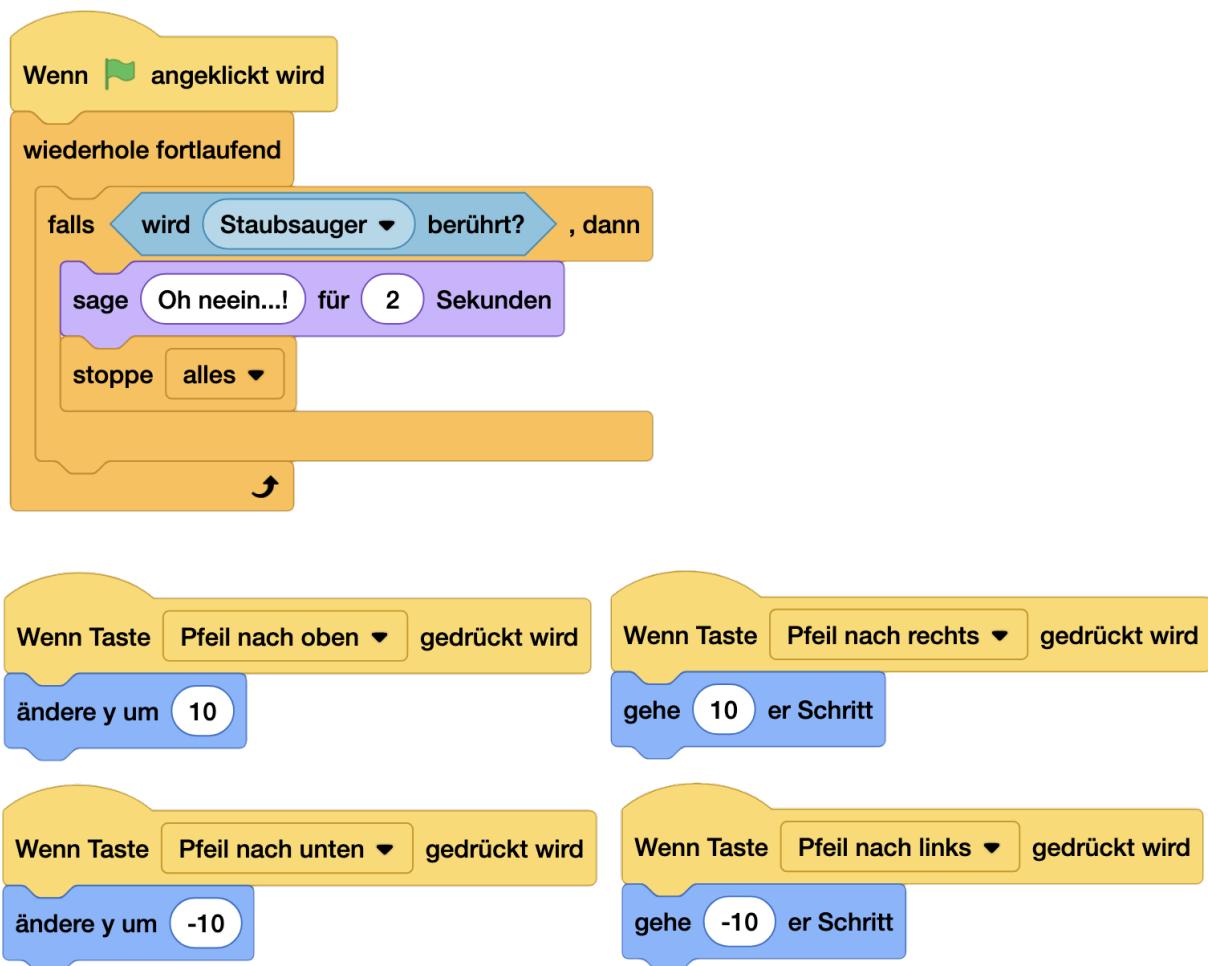


Abbildung 8: 03-Programm Einstein

Für den Saugroboter:



Abbildung 9: 03-Programm Sauger

Für die Quietsche-Maus:

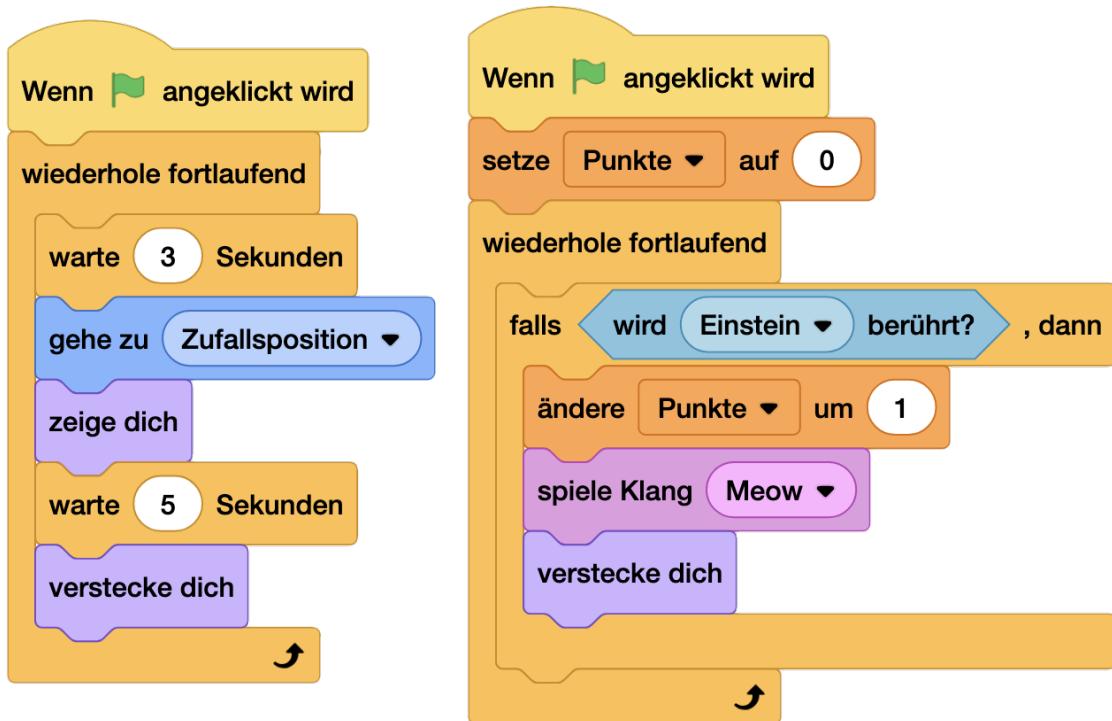


Abbildung 10: 03-Programm Maus



NerdY Side Quests 🔥

- Cleverer Staubsauger
 - Verschiedene Bewegungsmuster
 - Geschwindigkeit erhöhen mit Zeit
 - "Wütend" werden bei Punkten
- Füge einen Timer hinzu
- Lass den Staubsauger immer schneller werden

..und fertig:

Das fertige Projekt und mehr **Beispiele** und **Erklärungen** findest du hier:

<https://pad.kidslab.de/p/GamesLab-Katzenfreund> (QR Code)



nerdY Erkläromat Viele **Bilder** für dieses Handbuch haben wir mit **ChatGPT** und **Dall-E** erzeugt.
Du kannst deine **Bilder** selber malen, pixeln oder natürlich auch mit einer **KI** erzeugen!

Sound Master 🎵 [★★★]



NerdY Fun-Fact

Wusstest du? Das “Jump”-Geräusch von **Super Mario** wurde **1985** erfunden, weil der **Komponist** keine **Melodie** mehr unterbringen konnte - heute ist es eines der bekanntesten **Soundeffekte** der Welt! 🎮

Was du hier lernst

- **Soundeffekte** in dein **Spiel** einbauen
- **Töne** mit **Aktionen** synchronisieren
- Mit **Lautstärke** und **Effekten** experimentieren
- **Audio-Feedback** für **Spielereignisse**

Sound-Effects

Dirigiere deine eigene **Spiele-Symphonie!** Erschaffe dynamische **Soundeffekte**, die dein **Spiel** zum **Leben** erwecken.

Mit diesem **Block** kannst du **Töne** und **Musik** abspielen:



Abbildung 11: alt text

Deine Aufgaben:

- Füge den **Block** beim **Programm** von **Einstein** hinzu:
 - Falls wird ****Quietsche Maus**** berührt
 - Orientiere dich am **Code des Staubsaugerroboters**
 - **spiele Klang** **Miau** wenn er sich eine **Maus** schnappt.

**NerdY Pannenhilfe 🚨**

```
play sound [miau v] for [ganz] ticks
```

Obacht - es gibt 2 `play sound`-Befehle:

1. Der `play sound`-Befehl startet das **Abspielen**, aber das **Programm** läuft weiter.
2. Der `play sound ...` ganz wartet an dieser Stelle solange, bis der ganze **Klang** gespielt ist, erst dann läuft der Rest des **Programmes** weiter.

**NerdY Side Quests 🔥 - [] Füge den **Block ändere Effekt** hinzu, wenn man auf **Einstein** klickt.**

```
when green flag clicked
  [ ändere Effekt [ Höhe v um 10 ] for [ 10 ticks ] ]
```

Was passiert? Klicke ganz oft drauf!

- [] Mache eine wiederhole 10 mal um den **Ton-Effekt** - [] Bitte schau dir mal dieses **Projekt** an - es ist von **EricR**, einem der **Master-Minds** hinter **Scratch**, ich liebe es: <https://scratch.mit.edu/projects/537008383/>

Cookie King 🍪 [★★★]

Mmh, Kekse!

Wer kann dem Duft von frisch gebackenen **Keksen** widerstehen? Und auch der **Cookie-Clicker** erfreut sich großer Beliebtheit. Wie den **Keksen** gehts auch bei **Spielen** um das ganze **Erlebnis: Riechen, Sehen, Schmecken!**



NerdY Fun-Fact

Der originale **Cookie Clicker** wurde **2013** von einem französischen **Programmierer** in seiner Mittagspause entwickelt - heute haben über **4 Millionen Menschen** mehr als **1 Trillion virtuelle Kekse** gebacken! 🎉 Das originale **Spiel** findest du hier: <https://orteil.dashnet.org/cookieclicker/>

Was du hier lernst 🕵️

- Ein süchtig machendes **Idle-Game** entwickeln
- Mit **Variablen** rechnen und Werte speichern
- **Klick-Ereignisse** programmieren
- Automatische **Aktionen** durch **Schleifen**
- Ein **Upgrade-System** aufbauen
- **Spiel-Balance** gestalten

Der Keks!

Lege dir eine neue **Figur** an - der **Keks!**

Du kannst:

- Eine bestehende **Figur** wählen
- eine neue Malen
- oder eine mit **ChatGPT** erzeugen und die Figur hochladen



Abbildung 12: Figur-neu

Die Variablen!



Nerdy-Erkläromat: Var-I-Was?

Eine **Variable** ist wie eine beschriftete **Box**, in der du Dinge aufbewahren kannst. Stell dir vor, du hast mehrere **Kisten**: 1. Eine **Box** mit der Aufschrift "**Punkte**" - hier speicherst du deinen **Spielstand**

2. Eine **Box** mit "**Leben**" - hier merkst du dir, wie viele **Leben** noch übrig sind
3. Eine **Box** "**Name**" - hier steht der **Name** deines **Charakters** drin Das Besondere an diesen **Boxen** ist:
4. Du kannst jederzeit reinschauen, was drin ist
5. Du kannst den **Inhalt** ändern
6. Du kannst mit dem Inhalt rechnen (bei Zahlen)
7. Du kannst den Inhalt für verschiedene Dinge im Spiel verwenden

.. und so zählt dein Cookie hoch

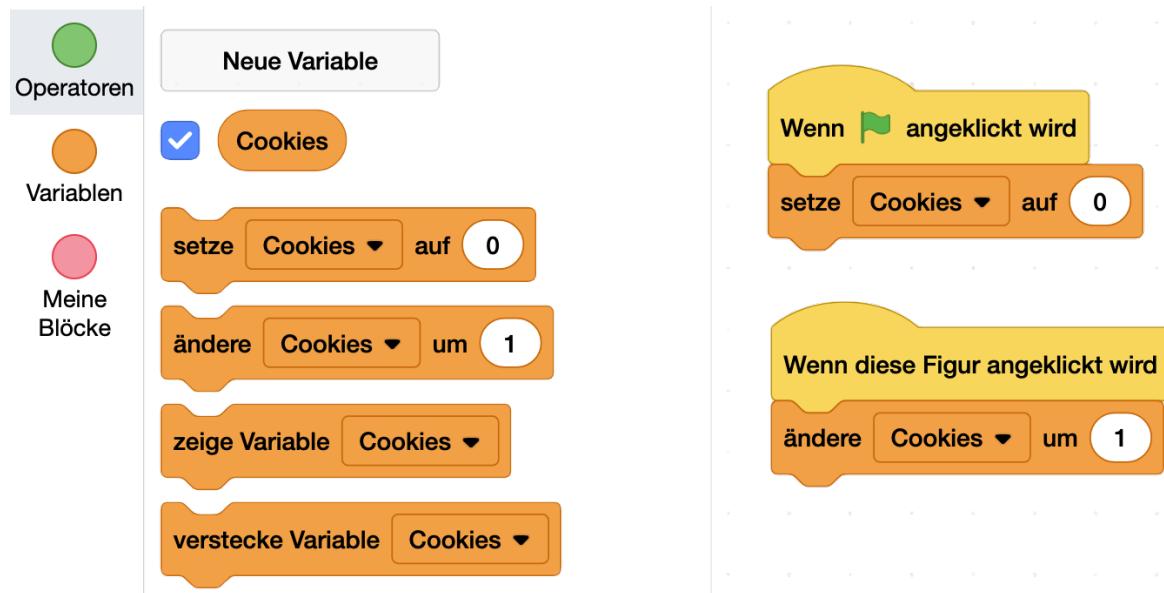


Abbildung 13: Programm für den Cookie

- Erstelle eine neue Variable: clicke auf Neue Variable
- Nenne die Variable “Cookies”
- Beim Start des Programms - setze Cookies auf 0 - das Spiel geht ja immer von 0 los.
- Wenn der Cookie angeklickt wird - ändere Cookies um 1 - das erhöht die Variable Cookies jeweils um eins.

Das Upgrade-System

Jetzt fehlen nur noch die Upgrades - wie genau Du sie programmierst, findest du heraus, wenn du die QR-Code scannst!

Back-Oma

Bäckerei

Auto-Klicker

Erklärung



<https://pad.kids-lab.de/p/GamesLab-CookieMaster> (QR Code)



NerdY Side Quests: Knusprige Animation!

Ein Spiel lebt ja auch davon, dass es sich gut “anfühlt” - also es beim Klicken Effekte gibt!

Probiere diesen Code mal bei deinem Spiel aus!

Du musst die Werte bei Größe entsprechend deinem Cookie anpassen, also die Zahlen: 150, 120 und 180.

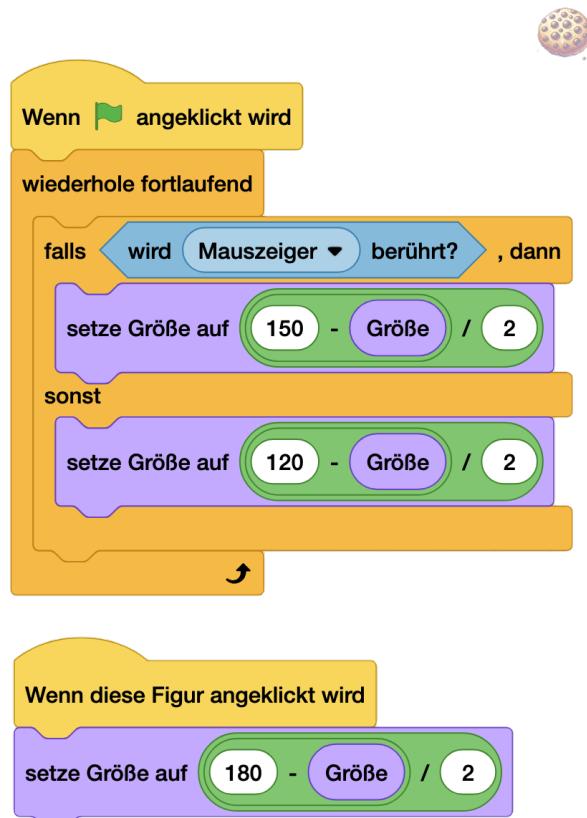


Abbildung 14: 05-cookie-effect

Remember: Der beste Cookie Clicker ist der, den man nicht mehr aufhören kann zu spielen! 🍪🌟

Dino Runner [★★★]

SPRING! Programmiere den Klassiker nach und lass deinen **Dino** über alle **Hindernisse** hüpfen.



nerdY Fun Facts:

Der Chrome **Dino** wurde **2014** von einem Chrome-Entwickler als Zeitvertreib entwickelt - aber er hat sich so einen Kult-Status erarbeitet. Du kannst das **Spiel** im Chrome Browser spielen, wenn du keine **Internetverbindung** hast. Du kannst es aufrufen, wenn du diesen Befehl in die Adresszeile eingibst: `chrome://dino` Warum kommt da ein **T-Rex-Dino**, wenn man kein **Internet** hat? Weil man dann informationstechnisch in **Prähistorischen-Zeiten** ist!

Was du hier lernst 

- Ein endlos laufendes **Jump'n'Run** entwickeln
- Mit **Clone-Objekten** arbeiten
- **Kollisionen** erkennen und reagieren
- **Gravitation** und **Sprungmechanik** programmieren

Dino-Runner - der Plan!

Am Anfang ist es immer gut, sich einen **Plan** zu machen: ob es ein **Storyboard** für dein **Spiel** ist, oder eben einen **Plan**, wie man was programmiert:

1. **Boden** und **Hauptfigur** erstellen
2. Wir bringen dem **Dino** das hüpfen bei
3. Bewegung des **Bodens**
4. Der **Kaktus** kommt!
5. **Game Over**



Abbildung 15: DinoRunner - der Plan

1. Hintergrund und Hauptfigur

- **Hauptfigur** ist bei mir die **Katze** - du kannst natürlich jede beliebige **Figur** nehmen!
- Der **Boden** ist auch eine **Figur**, so erstellst du ihn:
 - erstelle eine neue **Figur**
 - nenne sie **Boden**
 - gleichmäßig - waagerechte linien
 - keine **Steine** oder so - das kommt im nächsten Schritt

2. Wie bringt man einem Pixel-Dino das Hüpfen bei?



Abbildung 16: Huepfen - so gehts

Sprungkraft!

- Wir verwenden eine **Variable** namens “Sprungkraft” - das bestimmt, wie schnell die **Katze** nach oben fliegt
- Wenn die **Katze** los springt, hat sie erst mal volle “Sprungkraft”: sie bewegt sich mit 10 Punkten nach oben.
- Bei jedem Durchlauf wird die **Sprungkraft** um “1” kleiner gemacht, sie fliegt also immer langsamer nach oben
- Irgendwann wird die **Sprungkraft** “negativ” - dann fliegt die **Katze** wieder runter
- Sobald sie den **Boden** berührt, ist schluss mit Fliegen und wir setzen die **Sprungkraft** wieder auf 0

3. Boden, bewege dich!

Wir machen jetzt einen Trick, statt den **Boden** zu Bewegen, machen wir **Steine**, die sich auf der **Bühne** bewegen - dann siehts so aus, als ob sich alles bewegt.

So gehts:

1. Erstelle eine **Steinfigur**
2. Die **Steine** entstehen als “Klone” - zufällig
3. Wenn die **Klone** entstehen, bewege sie nach links
4. Wenn sie den **Rand** (links) berühren, sollen sie verschwinden



NerdY Erkläromat: Was sind Klone

Klone in **Scratch** sind wie **Kopien** von **Figuren**, die während des **Spiels** automatisch erstellt und wieder gelöscht werden können. Das ist besonders nützlich für sich wiederholende **Elemente** wie:

- **Regentropfen**
- **Münzen** zum Einsammeln
- **Projektile/Schüsse**
- **Partikeleffekte**

Wichtig:

- Das Original ist die “**Fabrik**” die **Klone** erzeugt
- **Klone** führen eigene **Befehle** aus
- **Klone** sollten gelöscht werden wenn nicht mehr benötigt
- Nicht zu viele **Klone** gleichzeitig (max ~300) sonst wird das **Spiel** langsam

Mit **Klonen** sparst du dir viel Arbeit, da du nicht jedes **Element** einzeln erstellen musst. Sie sind perfekt für sich wiederholende **Spielelemente**: beim **Dino-Runner** sind das die **Kakteen** und **Steine**...

4. Kaktus kommt!

Jetzt fehlt für die erste Version des **Spiels** nur noch ein **Kaktus**, der **Gegner** für unseren **Dino**!

Der **Kaktus** funktioniert ganz ähnlich wie die **Steine**, darum “dupliziere” den **Stein** und pass ihn an!

So gehts:

1. **Stein** duplizieren (wir sind faul) und umbenennen
2. **Kostüm** anpassen - male den **Kaktus**!
3. **Position** und **Größe** anpassen.

5. Game Over!

Die **Katze** soll schauen, ob sie den **Kaktus** berührt... wenn ja: Game Over!

... das war jetzt aber sehr schnell!

Kein Problem - hier noch mal alles in langsam, schritt für schritt:

<https://pad.kidslab.de/p/GamesLab-Dino> (QR Code)

High-Score Master 🏆 [★★★]

Für die Ewigkeit! Speichere die besten Leistungen



NerdY Fun-Fact

Der erste dokumentierte **High-Score** der Videospielgeschichte wurde **1976** im Spiel **Sea Wolf** aufgestellt. Damals musste man noch ein **Foto** vom Bildschirm machen, um seinen **Score** zu beweisen! 📸

Was du hier lernst 🎯

- Ein **Highscore-System** entwickeln
- Mit **Cloud-Variablen** arbeiten
- **Daten** dauerhaft speichern
- Eine **Bestenliste** programmieren
- Mit dem **Scratch Community-System** interagieren

Teil 1: Die Basics

Punkte sammeln

Punkte haben wir bis her ja auch schon gesammelt - dafür nimmst du eine **Variable**!

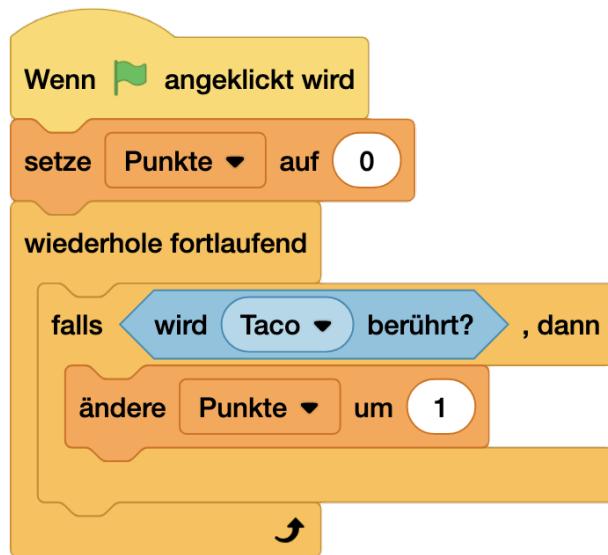


Abbildung 17: Programm Punkte

- Am Anfang: **setze Punkte auf 0**
- Wenn die **Katze** den leckeren Tako berührt...
- ... dann **ändere Punkte um 1**

Punkte abspeichern

Jedesmal, wenn du dein **Spiel** neu lädst, sind ja alle **Variablen** zurückgesetzt. Zum **Speichern** gibt es eine erweiterte **Variablen-Art**, die Cloud-Variable.



Abbildung 18: Cloud Variablen

Einfach prüfen, ob der aktuelle **Punktestand** größer ist, als der **Highscore** - wenn ja: denn ist ein neuer **Highscore** erreicht!

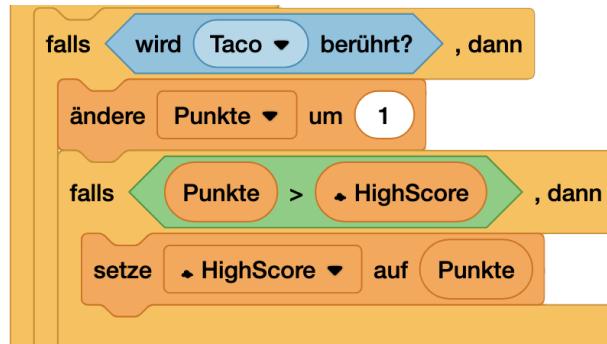


Abbildung 19: Punkte Highscore



NerdY Pannenhilfe 🚨

Um **Cloud-Variablen** benutzen zu können, musst du ein “Verifizierter” **Scratcher** sein. Mit einem neuen Konto bist du erst mal nur ein **NPC** - du musst dir den Status als echter **Scratcher** erst verdienen. Dazu musst du:

- aktiv auf **Scratch** sein
- **Projekte** veröffentlichen
- Andere **Projekte** kommentieren und liken

Das geschieht nach frühestens zwei Wochen durch einen Link (*Become a Scratcher*) auf der Profilseite hinter dem Profilbild.

Warum? Mit **Cloud-Variablen** kann man Blödsinn anstellen. Mehr Infos findest du hier: <https://de.scratch-wiki.info/wiki/R%C3%A4nge>

Teil 2: Pimp my Highscore!

Aber da geht noch mehr:

- **Top 5 Liste**
- **Namenseingabe**
- **Datum & Uhrzeit**

Und das findest du alles hier: <https://pad.kidslab.de/p/GamesLab-Highscore> (QR Code)

Effect Wizard ⚡ [★★★★★]

Abrakadabra! Zaubere magische **Effekte** in dein Spiel und lass es in neuem Glanz erstrahlen.



NerdY Fun-Fact

Die berühmten **Partikeleffekte** in Mario Kart entstanden ursprünglich durch einen **Bug** - die Entwickler fanden sie aber so cool, dass sie sie im Spiel gelassen haben! 🎉⚡

Was du hier lernst 📝

- **Visuelle Effekte** in deine Spiele einbauen
- **Partikel-Systeme** mit Klonen erstellen
- Mit **Bewegungs-** und **Grafikeffekten** arbeiten
- **Animationen** und **Übergänge** programmieren

Effekte-Spotlight: Paralax-Effekt!

Schnapp dir dein **Dino-Spiel** - wir pimpen es mal ein bisschen auf! Mit dem **Paralax-Effekt**!

Wie funktioniert das?

Dinge im Hintergrund:

- bewegen sich **langsamer**
- sind **kleiner**

Dinge weiter vorne:

- bewegen sich **schneller**
- sind **größer**

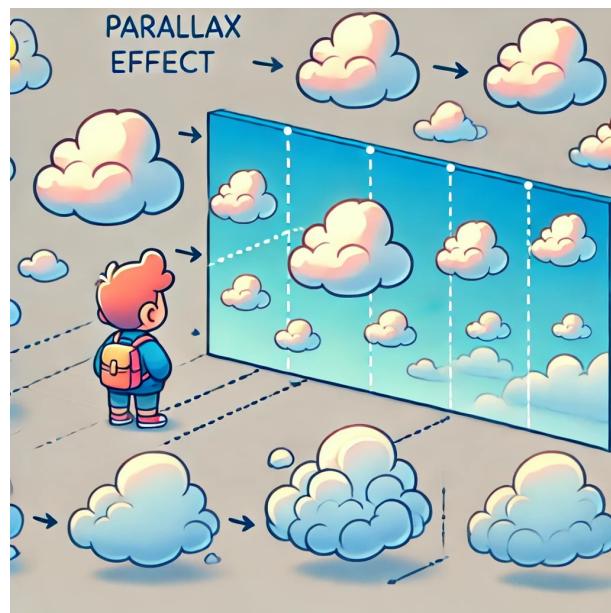


Abbildung 20: alt text

So gehts:

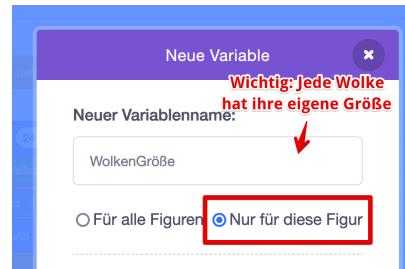
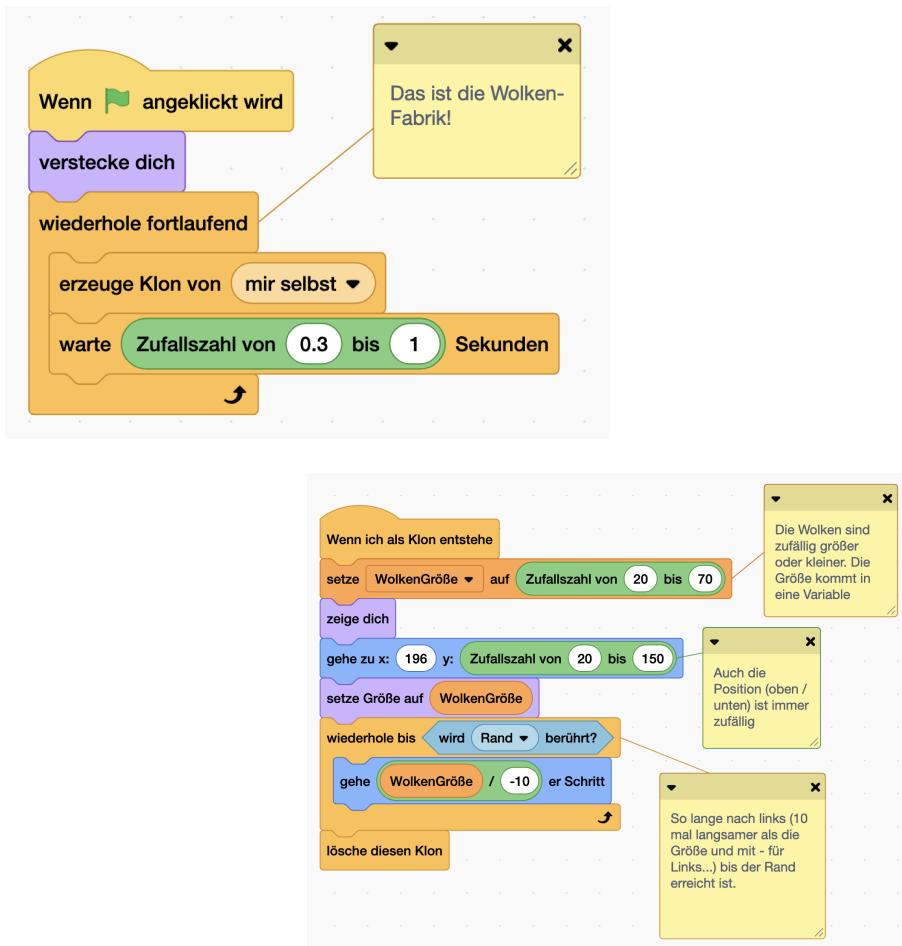


Abbildung 21: Variable erstellen

**Abbildung 22:** alt text(Beispielprojekt - QR Code? <https://scratch.mit.edu/projects/1103943291/>)

Regenbogen-Power 🌈

Mit diesem **Effekt** kannst du einer **Figur** einen Regenbogen-Schweif verleihen, wie ein Komet!

Eignet sich super für **Super-Power**, **Power-Ups** oder **Unverwundbarkeit**.

Und wieder sind die **Klonen** am Start!

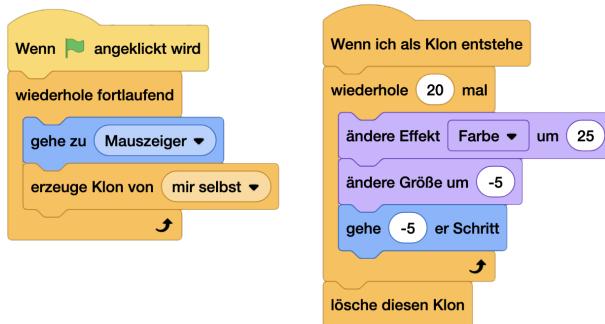


Abbildung 23: alt text

Beispiel: <https://scratch.mit.edu/projects/1103963666/>

Sparkle! Lass die Funken fliegen!

Noch ein kleiner feiner **Effekt** zum Abschluss - eignet sich hervorragend als **Game-Over** oder **Gewonnen-Effekt!**

Ach ja, wieder mal arbeiten wir mit **Klonen**! Nur gut, dass wir genug **Klonen** haben :-)

- Mach dir eine neue **Figur** - die werden dann zu deinen **Funken** oder **Partikeln**.

Die **Klon-Muddi**

Als erstes wieder unsere **Fabrik**:

- wenn der **Effekt** getriggert wird (bei mir mit der Leertaste)
- Geh zur **Figur** oder Stelle, wie es funkelt
- erzeug ganz viele **Klonen**...



Abbildung 24: 08-Sparkle1

... und die **Kids!**

- Erstelle 2 lokale **Variablen**:
 - SpeedX - wie schnell geht's links und rechts.
 - SpeedY - hoch und runter!
- SpeedX und SpeedY kriegen zufällige Werte - es soll ja immer ein bisschen anders aussehen.
- Die Y-Bewegung (hoch und runter) kennst du schon vom **Dino** - erst geht's hoch, aber wir zählen die Bewegung immer weiter runter...
- ... und am Rand ist Schluss mit dem fröhlichen **Klon-Leben**: lösche diesen Klon

**Nerdy Erkläromat:** Was sind lokale und globale Variablen?

*Sind das vielleicht **Variablen** aus deiner Region, die dann extra frisch? Weit gefehlt!*

Wenn du eine **Variable** erstellst, können alle **Figuren** darauf zugreifen und auch den Inhalt ändern. So kannst du zum Beispiel super Infos zwischen **Figuren** teilen. Programmierer nennen solche **Variablen** global.

Für uns bestimmen die beiden **Variablen** SpeedX und SpeedY die Bewegung jedes einzelnen Funkens.

Jeder soll sich anders bewegen, der eine links, der andere rechts, schneller und langsamer. Hier brauchen wir lokale **Variablen**.

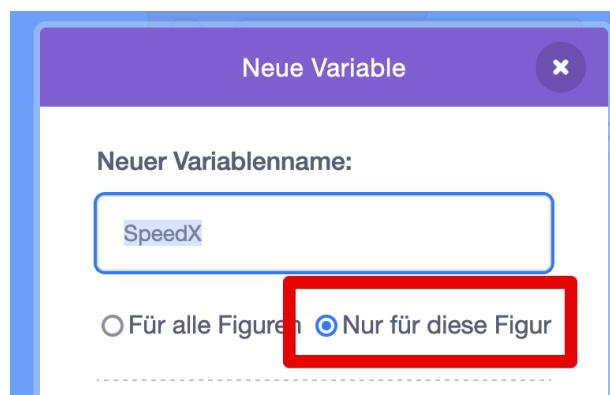


Abbildung 25: LokaleVariable

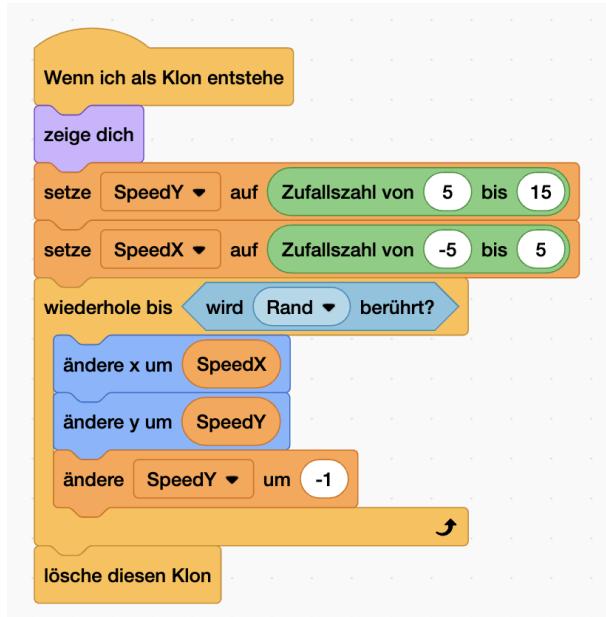


Abbildung 26: 08-Sparkle2

Mögen deine **Effekte** magisch sein! 🌟🔮

Remember: Mit großer **Effekt-Power** kommt große **Verantwortung!** 🚨

Weitere **Effekte** und wie sie funktionieren findest du hier: QR Code: <https://pad.kidslab.de/p/GamesLab-Effekte#>

Intro Designer 🎥 [★★★★★]

Der erste Eindruck zählt! Kreiere ein packendes **Intro** und einen dramatischen **Game-Over-Screen**.

Der erste Eindruck zählt!



NerdY Fun-Fact

Das erste **Videospiel-Intro** überhaupt war eine simple **Animation** im Spiel “Space Invaders” (1978) - die außerirdischen Invasoren marschierten von oben nach unten ins Bild! **Space Invader** ist so berühmt, dass es sogar sein eigenes **Emoji** hat



Was du hier lernst

- Ein packendes **Intro** für dein **Spiel** erstellen
- Dramatische **Game-Over Screens** designen
- **Text-Effekte** und **Animationen**
- **Sound** und **Musik** wirkungsvoll einsetzen
- Wie du mit Nachrichten zwischen den **Figuren** kommunizieren kannst



Nerdy-Erkläromat: Nachrichten senden und empfangen

Für das **Intro**, **Game-Over** etc. müssen wir die einzelnen **Figuren** im **Scratch** miteinander “verdrahten”. Wenn zum Beispiel der **Dino** mit dem **Kaktus** zusammenstößt, müssen auch die **Steine** und die **Wolken** wissen, dass jetzt “Game Over” ist.

Mit den Nachrichten können die **Figuren** sozusagen Gedankenübertragen: Das nutzen wir auch, um den **Figuren** unseres **Spiels** mitzuteilen, dass jetzt Zeit für ihren Auftritt ist. Oder eben Schluss: **Game Over!**

Programm Sender



Programm Empfänger



Teil 1: Das Intro

So fesselst du deine Spieler von Anfang an

Der große Auftritt: Titel-Animation

- Trick: Lass deinen **Spieltitel** nicht einfach erscheinen - bring ihn in Bewegung!
- arbeite mit **Farb-** und **Größenänderungen**

Teil 2: Game Over mit Stil

Aus Spielende mach Spielerlebnis

1. Der dramatische Abgang

- **Zeitlupe-Effekt** beim letzten **Leben**
- **Screen-Shake** beim **Game Over**
- Dramatische **Musik** oder **Sound-Effekte**

2. Der Neustart-Hook

- Motivierende **Nachricht**
- “Noch ein Versuch?”-**Button** mit **Animation**
- Tipp: Ein schneller **Neustart** hält die **Spieler** bei der Stange!

Wie das alles geht, erfährst Du hier:

<https://pad.kidslab.de/p/GamesLab-Intro> (QR Code)

Extension King [★★★★★]

Bereit für die geheimen Superkräfte von **Scratch**? Mit **TurboWarp**, **Erweiterungen** und **Add-ons** holst du noch mehr Power aus deinen **Spielen** raus.



NerdY Fun-Facts

Wusstest du? Der **TurboWarp Compiler** wurde von einem Teenager entwickelt, der **Scratch** schneller machen wollte - heute nutzen Millionen von **Scratchern** seine Erfindung!

Was du hier lernst

- **Scratch** mit **TurboWarp** beschleunigen
- Coole **Erweiterungen** nutzen
- Deine **Spiele** als echte **Programme** exportieren
- Mit anderen zusammen **programmieren**

TurboWarp (<https://turbowarp.org/>)

TurboWarp ist wie ein Power-Booster für deine **Scratch-Projekte**! Es ist eine spezielle Version von **Scratch**, die deine **Spiele** und **Animationen** bis zu 10x schneller macht.

Extensions (Erweiterungen):

TurboWarp hat eine ganze Menge zusätzlicher **Erweiterungen**, die du im normalen **Scratch** nicht findest. Das ist wie ein riesiger Werkzeugkasten mit Extra-Werkzeugen! Damit kannst du zum Beispiel:

- Neue **Effekte** einbauen
- Bessere **Physik** für deine **Spiele** nutzen
- **3D-Grafiken** erstellen
- Und vieles mehr!

Der TurboWarp Packager:

Das ist wie ein magischer Verpackungsroboter für deine **Projekte**! Er verwandelt dein **Scratch-Projekt** in eine richtige, eigenständige **App** oder **Programm**.

Scratch Addons (<https://scratchaddons.com/de/>)

Die **Browser-Erweiterung** von **Griftpatch** macht **Scratch** besser und praktischer. Sie bringt viele nützliche Features:

- Besserer **Code-Editor**
- Einfachere **Code-Bearbeitung** und **Fehlersuche**
- Praktische Extras wie **Vollbildmodus** und **Gamepad-Unterstützung**
- Die **Erweiterung** findest du im **Browser-Store** (Chrome, Firefox etc.) unter dem Namen "Scratch Addons".

BlockLive - Gemeinsam ist besser!

Die "Blocklive" **Extension** ist ein cooles Tool für **Teamarbeit** in **Scratch**!

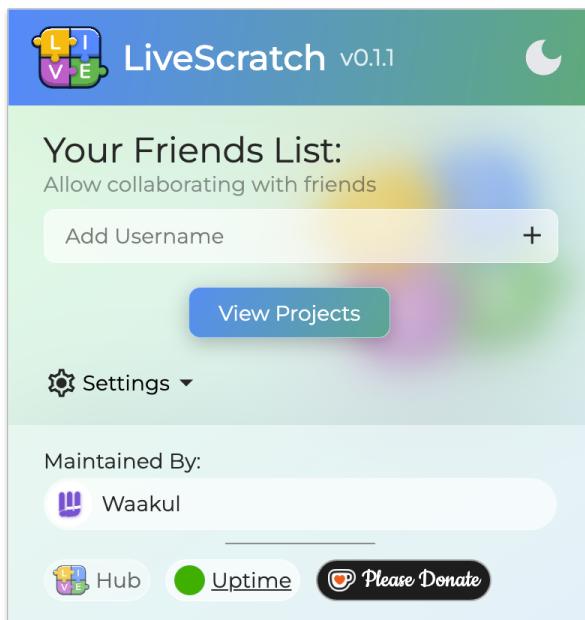


Abbildung 27: Zusammen Scratch benutzen

Installation und Infos: <https://pad.kidslab.de/p/GamesLab-LiveScratch> (QR code)

PengiumMod (<https://penguinmod.com/>)

PengiumMod ist ein aufgemotztes **Scratch**, es verbindet **TurboWarp** mit zusätzlichen Features!

Game Master 🎮 [★★★★★]

Das ultimative Ziel! Erschaffe dein Meisterwerk mit allem drum und dran. Du bist jetzt ein echter Game Developer!

Mein Spiel 🎮 Spielinformationen

Name deines Spiels:

Genre: (z.B. Jump'n'Run, Clicker, Quiz, etc.)

Was ist die Story in deinem Spiel:**Mein Spiel erklärt in einem Satz:**

2. Spielmechanik

Der Spieler kann:

- __
- __
- __

Steuerung:

Mit diesen Tasten / Maus etc. kann man im Spiel laufen, springen, schießen etc.:

• _____

Punktesystem:

So bekommt man Punkte: _____

So verliert man: _____

3. Die Hauptfigur(en)

So heißt meine Hauptfigur / das ist meine Hauptfigur:

Eine Zeichnung oder Beschreibung meiner Hauptfigur(en):

[Platz zum Zeichnen]

Besondere Fähigkeiten meiner Figur(en):

- _____
- _____

4. Level

Schwierigkeitsgrad meines Spiels: Leicht Mittel Schwer

Das macht mein Spiel besonders spannend:

- _____
- _____
- _____

Level/Abschnitte:

1. _____
2. _____
3. _____

5. Grafik & Sound

In diesem Art-Style ist mein Spiel gestaltet: Pixel Comic Realistisch Anders: _____

Diese Sounds/Musik kommen in meinem Spiel vor:

- Hintergrundmusik: _____
- Effekte für: _____
- Sonstiges: _____

6. Extra-Features

Nice-to-have (Falls noch Zeit ist):

- __
- __
- __

PRO-TIPPS

- Fang klein an! Du kannst später immer noch Features hinzufügen
- Teste früh und oft
- Lass andere dein Spiel probespielen
- Speichere regelmäßig Versionen deines Spiels
- Dokumentiere deinen Code mit Kommentaren
-

Der GamesPreis!

Reiche dein Spiel beim GamesPreis ein und gewinne mega krasse Preise!

- Preise
 - Nintendo Switch OLED
 - Meta-Quest 3
 - 3D-Drucker "Bambulab mini"
- **Checkliste zur Einreichung:**
 - Projekt fertiggestellt, veröffentlicht und getestet?
 - Alle Spielbeschreibungen hinzugefügt?
 - Kontaktinformationen angegeben?

Dann nichts wie los:

<https://kidslab.de/gamespreis/> (QR Code)

Der GamesPreis 2024

Verleihung des 1. Augsburger GamesPreis im Kleinen Goldenen Saal

Augsburg, 24. April 2024

Im Kleinen Goldenen Saal in Augsburg ist die Aufregung unter den jungen Augsburger Nachwuchs-Programmierern sowie deren Familien und Freunden deutlich zu spüren. Denn es geht um nichts Geringeres als um die Verleihung des 1. Augsburger Games-Preises, der von der gemeinnützigen KidsLab GmbH ins Leben gerufen wurde.

Die Games-Preis-Gala ist dabei der Höhepunkt des Augsburger GamesLab, dem kreativen Spiele-Studio, das Anfang des Jahres in der Augsburger "Zwischenzeit" stattgefunden hat (die AZ berichtete). Kinder und Jugendliche wurden dort kostenlos in die Welt der Spieleprogrammierung mit Scratch eingeführt, um fortan eigene Spielideen verwirklichen zu können.

Und das, was die jungen Bewerber des GamesPreis an selbst entwickelten Spielen eingereicht hatten, beeindruckte die Geschäftsführer der KidsLab gGmbH Gregor Walter und Regine Scheyer sowie die dreiköpfige Jury enorm. Auch wenn es schwerfiel, mussten drei Hauptgewinner gefunden werden. Ausgezeichnet für besonders herausragende Leistungen im Bereich Spieleanthropologie und -design wurden Johannes Kistler mit dem 1. Platz für sein Spiel "Gravity Glide", Dominik Gößler mit dem 2. Platz ("Water Facility") und Tonia Krüger mit dem 3. Platz ("Immerson 2009").

Die sichtlich stolzen Gewinner freuten sich über ihre Pokale und die von AUFWIND, die Kinder und Jugendstiftung der Stadtsparkasse Augsburg, gespendeten, hochwertigen Preise (Steam Deck OLED, Meta Quest 2 VR-Brille sowie Wacom Grafik-Tablett). Aufgrund der durchweg positiven Resonanz steht für die Macher des GamesLab bereits jetzt fest, dass das Projekt nächstes Jahr in die 2. Runde gehen wird.



Abbildung 28: GamesPreisGewinner

Die 3 Hauptgewinner:

1. Platz - Gravity Glide

In diesem fesselnden Videospiel schlüpft ihr in die Rolle eines UFO-Piloten und tretet eine epische Reise an, in der ihr durch die Manipulation von Planeten und ihrer Gravitation euer Ziel erreichen müsst. Doch das ist noch nicht alles! Strategisch platzierte Planeten und geschickte Nutzung ihrer Gravitationskräfte sind der Schlüssel, um euer UFO sicher durch jedes Level zu manövrieren. Herzlichen Glückwunsch und weiterhin viel Erfolg auf euren Reisen durch die unendlichen Weiten des Spieluniversums!



Abbildung 29: Gravity Glide

2. Platz - Water Facility

Lasst uns also einen tosenden Applaus für dieses mutige Axolotl und sein inspirierendes Abenteuer entfachen! Möge euer Kampfgeist euch stets leiten und euch den Weg zum Sieg ebnen.

Auf zu neuen Herausforderungen und möge die Flamme der Hoffnung niemals erlöschen!

<https://scratch.mit.edu/projects/966666194> (QR Code)



Abbildung 30: Water Facility

3. Platz - Immersion 2009

Taucht ein in die düstere Welt von 'Immersion 2009', einem visuellen Roman und Thriller, der euch auf eine fesselnde Reise durch Intrigen und Mysterien entführt. Begibt euch auf die Suche nach der Wahrheit und erlebt ein Abenteuer, das euch noch lange in Erinnerung bleiben wird!"

<https://scratch.mit.edu/projects/964668818>



Abbildung 31: Immersion 2009

Debugging-Tipps für dein Scratch-Projekt

Bugs? No Problem! 

Hey Gamedev! Dein Spiel macht nicht das, was es soll? Die **Katze** dreht sich in die falsche Richtung, der **Punktezähler** spinnt oder dein **Dino** schwebt plötzlich im Weltraum? Willkommen im Club!

Jeder - wirklich **JEDER** - Programmierer kämpft mit kleinen und großen **Fehlern** im Code. Das ist völlig normal und gehört zum **Spieleentwickeln** dazu wie Pommes zu Ketchup! Die gute Nachricht: Mit ein paar coolen **Tricks** und Kniffen findest du fast jeden **Bug**.

In diesem Kapitel zeige ich dir, wie du deinen **Code** auf Vordermann bringst und die häufigsten **Probleme** ganz easy löst. Und das Beste: Mit jeder **Fehlersuche** lernst du etwas Neues und wirst ein besserer Programmierer!

Also: Lass uns auf **Bug-Jagd** gehen! 

1. Laut vorlesen

Lies deinen **Code** laut vor und denk dabei wie ein **Computer**. Sind alle **Schritte** wirklich im **Code**? Sind die **Anweisungen** klar?

2. In Teile zerlegen

Teile große **Programme** in kleine Häppchen auf. Teste jedes **Teil** einzeln und füge sie dann wieder zusammen.

3. Verlangsamen

Füge “**warte**”-Blöcke ein, um zu sehen, was genau passiert. Entferne sie wieder, wenn alles läuft.

4. Sound-Checkpoints

Bau **Sounds** als Kontrollpunkte ein. Kein **Sound** = Fehler davor, **Sound** = Fehler danach.

5. Block-Reihenfolge checken

- Was muss zuerst passieren?
- Was kommt danach?
- Muss etwas zurückgesetzt werden?

6. Schleifen prüfen

Checke deine “**wiederhole**”- und “**fortlaufend**”-Blöcke:

- Sollen wirklich alle **Blöcke** in der **Schleife** sein?
- Fehlt ein “**warte**”-Block?
- Braucht es überhaupt eine **Schleife**?

7. Timing beachten

Wenn mehrere Dinge gleichzeitig passieren sollen, kann es **Chaos** geben. Baue kleine **Pausen** oder **Klick-Events** ein.

8. Mach Pausen!

Manchmal hilft es, einfach mal 5 Minuten vom **Computer** wegzugehen. Mit frischem **Kopf** sieht man **Fehler** oft sofort!

Remember: Jeder Programmierer macht **Fehler** - das Geheimnis ist, sie zu finden! 🎉🌟

Nutze den Telefon-Joker!

Alles ausprobiert, aber es geht immer noch nicht?

- **Veröffentliche** dein Projekt
- Schicke uns die **URL** des Projekts
- **Foto** reicht auch

Entweder per **Mail**:

team@kidslab.de

Oder per **WhatsApp**:



Abbildung 32: whatsapp

Danke! 🎮

Wow, du hast es geschafft! Du bist jetzt offiziell ein Game Developer und hast alle Level des GamesLab-Handbuchs gemeistert. Das ist echt cool! Ein großes Dankeschön! 🌟 Dieses Handbuch und das GamesLab wären nicht möglich ohne die Unterstützung von vielen engagierten Menschen und Organisationen:

- An alle Sponsoren, die an die Kraft von Coding und Kreativität glauben
- An die Schulen und Lehrer, die uns ihre Türen öffnen
- An das GamesLab-Team, das unermüdlich neue Ideen entwickelt
- An alle Mentoren, die ihr Wissen teilen
- Und natürlich an **dich**, weil du mit deiner Begeisterung das GamesLab lebendig machst!

Lizenz & Weitergabe

Lizenz CC BY 4.0 KidsLab

Danke an unsere Unterstützer! 🌟

Ohne die Hilfe von vielen tollen Menschen und Organisationen wäre das GamesLab nicht möglich. Ein besonders großes Dankeschön geht an:

Klaus Tschira Stiftung gGmbH

**Klaus Tschira
Stiftung**

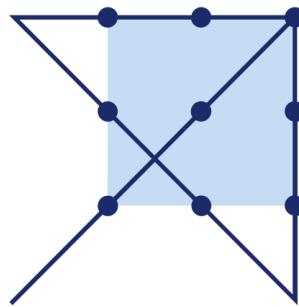


Abbildung 33: kts

Stiftung AUFWIND



Abbildung 34: aufwind

XITASO GmbH



Abbildung 35: xitaso

Dieses Handbuch wurde mit ❤ erstellt von deinem KidsLab-Team Version 1.0 - 2024