



# ANALOGES PROGRAMMIEREN

## Algorithmisches Zeichnen



Übung oder  
Warm-up



ca. 12 min.




6 Jahre+



Vorwissen:  
Keines



Durch diese Übung wird verdeutlicht, dass Handlungsanweisungen sehr spezifisch sein müssen, um das gleiche Ergebnis zu erzielen. Das Spiel kann frontal oder in 2er Teams gespielt werden. Eine Person zeichnet/entwirft zunächst eine Form und versucht dann durch genaue Beschreibungen / Anweisungen seine Teampartnerin / seinen Teampartner so anzuleiten, dass diese / dieser die gleiche Form zeichnen kann.

 Diese Übung eignet sich auch hervorragend als methodisches Training in Bezug auf die Bearbeitung von Aufgaben, da die Kinder, um die Aufgaben richtig lösen zu können, zunächst alle Arbeitsanweisungen lesen müssen, bevor sie die Befehlsfolge bzw. den Algorithmus richtig umsetzen können.



### Lernkompetenzen

- > Verständnis von algorithmischen Strukturen
- > Kennen von Voraussetzungen für erfolgreiches Programmieren

# Vorbereitungen für dein Algorithmisches Zeichnen

## Benötigtes Werkzeug und Material

- > Buntstifte oder Filzstifte
- > Papier
- > Radierer
- > Spitzer
- > Ausdrücke der Vorlagenblätter als Anschauungsmaterial. (optional)



## 1. Spielvorbereitung

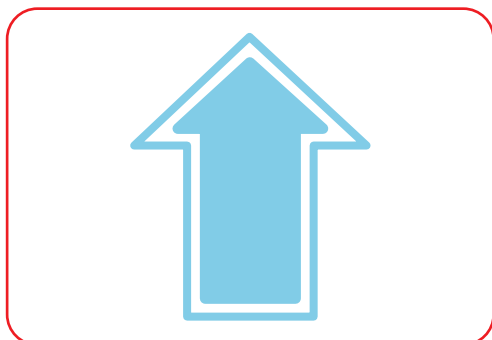
Die Durchführung des Spiels kann sowohl eigenverantwortlich in Zweiergruppen oder als auch durch eine übergeordnete Spielleitung gesteuert werden.

### Schritt 01: Aufteilung der Kinder

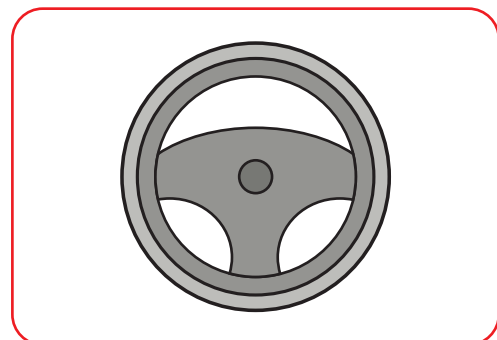
Wird die Arbeit in Kleingruppen bevorzugt, so sollten die Kinder zunächst in die jeweilige Gruppenstärke (am besten 2er oder 3er Teams) aufgeteilt werden. Eventuell braucht es mehr Lenkradsymbole.

### Schritt 02: Rollenverteilung

Sofern keine übergeordnete Spielleitung genutzt werden soll, müssen die Kinder nun zunächst in die jeweiligen aus dem Pair Programming bekannten Rollen Navigation und Programmierung sowie deren Aufgaben eingeführt werden. Weitere Informationen zum Thema Pair Programming lassen sich in unseren OER Materialien finden.



- > Rolle Navigation (1 Kind)  
Der Pfeil ist dein Erkennungsmerkmal.  
Deine Aufgabe ist es, den Überblick zu behalten und die Befehlsfolge vorzulesen.




- > Rolle Programmierung (min. 1 Kind)  
Das Lenkrad ist dein Erkennungsmerkmal.  
Du bist für die praktische Umsetzung zuständig und führst die Befehlsfolge aus.

## 2. Spielanleitung


**Ziel des Spiels:** Algorithmen sind Abfolgen von Anweisungen. In diesem Spiel sollen die Kinder anhand von Befehlsabfolgen unterschiedliche Formen aufmalen. Danach werden die jeweiligen Befehlsfolgen und die daraus resultierenden Ergebnisse verglichen und besprochen.

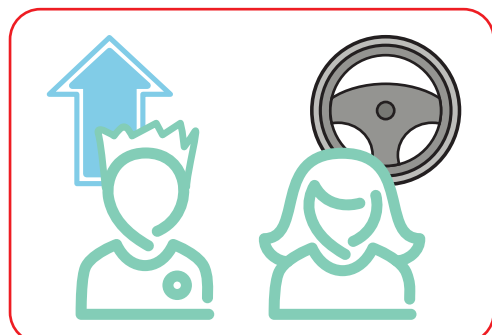
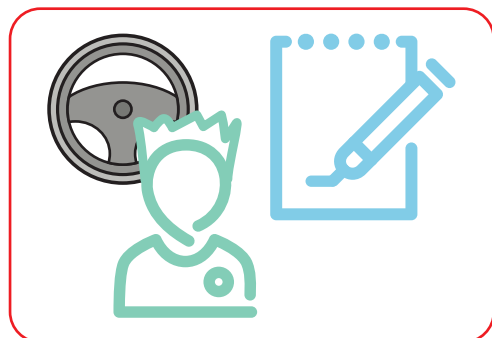
**Schritt 01:** Die Navigatorin / der Navigator überlegt sich eine Befehlsabfolge, liest diese zunächst in Gänge vor und überprüft, ob jede Programmiererin / jeder Programmierer die Aufgabenstellung verstanden hat. Gibt es Fragen zu den Anweisungen, werden diese geklärt.

 Du kannst die Anweisungen auch noch einmal einzeln vorlesen, wenn die anderen anfangen zu zeichnen. Es sollte aber in jedem Fall die ganze Befehlsfolge bereits bekannt sein.

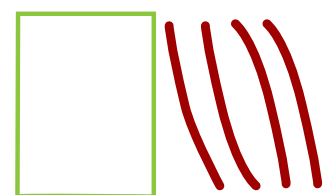
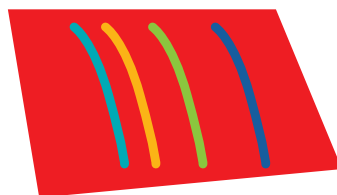
**Schritt 02:** Die Programmiererin / der Programmierer führen die Befehlsfolge aus und malen das, nach ihrem Verständnis richtige Bild.

**Schritt 03:** Haben alle Teammitglieder ein Ergebnis gemalt, wird die Rolle Navigation neu verteilt.

 Achtet darauf, dass am Ende des Spiels jedes Kind aus der Gruppe einmal navigieren durfte.



So könnte eine erdachte Befehlsfolge und die daraus entstandenen Zeichnungen aussehen. Ein Bild scheint irgendwie nicht zu passen. Findest du heraus, welches es ist?



**Schritt 01:** Zeichne ein Viereck.

**Schritt 02:** Zeichne Linien, die sich nicht berühren.

**Schritt 03:** Benutze für die Linien eine Farbe.

### 3. Auswertung/Reflexion



Wie bereits eingangs beschrieben, sollte der Reflexionsphase im Anschluss als wichtiges Element zur Aufarbeitung des Geschehens ein wenig Zeit eingeräumt werden.

Neben Fragen wie:

> „Wie hat es in eurer Gruppe geklappt?“

> „Warum hat es gut/schlecht funktioniert?“,

die sich vor allem mit der Thematik der Funktionalität und Arbeitsweise der Gruppen befassen, sollte man vor allem folgender Frage nachgehen.

> „Was hat das Spiel mit Programmierung zu tun?“

Nun ein ganze Menge, auch wenn es zunächst vielleicht nicht so aussehen mag. Doch um dieser Frage eine Antwort geben zu können, sollten wir vielleicht zunächst noch einmal ein paar grundsätzliche Dinge klären.

Was ist ein Algorithmus und woraus besteht er?

Nun wie oben bereits beschrieben, definiert der Begriff eine Folge von Anweisungen mit denen ein bestimmtes Problem gelöst werden kann. Dieser Definition entspricht so ziemlich jedes Kochrezept oder jede Bauanleitung, wenn man so möchte. In unserem Fall liegt das Problem darin, dass ein Teil der jeweiligen Gruppe ein Bild malen soll. Doch wie soll das Bild gemalt werden?

Es müssen also Arbeitsanweisungen/Befehlsfolgen formuliert werden, damit dieses Problem gelöst werden kann. Betrachtet man nun die Ergebnisse und bespricht diese mit den Kindern,

sollten zwei grundsätzliche Erkenntnisse entdeckt werden können.

1. Es gibt mehr als nur eine Lösung.
2. Je eindeutiger die Anweisungen bzw. die Befehlsabfolgen formuliert werden, desto einheitlicher werden die Bilder. Man spricht hierbei von "wohldefinierten" Algorithmen.

Wenn die Kinder im Vorfeld das Farbcode- oder Decodierungsspiel gemacht haben, so sollten sie nun die grundsätzlichen Prinzipien einer Kommunikation zwischen Mensch und Computer / Maschine verstehen können.

1. Computer/Maschinen benötigen eindeutige Befehle, um handeln zu können.
2. Jeder Befehl ist mit einer bestimmten, klar definierten Bedeutung versehen.
3. Ein Programm besteht aus aneinandergereihten Befehlen. Diese Befehlsfolgen heißen Algorithmen.

Als Ergänzung zum Thema Algorithmus bietet sich auch unser Zahnputzspiel an. Es finden sich noch jede Menge weitere Beispiele für Algorithmen in unserem Alltag. Also, mach dich auf die Suche.

Zur weiteren Vertiefung solltest du nun einmal unsere Übungsangebote zu ScratchJr z.B. Erstes Programmieren oder Scratch z.B. Mein erstes Fangspiel anschauen. Du findest die Sachen auf unserer Webseite.




**Viel Spaß beim  
Ausprobieren!**



# Vorlagen-Blatt

## Beispiele für einen Malalgorithmus

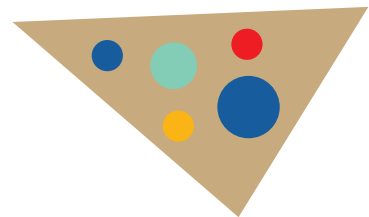
Diese Beispiele kannst du nutzen um den Einstieg in das Spiel zu erleichtern. Anhand der folgenden vier Befehlsabfolgen sind vier Formen entstanden. Welche Anweisungsabfolge passt zu welcher Form?

 **Aufgabe:** Lies dir die vier Befehlsabfolgen durch und verbinde sie mit den dazu passenden Bildern.

**Schritt 01:** Zeichne ein Viereck.  
**Schritt 02:** Zeichne Linien, die sich nicht berühren.  
**Schritt 03:** Benutze für die Linien eine Farbe.



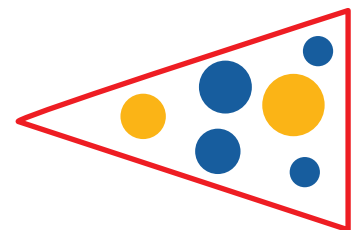
**Schritt 01:** Zeichne ein Dreieck.  
**Schritt 02:** Zeichne Kreise in drei Größen hinein.  
**Schritt 03:** Benutze für die Kreise zwei Farben.




**Schritt 01:** Zeichne einen Kreis.  
**Schritt 02:** Zeichne einzelne Linien hinein.  
**Schritt 03:** Benutze für die Linien mehrere Farben.



**Schritt 01:** Zeichne ein Dreieck.  
**Schritt 02:** Zeichne Kreise hinein.  
**Schritt 03:** Benutze für die Kreise vier Farben.



 **Aufgabe:** Erfinde eigene Befehlsabfolgen und lies sie der Gruppe vor. Es können auch mehr als 3 Schritte sein.

**Schritt 01:** ...  
**Schritt 02:** ...  
**Schritt 03:** ...

## Vorlagen-Blatt Pair Programming Symbole



### **Programmierer\*in - Symbol**

Bekommt das Kind, das gerade der Driver, also der/die Programmierer\*in ist, als Erkennungszeichen.

### **Navigator\*in - Symbol**

Bekommt das Kind, das gerade der/die Navigator\*in, also der/die Richtungsweiser\*in ist, als Erkennungszeichen.