Werkzeuge für den Informatikunterricht

---- Scratch 3 ----

1. Kurzvorstellung

„Scratch 3“ (online verwendbar unter https://scratch.mit.edu/, offline als Desktop-Anwendung) soll Neueinsteiger – besonders Kinder und Jugendliche – mit den Grundkonzepten der Programmierung vertraut machen. Im Fokus steht dabei die kreative und explorative Erstellung eigener Spiele und Multimedia-Anwendungen. Ergebnisse können zusätzlich in einer internationalen Online-Community abgespielt, diskutiert und weiterentwickelt werden.

Der Name „Scratch“ leitet sich von der DJ-Scratchtechnik ab. Vorbild ist die leichte Wiederverwendbarkeit von Versatzstücken. In Scratch können alle interaktiven Objekte, Grafiken und Töne leicht von einem in ein anderes Scratch-Projekt übertragen und dort neu kombiniert werden. „Scratch“ wurde erstmals 2007 veröffentlicht, steht komplett in Deutsch zur Verfügung und besitzt eine sehr große Community.

Das aktuelle „Scratch 3“ basiert auf JavaScript.

2. Einordnung in die Lehrpläne

„Scratch 3“ eignet sich grundsätzlich für alle Lernbereiche, die sich mit „Algorithmen“, „Programmstrukturen“, „Computerspielen“ oder den Begriffen „Objekt – Attribut - Methode“ auseinandersetzen. Es ist ebenso für Arbeit in Projekten anwendbar. „Scratch 3“ unterstützt dabei einen schülergerechten, kreativen, anwendungsorientierten, konstruktivistischen und produktionsorientierten Ansatz.

Schulform	**Lernbereich**	**Bemerkung**
Oberschule	**Klasse 8, LB 2:** „Informationen verarbeiten: Modell – Algorithmus – Lösung“	Kennen grundlegender Programmstrukturen (Folge, Verzweigung, Wiederholung), Kennen des Problemlöseprozesses, Übertragen der Kenntnisse zum Problemlöseprozess auf selbstständiges Lösen einfacher Probleme und kritisches Bewerten der Resultate
	Klasse 8, LB 1: „Informationen repräsentieren: Klassen und Objekte“	Kennen von Klassen (in der Informatik), Beherrschen der Zuordnung zwischen Objekten und Klassen
	Klasse 8, WB 1: „Computerspiele“	Anwenden der Erkenntnisse zu Objekten auf Simulationsspiele
	Klasse 10, LB 2: “Arbeit in Projekten“	Gestalten eines eigenen Projektes
Gymnasium	**Klasse 9/10, LB 4:** „Algorithmen und Programme“	Kennen der Grundlagen der Programmierung, Anwenden der Phasen des Problemlöseprozesses
	Klasse 7, LB 2: „Computer benutzen – Elemente und Strategien“	Beherrschen typischer Handlungen bei der Nutzung von Anwendungen (Objekte auswählen, Attribute festlegen, Methoden nutzen)

Berufliches Gymnasium	**Informatik**	„Scratch 3“ ist leider nicht geeignet
	Informatiksysteme **Klasse 11, LB 2:** „Prozedurale Programmentwicklung“	Kennen von Begriffen und Modellen - Algorithmus, Daten- und Programmstruktur
	Klasse 11, LB 3: „Einführung in die objektorientierte Programmentwicklung“	Kennen der Prinzipien der Objektorientierung - Objekt, Klasse, Attribute, Methoden

3. Lernziele

Kognitive Lernziele:

Die SuS
- lernen den Begriffe „Algorithmus“ sowie seine Eigenschaften kennen.
- lernen die Begriffe „Objekte“, „Klasse“, „Attribute“ und „Methoden“ kennen.
- lernen Programmstrukturen anhand der Begriffe „Folge“, „Wiederholung“ und „Verzweigung“ kennen.
- setzen das Modell eines Algorithmus in einem einfachen Beispiel um.
- lernen die Grundlagen der Programmierung kennen.
- lernen den Problemlöseprozess durch Problemanalyse, Lösungsentwurf, Umsetzung und Test kennen.
- übertragen die Kenntnisse über den Problemlöseprozess auf das selbständige Lösen einfacher Probleme und wenden die Phasen des Problemlöseprozesses an.
- wenden ihre Kenntnisse zu Objekten auf Simulationsspiele an.
- teilen komplexe Probleme sinnvoll in einfachere Teilprobleme auf.
- synthetisieren Lösungen einfacher Probleme zu einer Gesamtlösung.
- gestalten ein eigenes Projekt.
- beurteilen ein Programm und den Programmcode hinsichtlich Funktionalität und Benutzerfreundlichkeit

Psychomotorische Lernziele:

Die SuS
- wenden typische Handlungen bei der Nutzung von Anwendungen (Objekte auswählen, Attribute festlegen, Methoden nutzen) an
- imitieren, präzisieren und naturalisieren den Umgang mit grafischen Oberflächen und Eingabeprinzipien wie dem Drag-and-Drop-Verfahren.

Affektive Lernziele:

Die SuS
- werden auf die besonderen Eigenheiten von Partner- und Gruppenarbeit aufmerksam.
- werden auf die Möglichkeiten kooperativer und kollaborativer Arbeitsweisen aufmerksam.
- werden auf die Vorteile der Modularisierung komplexer Probleme aufmerksam.
- werten ein Programm und den Programmcode hinsichtlich Funktionalität und Benutzerfreundlichkeit.
- organisieren ein eigenes Projekt.

4. Kompetenzentwicklung

Fachkompetenz:

Die SuS
- können den Begriff „Algorithmus“ sowie seine Eigenschaften erläutern.
- können die Begriffe „Objekte“, „Klasse“, „Attribute“ und „Methoden“ erläutern.
- können zu einem Objekt Attribute, Attributwerte und Methoden zuordnen.
- können verschiedene Datentypen verwenden.
- können Variablen und Wertzuweisungen verwenden.
- können das Modell eines Algorithmus in einem einfachen Beispiel anwenden und analysieren.
- können ihr Wissen zu Grundlagen der Programmierung in einem Beispiel anwenden.
- können Kenntnisse über den Problemlöseprozess auf das selbständige Lösen einfacher Probleme anwenden.
- können ein Programm und den Programmcode hinsichtlich Funktionalität und Benutzerfreundlichkeit analysieren und bewerten.
- können die algorithmischen Grundstrukturen Sequenz, Selektion und Zyklus in einem Scratch-Programmcode identifizieren und eigenständig implementieren.
- können einen gegebenen oder selbst erstellten Algorithmus in Scratch implementieren.

Lern-/Methodenkompetenz:

Die SuS
- können typische Handlungen bei der Nutzung von Anwendungen (Objekte auswählen, Attribute festlegen, Methoden nutzen) anwenden.
- können den Umgang mit grafischen Oberflächen und Eingabeprinzipien wie dem Drag-and-Drop-Verfahren anwenden.
- können komplexe Probleme sinnvoll in einfachere Teilprobleme aufteilen.
- können Vorteile der kooperativen und kollaborativen Gruppenarbeit wiedergeben.
- können Aufgaben innerhalb eines Problemlösungsprozesses auf verschiedene Gruppenmitglieder verteilen
- können ein eigenes Projekt erstellen und gestalten.

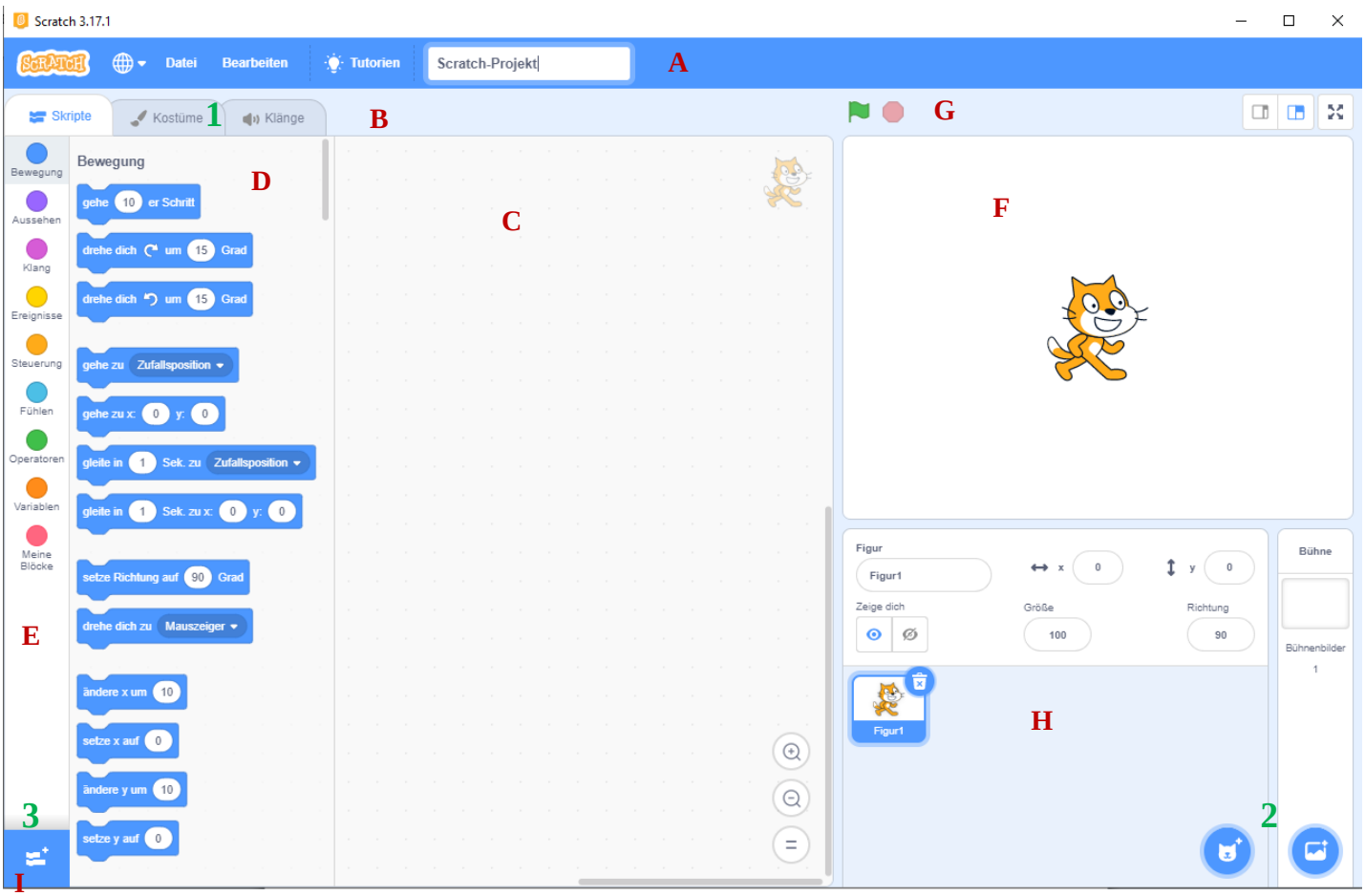
Sozialkompetenz:

Die SuS
- können gemeinsam mit anderen SuS ein Problem in Teilprobleme aufteilen und einzelne Arbeitsschritte innerhalb einer Partner- oder Gruppenarbeit verteilen.
- können mit anderen SuS problembezogen kommunizieren.
- können eigene Beurteilungen und Bewertungen wertschätzend formulieren und äußern.
- können im Rahmen einer Projektarbeit gemeinsam verantwortungsvoll miteinander arbeiten.

Selbstkompetenz:

Die SuS
- können ein Programm und den Programmcode hinsichtlich Funktionalität und Benutzerfreundlichkeit analysieren und bewerten.
- können die Vor- und Nachteile verschiedener Ansätze bei der Umsetzung einer Problemlösung erkennen und ihren eigenen Lösungsansatz einordnen.
- können ihr eigenes Ergebnis oder ihre Teilhabe an einer Gruppen- oder Partnerarbeit selbstkritisch beurteilen und bewerten.

5. *Prinzipieller Aufbau*



Scratch 3 ist grundsätzlich sehr übersichtlich und intuitiv aufgebaut. Das Programm ist zudem grafisch und ästhetisch sehr ansprechend gestaltet und so für Kinder und Jugendliche gut geeignet.

A- Ganz oben befindet sich die **äußere Menüleiste**. Hier lässt sich die Sprache einstellen, lassen sich Dateien laden und speichern, ein Turbo-Modus einschalten und ausführliche Tutorien einsehen und abspielen. Außerdem wird der Name des aktuellen Projekts angezeigt.

B- Die **innere Menüleiste** erlaubt die Auswahl zur Bearbeitung der Skripte, der Kostüme (dem Aussehen der Figuren), der Bühnenbilder (Hintergründe) und der Klänge.

C- Der **Arbeitsbereich** erlaubt das Anordnen der Skriptbausteine per Drag and Drop nach dem Puzzle-Prinzip. An dieser Stelle wird der Quellcode erstellt, indem die einzelnen „Puzzleteile“ (Befehle, Bedingungen usw.) aneinander und ineinander gefügt werden. Ist in der inneren Menüleiste nicht „Skripte“ ausgewählt, können hier das Aussehen der Figuren, die Hintergründe oder einzelne Klänge bearbeitet werden.

D- Im **Befehls-Blöcke-Bereich** befinden sich die einzelnen Bausteine, die für das Skript verwendet werden können.

E- Diese sind im **Blockauswahl-Bereich** farblich sortiert und mit einfachen Begriffen wie „Bewegung“, „Steuerung“ oder „Fühlen“ beschriftet. Sie bieten verschiedene Befehle, Bedingungen, Operatoren,

Variablen und algorithmische Strukturen, welche im Programmcode verwendet werden können. Es können auch eigene Blöcke erstellt werden.

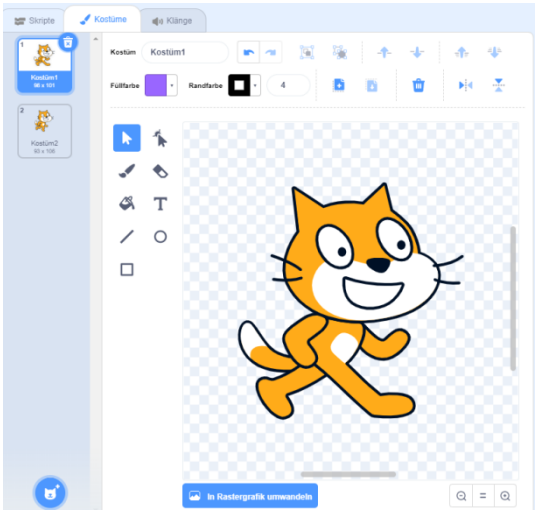
F- In der **Szenenbildanzeige** lässt sich das entstandene Programm ausprobieren.

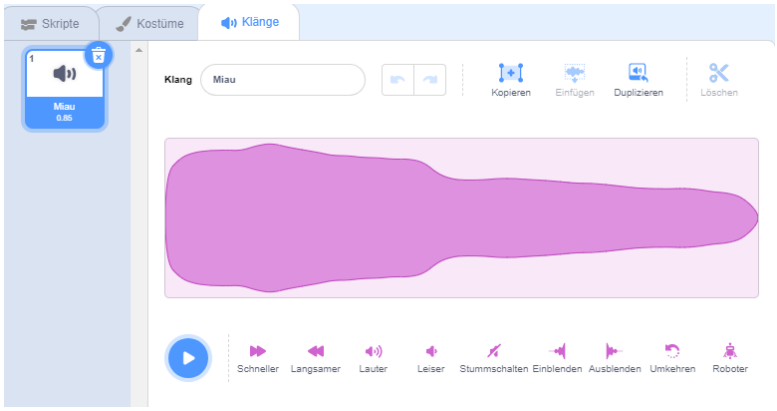
G- Dafür muss die Ausführung mit einer **grünen Fahne** (Startsymbol) gestartet oder mit **einem Stopp-Symbol** gestoppt werden.

H- Die einzelnen Objekte bzw. Figuren (standardmäßig ist eine Katze eingestellt) und Hintergründe (Bühnenbilder) lassen sich im **Objekt-Auswahl-Bereich** rechts unten auswählen. Dafür bietet *Scratch 3* eine sehr reiche Auswahl an Figuren und Hintergründen bzw. Bühnenbildern an.

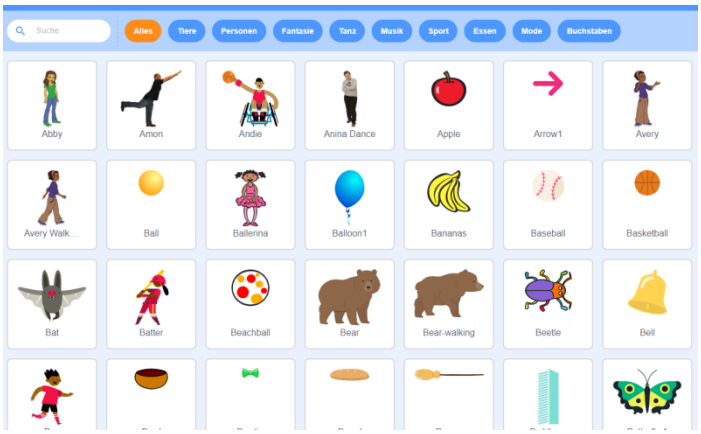
I- Im Bereich **Erweiterungen** lassen sich weitere Elemente integrieren, beispielsweise eine Videoerfassung.

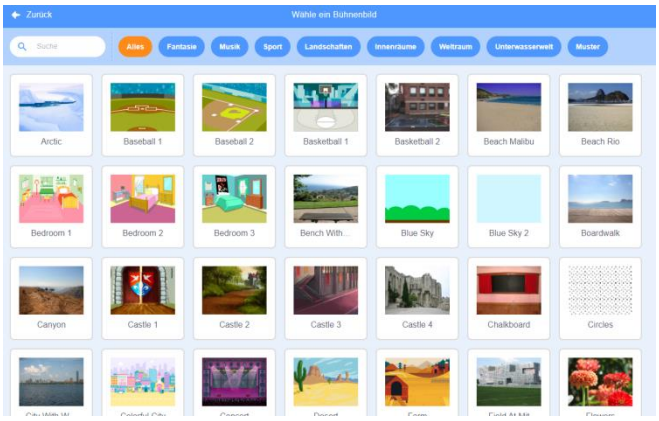
1- Die Bearbeitung von Kostümen und Klängen ist ebenso klar gestaltet und intuitiv leicht erfassbar und bietet umfangreiche Funktionsmöglichkeiten.



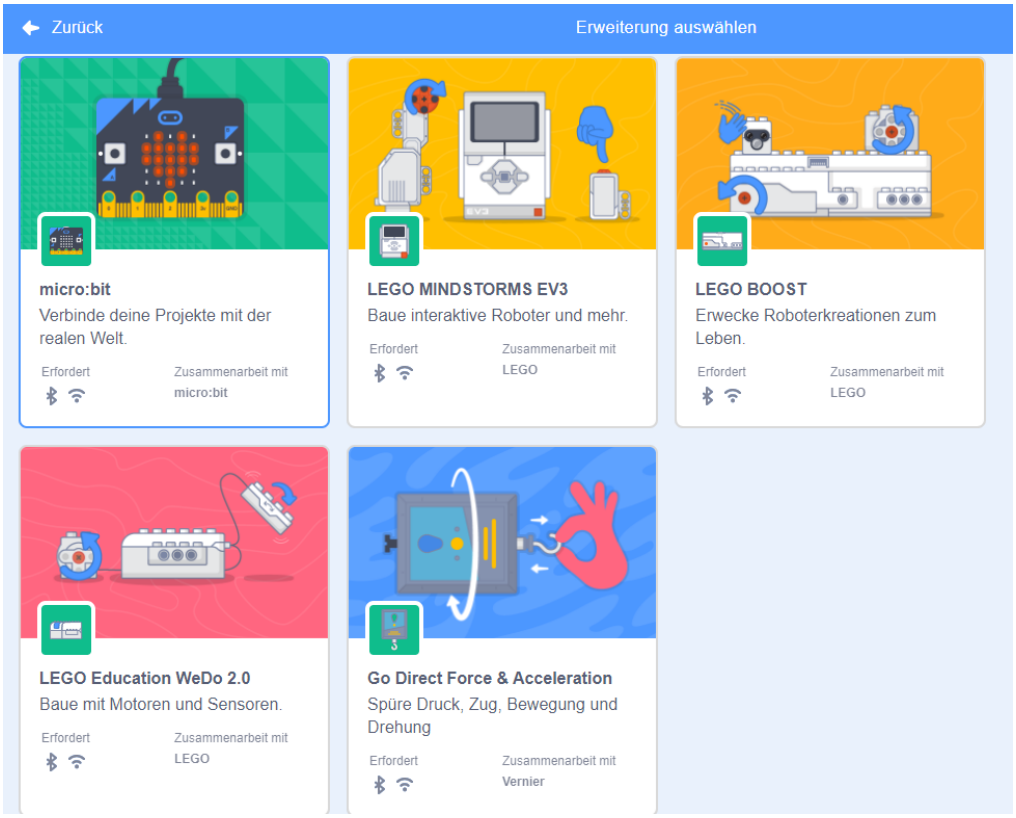


2- Die Auswahl an Figuren und Bühnenbildern ist gut sortiert und umfangreich.





3- Die Erweiterungen ermöglichen auch Unterrichtsstunden im Bereich „Robotik“. So können mit LEGO MINDSTORMS EV 3 oder LEGO BOOST sowie MakeyMakey verschiedene Hardwarekomponenten intergiert und ansteuert werden.



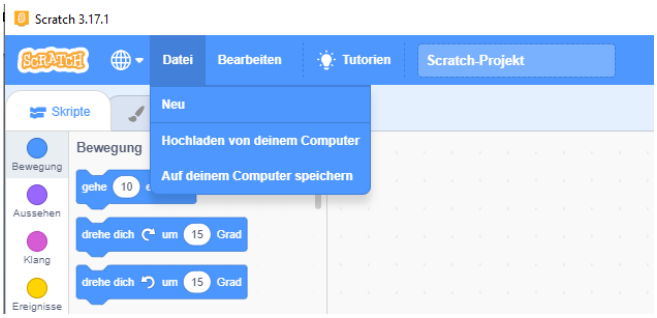
6. *Handhabung*

Die Handhabung ist grundsätzlich sehr einfach und intuitiv. Sie ist grafisch mit dem Drag und Drop – Verfahren realisiert. Es können für die verschiedenen Figuren (Objekte) einzelne Skripte (Methoden) erstellt werden. S ist also nicht notwendig, das gesamte Programm in einem Skript anzuordnen.

Bei den Schülerlösungen sollte darauf geachtet werden, dass die Methoden immer abgeschlossen sind. Das heißt, der Block „Stoppe“ im Skript angeordnet wurde. Zudem sollte mindestens einmal im Skript der Startblock „Wenn [grüne Fahne] angeklickt“ angeordnet sein.

Die Figuren, Hintergründe, Kostüme und Klänge der Figuren können aus einer großen Auswahl gewählt werden oder auch selbst erstellt oder verändert werden.

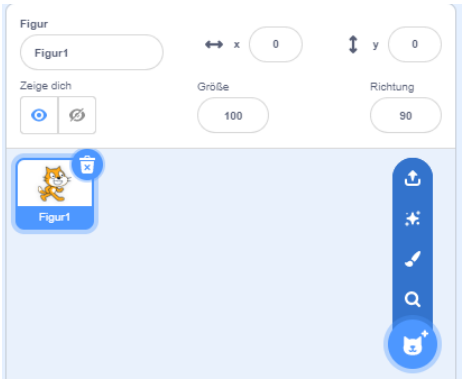
Insgesamt bietet das Programm so sehr große kreative Möglichkeiten.



1. **Erstellen eines neuen Projekts, Laden und Speichern**

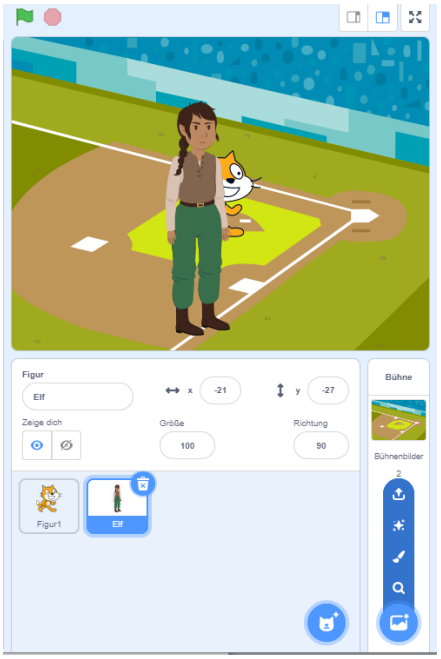
   Nutzung von **Datei -> Neu** bzw. **Datei -> Hochladen** oder **Datei -> Speichern.**

Der Begriff „Hochladen“ orientiert sich dabei an der Web-Anwendung von *Scratch 3*.



2. Hinzufügen eines Objekts (Figur)

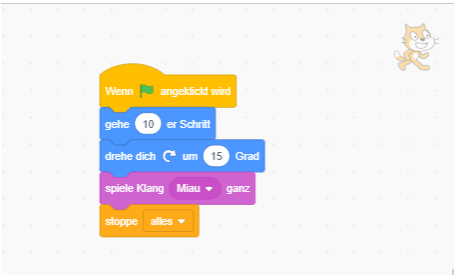
Mit einem Klick auf den Button „Figur wählen“ kann eine Figur vom Computer geladen, selbst erstellt oder aus der Galerie gewählt werden. Es kann auch eine Zufallsfigur generiert werden.



3. Hinzufügen eines Hintergrunds (Bühnenbild)

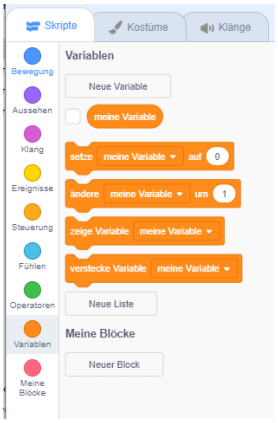
Mit dem Button „Bühnenbild wählen“ kann ebenso ein Hintergrund vom Computer geladen werden, selbst erstellt oder aus einer Galerie ausgewählt werden. Auch hier kann ein zufälliger Hintergrund generiert werden.

Hintergrund und Kostüm können im Blockauswahl-Bereich unter „Aussehen“ geändert werden, sodass ein Wechsel von Kleidung oder Bühnenbild während des Programmablaufs möglich wird.



4. Programmcode erstellen

Programmcode kann für jede Figur und jedes Bühnenbild einzeln erstellt werden. Dafür müssen die entsprechenden Befehlsblöcke aus dem Befehls-Blöcke-Bereich in den Arbeitsbereich gezogen und entsprechend zusammengesetzt werden. Durch das zugrundeliegende Puzzle-Prinzip ist es hier nicht möglich, einen ungültigen oder fehlerhaften Programmcode zu erzeugen.



Es sollte zumindest einmal der Baustein „Wenn [grüne Flagge] angeklickt wird“ sowie der Baustein „stoppe alles“ verwendet werden.

Neben einfachen Anweisungsblöcken gibt es die Möglichkeit Tastatur-, Maus- oder Bildschirmabfragen vorzunehmen, Schleifen und Bedingungen zu gestalten sowie Variablen und Operatoren einzusetzen.

Im Feld „Meine Blöcke“ können einzelne Funktionen erstellt werden, die dann für das jeweilige Objekt zur Verfügung stehen.

Mit Hilfe der „Erweiterungen“ können auch externe Geräte eingebunden werden. Außerdem besteht die Möglichkeit, Instrumente wiederzugeben, die Figuren zeichnen zu lassen und vieles mehr.

7. Weitergehende Hinweise

Die neuste Version Scratch 3.19.2 wird ergänzt durch eine Java-Script-basierte Browserversion, welche über die gleichen Funktionen wie die Desktop-Version verfügt und identisch aufgebaut ist. *Scratch 3* muss also nicht auf den Schulrechnern installiert werden.

Sie ist erreichbar unter https://scratch.mit.edu/projects/editor/?tutorial=getStarted (abgerufen am 23.02.21).
Eine Anmeldung bzw. Registrierung ist möglich, aber nicht notwendig dafür.

Der Scratch-Offline-Editor (Desktop-Version) kann für verschiedene Plattformen unter dieser Adresse heruntergeladen werden: https://scratch.mit.edu/download (abgerufen am 23.02.21)

Auf der Seite https://scratch.mit.edu/explore/projects/all (abgerufen am 23.02.21) findet sich eine umfangreiche Sammlung von frei verfügbaren Scratch-Projekten.

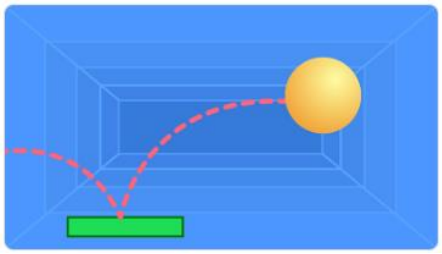
Im Bereich „Scratch für Lehrkräfte“ gibt es zudem u.a. die Möglichkeit, mit Hilfe eines Lehrerbenutzerkontos Schülerkonten anzulegen sowie Projekte und Kommentare zu verwalten:
https://scratch.mit.edu/educators#teacher-accounts (abgerufen am 23.02.21).
Hier finden sich auch umfangreiche Materialien und Hilfen für Lehrkräfte sowie die Möglichkeit, sich mit anderen Lehrkräften auszutauschen.

Weitere Informationen finden sich auf der Website https://scratch.mit.edu/ (abgerufen am 23.02.21).

8. Aufgaben

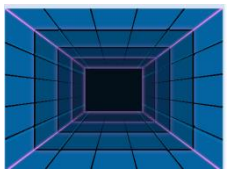
Scratch 3 - Aufgaben

1. Mache dich zuerst mit dem Programm vertraut. *(5 min)*
   a) Entferne die Figur „Katze“ und probiere stattdessen andere Figuren aus.
   b) Probiere verschiedene Hintergründe (Bühnenbilder) aus.
   c) Schau dir die verschiedenen Möglichkeiten im Bereich „Skripte“ an und überprüfe, wie sie jeweils zusammenpassen können. (Puzzle-Prinzip)
   d) Probiere die Funktionen „Auf deinem Computer speichern“ und „Hochladen von deinem ter“ aus. Wähle einen geeigneten Namen.
   e) Starte nun ein neues Projekt.



2. Max will für seinen Freund Jonas ein Geschicklichkeitsspiel programmieren. Es soll „**Ping Pong**“ heißen. Dabei soll ein **Ball** mit einem **Paddel** (ein flacher Schläger) getroffen werden.
   Der Ball soll ansonsten in der Welt frei „herumhüpfen“ und **am Rand abprallen**.

   a) Wähle zunächst den Ball und ein Bühnenbild aus.
   Zum Beispiel dieses:
   (5min)



   b) Schreibe nun zuerst ein kleines Programm für den Ball: *(5min)*

Der Ball startet, wenn die grüne Flagge gedrückt wird.

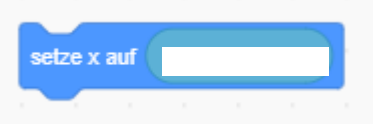
Der Ball bekommt eine Startrichtung.

Der Ball bewegt sich fortlaufend mit Hilfe einer Schleife.

Geschwindigkeit und Bewegung werden durch „Gehe … Schritt“ bestimmt.

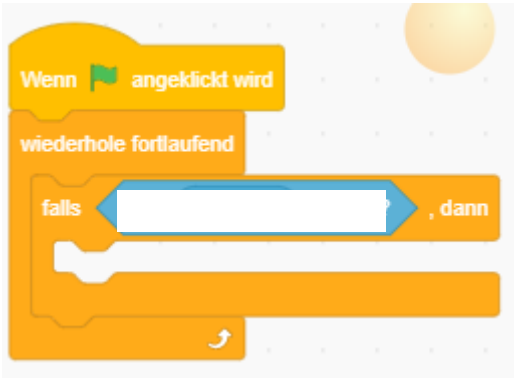
Der Ball soll am Rand abprallen. Wähle aus.

   c) Nun benötigt Max noch ein Paddel. *(5min)*
   Wähle die Figur „Paddle“ aus, setze sie an die richtige Stelle im Bühnenbild und schreibe für das Paddle eine Methode, mit der man es bewegen kann.
   Nutze dafür wieder die Befehle „Wenn grüne Flagge gedrückt wird“, eine Schleife und eine Kombination aus „Bewegung“ und „Fühlen“, bei der immer die aktuelle x-Position der Maus gewählt werden soll.
   Probiere einmal aus, ob du das Paddle mit der Maus steuern kannst.



d) Der Ball soll nun vom Paddel abprallen. Gehe dafür zurück zum Ball und „schreibe“ eine Funktion, die das erreicht. *(7 min)*

Nutze dafür diesen unvollständigen Programmteil:

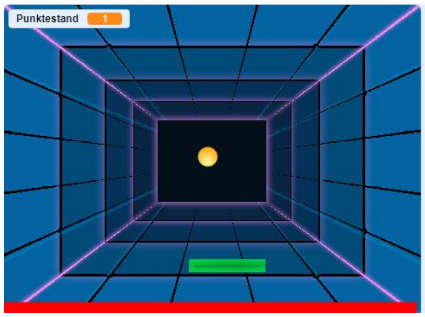


Tipp: Bedenke, dass der Ball die Richtung ändern muss und sich danach weiterbewegen. Überlege, in welche Richtung der Ball am besten abprallen sollte.

e) Max möchte nun, dass seine Punkte gezählt werden.
Erstelle dafür eine **Variable** mit dem Namen „Punktestand“.
(5 min)

Diesen musst du zunächst auf „0“ stellen und dann bei jeder Berührung des Paddels um eins erhöhen. Nutze dafür geeignete Befehle und füge sie an der richtigen Stelle in dein Programm ein.
Bei jedem Erhöhen des Punktestands soll der Ball außerdem ganz kurz – 0,5 Sekunden – verzögert werden.

f) Max möchte nun noch, dass das Spiel beendet ist, wenn eine rote Linie berührt wird, die sich unter dem Paddle befindet.
(7 min)



Ergänze diese Linie und füge eine Methode hinzu, die nach Klicken der grünen Fahne fortlaufend prüft, ob der Ball die Linie berührt. Dies kannst du entweder bei dem Objekt „Ball“ oder bei dem Objekt „Linie“ tun. Sobald die Linie berührt wird, soll alles gestoppt werden.

g) Speichere dein Programm ab und probiere es einige Mal aus. Was gefällt dir daran, was könnte man noch verbessern? Schreibe mindestens drei Verbesserungsmöglichkeiten auf. *(8 min)*

Weitere mögliche Aufgaben:

3. Das Spiel von Max ist nun eigentlich fertig. Allerdings möchte es Max noch verbessern. Ergänze das Programm um folgende Funktionen:

   a) Beim Treffen des Paddels soll ein geeignetes Geräusch, zum Beispiel „Pop“, zu hören sein. Ebenso soll beim Treffen der Linie ein Geräusch zu hören sein. Dieses soll jedoch ein bisschen anders klingen als der vorgefertigte Klang „Boing“.
   Tipp: Du kannst Klänge auch bearbeiten, du kannst sie zum Beispiel verdoppeln.

   b) Vielleicht ist dir aufgefallen, dass der Ball, wenn er die rote Linie berührt hat, dort manchmal „steckenbleibt“. Ändere dein Programm so, dass der Ball zu Beginn jedes Spiels wieder in der Mitte startet.

   c) Max findet sein Spiel zu einfach. Er möchte, dass es eine neue Variable gibt, die „Level“ heißt. In Abhängigkeit von diesem Level soll sich die Geschwindigkeit des Balles ändern. Das nächste Level soll dabei immer nach 10 Punkten erreicht werden.

   Tipp: Du kannst dafür zum Beispiel die folgende Kombination aus Operatoren nutzen. Überprüfe auch, welche Variablen der Spieler sehen können soll und welche nicht.

   d) Jetzt möchte Max noch, dass sich bei jedem neuen Level das Bühnenbild nach einer festen Abfolge ändert. Dafür geht er in den Bereich „Bühnenbild“, fügt eine Auswahl an möglichen Bühnenbildern hinzu und ergänzt sein Programm entsprechend.

   e) Max hat sich überlegt, dass es auch schön wäre, das Spiel zusammen mit Jonas spielen zu können. Dafür möchte er ein zweites, rotes Paddel hinzufügen, dass sich mit den Pfeiltasten steuern lässt. Erstelle dieses Paddel.

   f) Du hast dir in Aufgabe 2g auch selbst Gedanken gemacht, wie du das Programm verbessern kannst. Versuche einmal, ob du diese Verbesserungen vornehmen kannst.

4. Max ist zufrieden mit seinem Spiel und möchte nun noch andere Möglichkeiten von ***Scratch*** ausprobieren. Beispielsweise ein Autorennen:

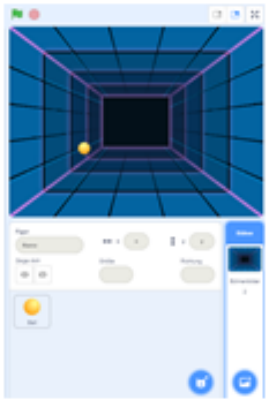
   Es soll zwei Autos beinhalten. Wenn die Autos die rote Linie berühren, sollen sie langsamer werden, wenn sie auf dem Rasen sind, sollen sie fast stehen bleiben. Ein Zähler soll angeben, wie viele Runden das rote und das blaue Auto schon gefahren sind. Außerdem soll es vor dem Beginn einen Countdown geben.

5. Scratch bietet viele weitere Möglichkeiten. Du kannst Adventure-Spiele, Sprachausgaben oder Animationen erstellen. Schaue dir die Tutorials an, die dich interessieren und programmiere deine eigene Welt – ganz nach deinem Geschmack!

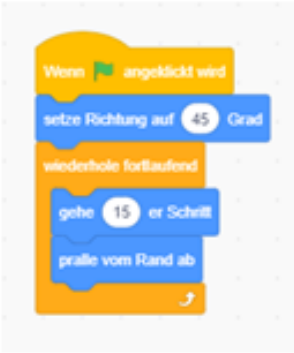
9. Lösungen zu den Aufgaben

1. *a-e – individuelle Lösungen*

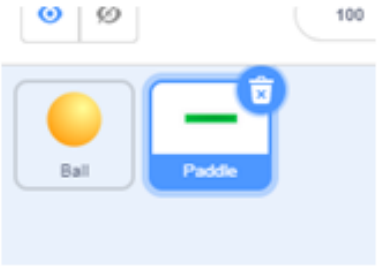
2. a)



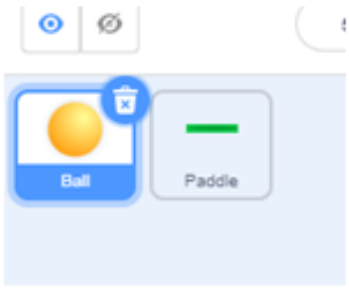
b)



c)



d)



e)


Wenn angeklickt wird
setze Punktestand auf 0
setze Richtung auf 45 Grad
wiederhole fortlaufend
gehe 15 er Schritt
pralle vom Rand ab
Wenn angeklickt wird
wiederhole fortlaufend
falls wird Paddle berührt? , dann
ändere Punktestand um 1
drehe dich um 90 Grad
gehe 15 er Schritt
warte 0.5 Sekunden

f)


Wenn angeklickt wird
setze Punktestand auf 0
setze Richtung auf 45 Grad
wiederhole fortlaufend
gehe 15 er Schritt
pralle vom Rand ab
falls wird Paddle berührt? , dann
ändere Punktestand um 1
drehe dich um 90 Grad
gehe 15 er Schritt
falls wird Line berührt? , dann
stoppe alles

3. a)


Wenn angeklickt wird
setze Punktestand auf 0
setze Richtung auf 45 Grad
wiederhole fortlaufend
gehe 15 er Schritt
pralle vom Rand ab
falls wird Paddle berührt? , dann
spiele Klang Pop ganz
ändere Punktestand um 1
drehe dich um 90 Grad
gehe 15 er Schritt
falls wird Line berührt? , dann
spiele Klang Boing ganz
warte 2 Sekunden
stoppe alles

b)


Wenn angeklickt wird
setze Punktestand auf 0
setze x auf 0
setze y auf 0
setze Richtung auf 45 Grad
wiederhole fortlaufend
gehe 15 er Schritt
pralle vom Rand ab
falls wird Paddle berührt? , dann
ändere Punktestand um 1
spiele Klang Pop
drehe dich um 90 Grad
gehe 15 er Schritt
falls wird Line berührt? , dann
spiele Klang Boing ganz
stoppe alles

c)


Wenn angeklickt wird
setze x auf 0
setze y auf 0
setze Punktestand auf 0
setze Level auf 1
setze Geschwindigkeit auf 15
verstecke Variable Geschwindigkeit
setze Richtung auf 45 Grad
wiederhole fortlaufend
gehe Geschwindigkeit er Schritt
pralle vom Rand ab
falls wird Paddle berührt? , dann
spiele Klang Pop ganz
ändere Punktestand um 1
drehe dich um 90 Grad
gehe Geschwindigkeit er Schritt
falls wird Line berührt? , dann
spiele Klang Boing ganz
warte 2 Sekunden
stoppe alles
falls Punktestand > Level * 10 , dann
ändere Geschwindigkeit um 5
ändere Level um 1

d)


Wenn angeklickt wird
setze x auf 0
setze y auf 0
setze Punktestand auf 0
setze Level auf 1
setze Geschwindigkeit auf 15
verstecke Variable Geschwindigkeit
setze Richtung auf 45 Grad
wiederhole fortlaufend
gehe Geschwindigkeit er Schritt
pralle vom Rand ab
falls wird Paddle berührt? , dann
spiele Klang Pop ganz
ändere Punktestand um 1
drehe dich um 90 Grad
gehe Geschwindigkeit er Schritt
falls wird Line berührt? , dann
spiele Klang Boing ganz
warte 2 Sekunden
stoppe alles
falls Punktestand > Level * 10 , dann
ändere Geschwindigkeit um 5
ändere Level um 1
wechsle zu Bühnenbild nächstes Bühnenbild

e)


Zeige dich
Größe
100
Ball
Paddle
Line
Paddle2


Wenn angeklickt wird
wiederhole fortlaufend
falls Taste Pfeil nach rechts gedrückt? , dann
gehe 10 er Schritt
falls Taste Pfeil nach links gedrückt? , dann
gehe -10 er Schritt


Zeige dich
Größe
50
Ball
Paddle
Line
Paddle2


setze Richtung auf 45 Grad
wiederhole fortlaufend
gehe Geschwindigkeit er Schritt
pralle vom Rand ab
falls wird Paddle berührt? , dann
spiele Klang Pop ganz
ändere Punktestand um 1
drehe dich um 90 Grad
gehe Geschwindigkeit er Schritt
falls wird Paddle2 berührt? , dann
spiele Klang Pop
ändere Punktestand um 1
drehe dich um 90 Grad
gehe Geschwindigkeit er Schritt
falls wird Line berührt? , dann

10. Screencast

(hohe Auflösung, 28,6 MB)
https://wolke.schullogin.de/index.php/s/7wymcyAcm8A98xe

(geringere Auflösung, 9,4 MB)
https://wolke.schullogin.de/index.php/s/bEdF3GdmoQEw2Zy

11. Quellennachweis

Textnachweise:

- https://de.wikipedia.org/wiki/Scratch_(Programmiersprache) (zuletzt abgerufen am 02.03.2021) (1. Kurzvorstellung)
- https://scratch.mit.edu/ (zuletzt abgerufen am 02.03.2021) (7. Weitergehende Hinweise)
- Sächsisches Staatsministerium für Kultus , Lehrpläne & weitere Materialien www.bildung.sachsen.de/apps/lehrplandb/ (zuletzt abgerufen am 02.03.2021)

Bildnachweise:

- https://scratch.mit.edu/ (zuletzt abgerufen am 02.03.2021) (5. Prinzipieller Aufbau)
- Screenshots des Programms Scratch 3 in der Version 3.17.1
- (5. Prinzipieller Aufbau, 6. Handhabung, 8. Aufgaben, 9. Lösungen zu den Aufgaben)