Game Creator Workshop mit Scratch für Einsteiger und Fortgeschrittene

**Kurzbeschreibung und pädagogischer Ansatz**

In diesem Workshop erfahren die Teilnehmer\*innen erste Ansätze der Spiele-programmierung. Dabei können in diesem Workshop auch Bereiche aufgegriffen werden, die nicht direkt mit digitalen Medien in Verbindung gebracht werden. Dadurch können Teilnehmer\*innen – wie auch Mitarbeiter\*innen der Einrichtungen vor Ort – angesprochen werden, die sonst keinen Zugang zu dem Thema haben. Was wird vermittelt?

* Planung eines Projekts
* Einführung in das Programmieren auf Englisch „Coding“
* Spiel-Design / Controller-Design
* Zeichentechniken / Handwerkliche Fähigkeiten

**Gruppengröße**

**20 Teilnehmer\*innen** bei einem Betreuungsschlüssel von 1:5.

Die Werkstatt muss in jedem Fall von einer Person betreut werden, da hier gesägt und gebohrt wird. Diese Person muss gegebenenfalls noch zusätzlich vor Ort sein.

**Setting**

Ein **geschlossener Workshop** als Angebot der außerschulischen Jugendarbeit in den Räumen des Kooperationspartners.

**Dauer**

Eine **5-tägige Veranstaltungswoche** mit **täglich drei Stunden** Workshop. Auch ein Wochenend-Workshop ist mit diesem Setting möglich – z.B. weglassen der Design- und Zeicheneinheiten und Verlängern der Workshop Zeit auf täglich **vier Stunden**.

**Zielgruppe**

Kinder und Jugendliche zwischen **10** und **14** Jahren

**Ausstattung ,Technik, Material und Raumbedarf**

* ***Veranstaltungsraum:***WLAN/LAN, je 1 Laptop/PC für jede Programmiergruppe und zwei MakeyMakey Controller (mit Draht, Alufolie, Kabel).
* ***Werkstatt:*** Werkzeug (Sägen, Hämmer, Schraubendreher, Feilen), Holz insgesamt

etwa 1m² für Controller, entsprechend mehr für größere Projekte, Holzschrauben, Leim, Schmirgelpapier .

* ***Projektplanung und Design/Zeichnung****:* Papier DIN A3 bis A0 oder Tafel, DIN A4 Papier und Zeichenstifte für etwa 5 Teilnehmer\*Innen, Acrylfarben oder ähnliches zum Gestalten der Werkstücke, Scanner oder Fotoapparat mit Stativ zum Digitalisieren der Zeichnungen.
* Einverständniserklärungen der Eltern, wenn Fotos der Teilnehmer\*innen im Spiel verwendet werden sollen.

**Aufwand**

* Anlegen eines Workshop-Kontos unter <https://scratch.mit.edu/>
* Vorbereiten einiger Beispielcodeschnipsel zum Beispiel für Tastatursteuerung
* empfohlener Betreuungsschlüssel 1:3 (Teamer\*Innen: Teilnehmer\*Innen)
* Karten, Blätter oder ähnliches mit Ideen für eine Geschichte – z.B. „Drache“, „Wettrennen“, „Schatz“, „Meer“…
* Scratch-Karten nach Spiel-Konzept
* Scratch-Tutorial (bewegter Hintergrund, realistisches Springen)

**Ablaufplan**

**Begrüßung** des Workshop-Referenten und Kennenlern-Runde für die Teilnehmer\*Innen

**Einführung in das Thema**:

Was braucht man alles, um ein Spiel zu programmieren?

Dies sind die Punkte, auf die es hinausläuft:

* Idee/Geschichte/Ende
* Charaktere, Gegner und Gegenstände, Hintergründe/Umgebung
* Aufgaben/Rätsel/Bosskämpfe
* Controller/Steuerung
* Programmcode

Jetzt wird geklärt, was für eine Art von Spiel in der Woche entwickelt wird:

z.B. ein Adventure - Was ist das? Was gibt es für Beispiele?

*(Grundsätzlich sind auch andere Genre möglich. Vor allem Jump‘n‘Run-Spiele bieten sich an, da die meisten Kinder unabhängig vom Geschlecht Mario & Co. mögen.*

*Vorteil bei einem einfachen Adventure-Setting: Es ist offener und einfacher zu programmieren, da vieles über Textfelder erzählt wird und kaum Spielphysik verwendet wird -> interaktiver Film.)*

**Spielgeschichte**

Gemeinsam wird anschließend eine Geschichte entwickelt. Dafür werden alle Ideen auf einem Plakat ähnlich einer Mindmap gesammelt. Wichtige Punkte sind:

* Womit fängt die Geschichte an?
* Womit hört sie auf?
* Was kann es für spannende Ereignisse im Laufe der Erzählung geben?
* Welche Charaktere gibt es?
* In welcher Umgebung spielt die Geschichte?

Dieser Teil entwickelt sich erst im Laufe des Workshops, daher ist er nicht komplett von Beginn an planbar. Es wichtig, zu jeder Zeit alle Jung-Programmierer\*innen aufzuklären, wie der Stand der Dinge ist. Dafür werden die Ideen auf einem Plakat mit dem Vermerk festgehalten, in welcher Kleingruppe der entsprechende Punkt behandelt wird. An dieser Stelle muss darauf geachtet werden, dass die Ideen realisierbar sind.

Zudem treffen sich alle Kleingruppen zu Beginn des Tages, nach der Pause und zum Tagesabschluss gemeinsam in der großen Gruppe. Hier wird geklärt, an welcher Stelle noch nachgearbeitet werden muss. Auch Ideen für bestimmte Spielsituationen können hier noch weiter gesammelt werden. Danach wird die Gruppe in folgende etwa gleichgroße Kleingruppen aufgeteilt:

**Gruppe 1: Coding**

Mit Scratch wird das eigentliche Spiel erstellt. Gestartet wird dabei mit dem gemeinsam erarbeiteten Beginn der Geschichte. In der Kleingruppe wird besprochen, wie die einzelnen Szenen und Level gestaltet werden. Eine Person setzt dies dann in Programmcode um. Die anderen helfen, bzw. korrigieren den Code. Nach etwa fünf Minuten kann gewechselt werden und eine andere Person programmiert weiter. Dazu wird das Prinzip der Paarprogrammierung \*, welches hier ein wenig erweitert wird, erklärt.

Ein wichtiges Element sind Rätsel oder besondere Hindernisse, die das Spiel interessant machen. In der Kleingruppe wird überlegt, welche der Vorschläge übernommen werden können und wie dies umzusetzen ist. Auch hier führt ein\*e Teilnehmer\*in die Programmierung aus und die anderen unterstützen.

All diese Programmierungen werden mit Standardfiguren aus der Scratch-Bibliothek realisiert und später dann auf das endgültige Spiel übertragen. Um die Grundlagen zu erlernen bieten sich, für Programmier-Einsteiger, die Scratch-Aufgabenkarten und projekteigene Tutorien an.

**Gruppe 2: Coding-Spezial: Animation, Physik**

*(Bei wenigen Teilnehmer\*innen kann diese Gruppe der Gruppe 1 zugeordnet werden.)*

In dieser Kleingruppe werden gemeinsam bestimmte Sequenzen programmiert. Am wahrscheinlichsten sind dies Programmblöcke, um den Charakter springen zu lassen. Je nach Vorkenntnissen der Gruppe kann dies einfach oder komplex gelöst werden. Bei der komplexeren Variante lässt sich die Figur realistischer durch die Spielewelt steuern.

\* Paarprogrammierung

Bei der Erstellung des Quellcodes arbeiten jeweils zwei Programmierer gleichzeitig an einem Arbeitsplatz: Einer schreibt den Code, während der andere über die Problemstellungen nachdenkt, den geschriebenen Code kontrolliert sowie Probleme, die ihm dabei auffallen, sofort anspricht. Diese können dann sofort im Gespräch gelöst werden. Die beiden Programmierer sollten sich in den beiden Rollen abwechseln. Auch die Zusammensetzung der Paare sollte sich häufig ändern. *Quelle: Wikipedia Stand: Dezember 2020*

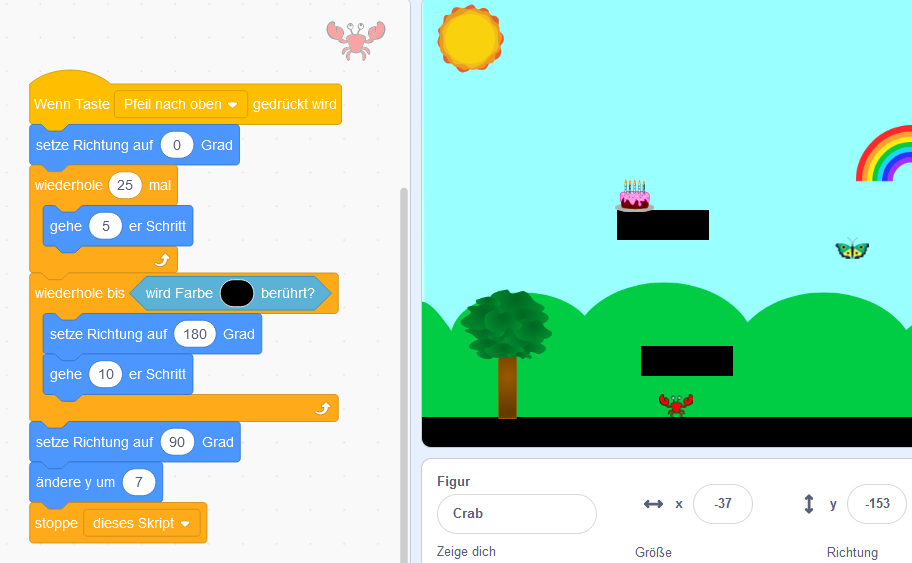


Abbildung 1: einfacher Beispielcode Bewegung Krabbe

Eine Person setzt dies dann in Programmcode um. Die anderen helfen bzw. korrigieren. Nach etwa fünf Minuten kann gewechselt werden und eine andere Person führt die Programmierung durch. Dazu wird das Prinzip der Paarprogrammierung, welches hier ein wenig erweitert wird, erklärt.

Auch die Animationen der verschiedenen Figuren werden hier exemplarisch erstellt. All diese Programmierungen werden mit Standardfiguren realisiert und später dann auf die eigentliche Figur im endgültigen Spiel übertragen.

Für Einsteiger in Scratch bieten sich für den Beginn die Scratch-Karten und die internen Tutorien an, um die Grundlagen zu erlernen.

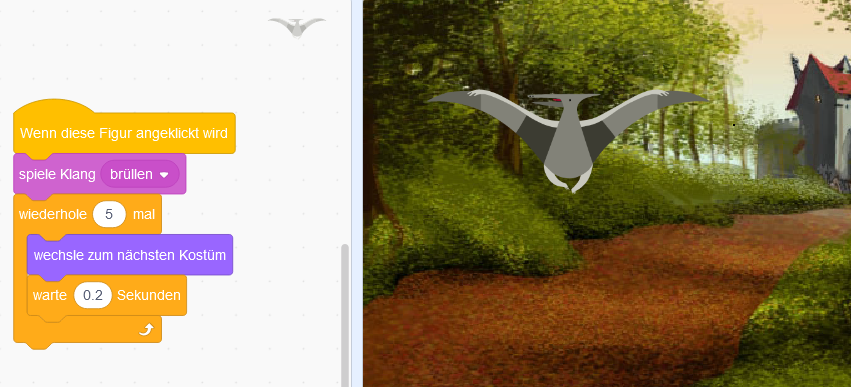


Abbildung 2: kurze Animation eines Dinosauriers

**Gruppe 3: Bauen**

Diese Kleingruppe überlegt sich, wie das Spiel gesteuert werden soll. Zu berücksichtigen sind dabei die gesamten Aktions- und Interaktionsmöglichkeiten der zu steuernden Figur. Für einen klassischen „Plattformer“ reichen drei Tasten für die Bewegungen nach links, rechts und oben. Bei einem Adventure wird eventuell gekämpft oder mit den anderen Figuren geredet. Mit einem MakeyMakey Controller sind viele Tastatureingaben zu realisieren. Es müssen zur Auslösung lediglich Stromkreise mit schwachen Strömen geschlossen werden. Dabei soll der MakeyMakey so verbaut werden, dass die Kabel und das Board selber nicht mehr zu sehen ist. Hierfür eignet sich Holz am besten. Aber auch andere Materialien oder Gegenstände, in die die Steuerung eingearbeitet werden kann, sind möglich. Mit letzterem könnte z.B. ein kleiner Schrank in eine Spielkonsole verwandelt werden.

Nach der Vorüberlegung muss genau geplant werden, wie der Controller aussehen soll. Für größere Vorhaben empfiehlt sich eine Zeichnung. Abschließend wird das Werkstück noch verziert. Dafür bieten sich Acrylfarben an. Auch Graffiti ist eine Möglichkeit.



Abbildung 3: Fertiger Controller für ein Rennspiel

**Gruppe 4: Hintergründe und Charaktererschaffung**

Hier werden die Figuren und die Umgebung des Spiels gestaltet. Wichtige Überlegungen sind: Welche Figuren gibt es? Und in welchen Umgebungen spielt die Geschichte? Was haben die Figuren für Merkmale bzw. Wesenszüge und wie kann dies gestalterisch ausgedrückt werden? *(Hier lassen sich gut soziale Werte vermitteln: z.B. es werden Gesellschaftsgruppen abgebildet oder warum sehen die Figuren so aus, wie wir sie zeichnen?)* Grundsätzlich können die Teilnehmer\*innen hier auf zwei verschiedene Arten gestalterisch tätig werden: Scratch hat ein eingebautes Malprogramm, in dem mit virtuellen Stiften und geometrischen Formen „gemalt“ werden kann.

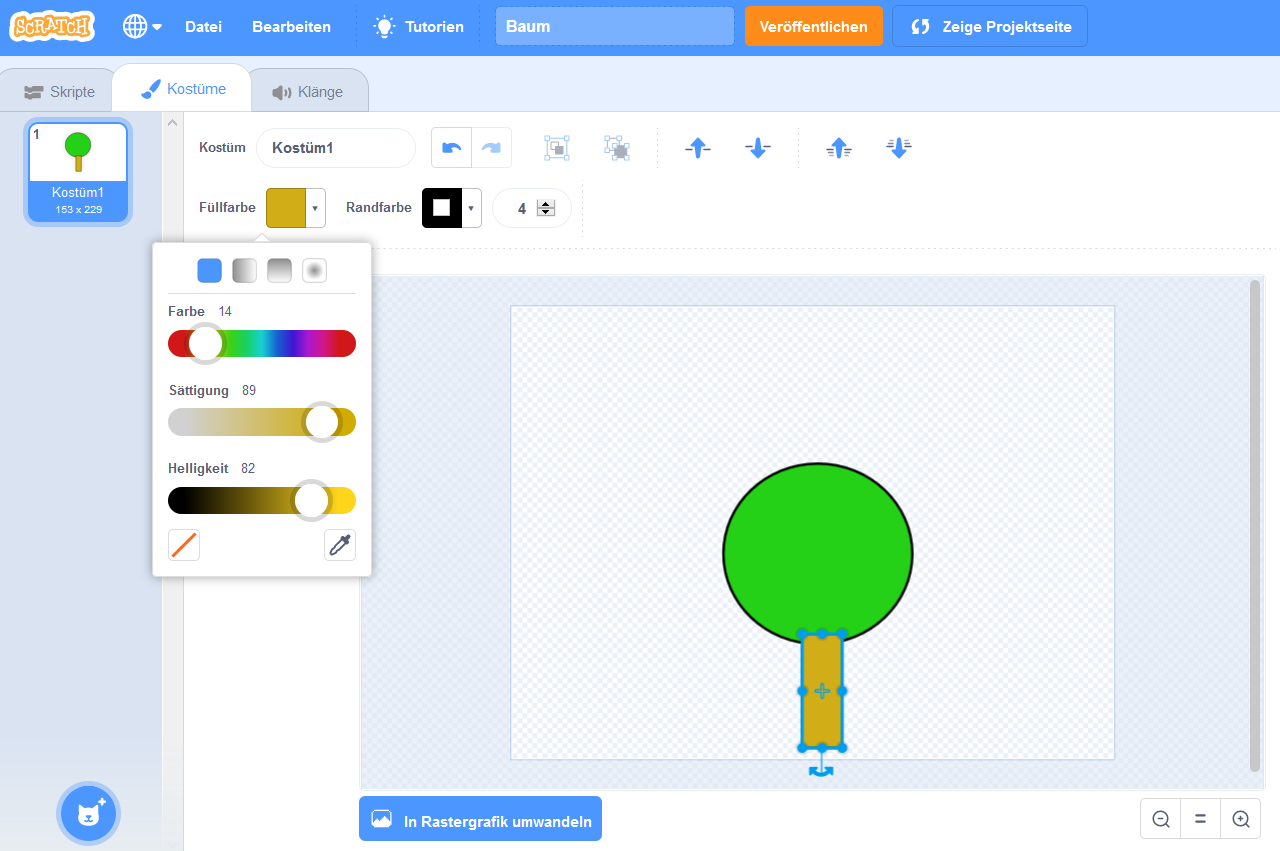


Abbildung 4 Hier wird eine Figur mit dem Namen „Baum“ erstellt.

Die Ergebnisse werden dann in die Programmierung über ihren „Kostüm“-Namen eingebunden.

Eine gänzlich andere Möglichkeit besteht darin, die Bilder real zu zeichnen, zu malen oder zu sprayen. Diese Werke werden dann entweder eingescannt oder mit einer Kamera (am besten auf einem Stativ) fotografiert. Anschließend können diese Bilder vom Computer in das Programm hochgeladen werden. Damit ist es möglich sowohl Teilnehmer\*innen wie auch Mitarbeiter\*innen einzubinden, die mit dem Thema Programmierung nicht viel anfangen können und sich eventuell ausgegrenzt fühlen. Grundsätzlich ist dies auch mit Fotos von Teilnehmer\*innen möglich. Hierbei ist zu beachten, dass die Fotos tatsächlich auf amerikanische Server hochgeladen werden und sich alle Beteiligten über die Datenschutzbestimmungen des Servers informieren.

**Ende des Workshops**

Am letzten Tag ist darauf zu achten, dass die Jung-Programmierer\*innen ihre Projekte zu Ende bringen. Daher wird zu Beginn des Tages geklärt, was noch zu tun ist. Neue Ideen können nur noch aufgenommen werden, wenn sie dazu dienen, bestehende Probleme zu lösen und in der übrig gebliebenen Zeit noch realisierbar sind.

Eine große **Feedbackrunde** mit allen Beteiligten bietet die Möglichkeit den Workshop zu verbessern und auszubauen. Mögliche methodische Hilfsmittel:

Hammer und Schraubenzieher wechseln durch die Runde: Was war „Hammer?“ an was muss noch „geschraubt“ werden?

Zuwerfen eines (Jonglier)-Balls der Fänger gibt sein sein Feedback an die Runde.

Am Ende des Workshops wird das gemeinsam erstellte Spiel gespielt und Ideen und Verbesserungen für einen neuen Spiele-Workshop ausgetauscht.

Autoren: Harald Waltz & Edgar Martinez Erstellt im Dezember 2020