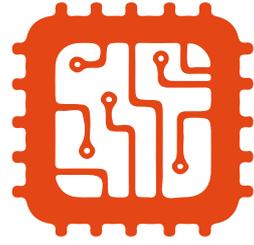


# Patching



# Modul 5

---

# Impressum

---

## Herausgeber

Landesanstalt für Kommunikation Baden-Württemberg (LFK)  
Anstalt des öffentlichen Rechts,  
vertreten durch den Präsidenten Dr. Wolfgang Kreißig  
Reinsburgstraße 27  
70178 Stuttgart

Gefördert durch das Ministerium für Kultus, Jugend und Sport  
Baden-Württemberg

## Autor

Chris Binder

## Redaktion

Laura Jaenicke, Landesanstalt für Kommunikation BW (LFK)  
Dejan Simonović, Stadtmedienzentrum Stuttgart am LMZ BW

## Design und Layout

Jana Falkner

## Illustrationen

Ilan Backmann

Stuttgart | September, 2022

2. Auflage

Lizenz CC-BY-SA 4.0

# Modul 5 Impressum

---

Die Handreichung steht unter <https://games-im-unterricht.de/toolkit/> auch als PDF zur Verfügung.

Die Rechte aller verwendeten Grafiken und Bilder liegen beim Stadtmedienzentrum Stuttgart.

Das Werk enthält Screenshots aus dem verwendeten Programm Minetest. Diese sind selbst erstellt und werden im Sinne eines Zitats zu Bildungszwecken genutzt.

Beschreibung	5
Einbindung in die Story	6
Rahmenbedingungen	6
Vorbereitung	8
Durchführung Modul 5 Patching	15
5.1 Analoges Programmieren	15
5.2 Chip (Bot) programmieren	17
5.3 Eigene Problemstellungen definieren	19
5.4 Chip (Bot) durchs Labyrinth schicken	21
Varianten/Abänderungen	22
Hilfestellungen	22

Hallo, ich bin  
Platina Toolkid!





## Beschreibung

### Modul 5 | Patching

#### Modul 5.1

Analoges  
Programmieren

#### Modul 5.2

Chip (Bot)  
programmieren

#### Modul 5.3

Eigene Problem-  
stellungen definieren

#### Modul 5.4

Chip (Bot) durchs  
Labyrinth schicken

Minetest ist ein freies Erkundungs- und Bauspiel ohne bestimmtes Ziel, in dem die Spielwelt aus quadratischen Blöcken (Voxeln) aufgebaut ist. Als Einzel- und Mehrspielerspiel bietet es vielfältige Anknüpfungspunkte im Bildungsbereich und als virtueller Interaktions- und Lehr-/Lernraum. Mithilfe von Modifikationen lässt sich die Spiellogik umfangreich erweitern.

In diesem Modul werden Visual Bots programmiert, die als virtuelle Bewohner von Minetest die Befehle der Spieler:innen ausführen. Es können Befehle ausgeführt und aneinandergereiht werden. Es soll vermittelt werden, wie Probleme (Struktur bauen, Hindernis überwinden) durch programmierbare Einheiten gelöst werden können. Dabei kommen Schleifen und Funktionen zum Einsatz. Die Visual Bots führen die Befehle aus und interagieren mit der Umwelt. Die Interaktion ist visuell und auf nachvollziehbare Weise in die virtuelle Umgebung von Minetest eingebunden.

Spieleransicht Minetest



Modul 5 des Toolkits ist auf den Bildungsplan Baden-Württemberg für den Aufbaukurs Informatik in Klasse 7 ausgerichtet. Dabei werden aus den prozessbezogenen Kompetenzen Strukturieren und Vernetzen, Implementieren, Kooperativ arbeiten als auch Testen und Reflektieren abgedeckt. Darüber hinaus vermittelt das Modul alle Kompetenzen des inhaltsbezogenen Bereichs Algorithmen (siehe Einleitung, S. 6&7).

## Einbindung in die Story

---

Platina und Chip besuchen die virtuelle Welt, in der sie ganz anders und „blockig“ aussehen. Sie finden ein Labyrinth, das Platina nicht betreten kann. Deshalb wird Chip losgeschickt, um die andere Seite zu erreichen. Chip muss so programmiert werden, dass er den richtigen Weg durch das Labyrinth findet.



## Rahmenbedingungen

---



### Zeit

4 Untermodule à 90 Minuten

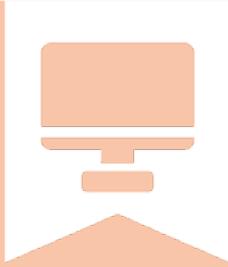
Das Modul besteht aus 4 Untermodulen, welche jeweils auf 90 min ausgelegt sind.



### Raum

Präsenz- oder Fernunterricht

Dieses Modul eignet sich sowohl für den Präsenz-, als auch für den Onlineunterricht. Um den Onlineunterricht möglichst immersiv zu gestalten, bietet sich die Nutzung einer gleichzeitigen Audiokonferenz an.



# Technik und Materialien

Geräte : PC/Tablet

Pro Schüler:in ein Android-Tablet oder PC, das mit dem Internet kommunizieren kann.

## **Online:**

Virtuelles Klassenzimmer; Lehraccount und SuS-Login werden vom Kooperationspartner bereitgestellt

## **Offline:**

Lokaler Server mit den folgenden Voraussetzungen:

- Wlan-fähig
- Windows, 7, 8, 10, 11

→ Wird das Modul online genutzt, sollten die Ports 30000-31000 für UDP im Schulnetz geöffnet sein

→ Jedes Schulnetz ist anders: Wenden Sie sich bei Problemen an Ihre:n Netzwerkadministrator:in

## Vorbereitung

---

Der Abschnitt Hilfestellungen (siehe S. 22) bietet eine Einführung in die Grundlagen, das Inventar und pädagogische Werkzeuge von Minetest.

Darüber hinaus bieten folgende Videos knappe Einführungen in Minetest:

### Grundlagen



<https://vimeo.com/476209254>

### Inventar



<https://vimeo.com/476209277>

### EduTest



<https://vimeo.com/529216016>

### Server-Login



<https://vimeo.com/499348830>

Das Landesmedienzentrum Baden-Württemberg verwaltet im Rahmen des Pilotprojekts BLOCKALOT virtuelle Klassenzimmer für Minetest, die Ihnen für den Unterricht und ggf. auch für außerschulische Bildungsprojekte kostenfrei zur Verfügung gestellt werden. Sie können hier ein Klassenzimmer für Modul 5 beantragen:



<https://www.blockalot.de>

Innerhalb der Spielwelt haben die Schüler:innen ihre eigenen Avatare. Durch die virtuelle Lernumgebung können Sie die Schüler:innen an verschiedenen Orten versammeln und so Plenumsdiskussionen oder Gruppenarbeiten abbilden.

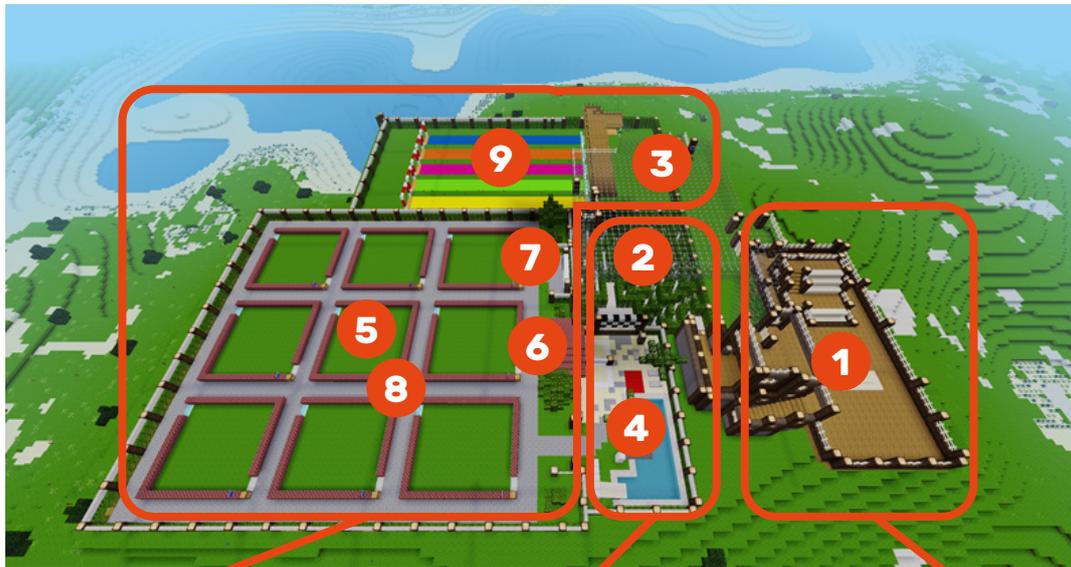
Die Schüler:innen können im Inventar ihr Aussehen über Skins selbst festlegen und so einen persönlichen und wiedererkennbaren Avatar steuern.

Wenn Sie das Modul online an Computern verwenden, können Sie vom Landesmedienzentrum ein vorkonfiguriertes Paket erhalten, mit dem Sie parallel zu Minetest eine Sprachkonferenz mit Unterräumen abhalten können. Dies wird über die Open-Source-Audiokonferenz-Software Mumble ermöglicht.

Schauen Sie sich den Minetest-Level auf der für Sie bereitgestellten Welt oder mit Hilfe des lokalen Webservers an.

## Aufbau Minetest Level

Der Level ist in folgende Bereiche aufgeteilt:



Baubereich:  
Untermodule 5.2 – 5.4

Labyrinth-Bereich:  
Unterm modul 5.1

Lehrbereich:  
Start

Im Detail befinden sich im Level folgende Strukturen:

1

Lehrbereich  
mit Podium



2

Labyrinth für eine/  
einen Schüler:in



3

Glas-Ebene, von der  
aus ein Weg durch  
das Labyrinth gese-  
hen werden kann



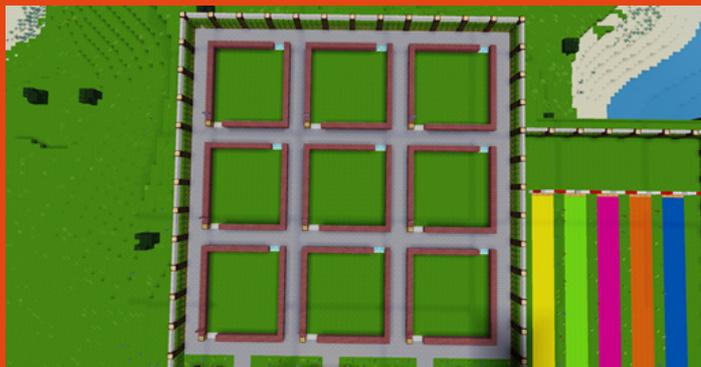
4

Ruhebereich für die/  
den Gewinner:in des  
Labyrinths



5

Bauplätze für  
Gruppen



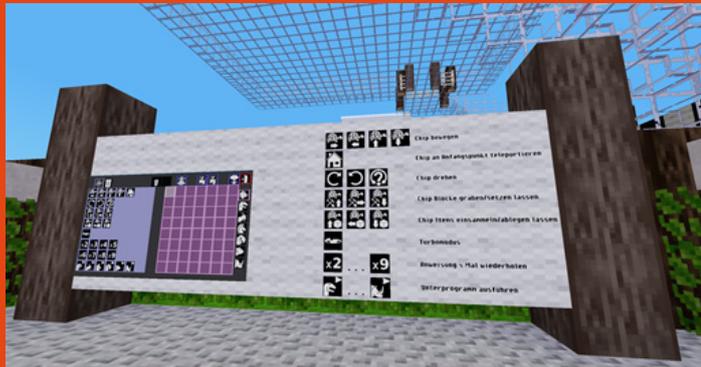
6

Lagerplatz für Chip-Bots



7

Erklärungstafel mit Hilfestellungen für die Programmierung von Chip



8

Gruppenbauplatz mit Schild und Materialtruhe



9

(Bonus) Rennstrecke



10

(Bonus) Hindernislager für Rennstrecke



Chip in Minetest

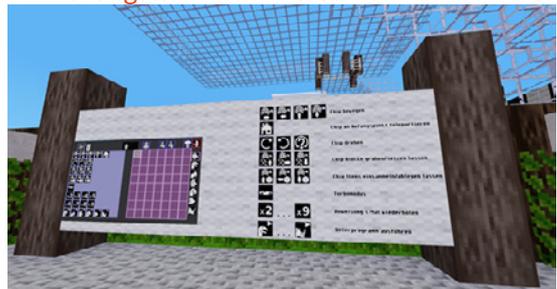


In den Untermodulen 5.1 bis 5.4 werden die Schüler:innen mit Chip interagieren, der in der Spielwelt platziert und programmiert werden kann. Per Rechtsklick auf den Bot kann dieser visuell programmiert werden. Dabei werden die möglichen Befehle (links) durch Klicken nacheinander in das Programm geladen. Mit Klick auf den Startknopf beginnt der Bot sein Programm.

Eine Erklärungstafel, die im Spiel als Referenz dient, befindet sich im Baubereich und kann von den Schüler:innen als Referenz herangezogen werden.

Durch Klicken auf einzelne Blöcke werden diese in das rechte Feld übertragen, in dem der Code gespeichert wird.

Erklärungstafel in Minetest



## Programmierung von Chip

Durch Rechtsklick auf Chip öffnet sich die Programmieroberfläche.

Sie beinhaltet folgende Codeblöcke:

The image shows a grid of programming blocks for a chip. The blocks are arranged in a grid and include:

- Navigation blocks: Turtle icons with arrows pointing up, down, left, and right, and a house icon.
- Rotation blocks: Circular arrows and a question mark icon.
- Block manipulation blocks: Turtle icons with pickaxe and block icons, and arrows pointing up, down, left, and right.
- Turbomodus block: A car icon.
- Repetition blocks: Blocks labeled x2, x3, x4, x5, x6, x7, x8, and x9.
- Subprogram blocks: Blocks with icons of a dinosaur, a goat, a horse, a yin-yang, a dog, and a cat.

Red lines connect these blocks to descriptive text boxes on the right:

- Chip bewegen und auf Anfangsposition zurücksetzen
- Chip drehen
- Chip Blöcke abbauen und setzen lassen
- Turbomodus einschalten
- Anweisung x-mal wiederholen  
→ Dieser Block muss hinter der zu wiederholenden Anweisung platziert werden!
- Unterprogramme ausführen



Durch Klick auf (Ausführen) arbeitet Chip seine Anweisungen ab.



Anweisungen können wieder gelöscht werden, indem die Mülleimertaste geklickt wird oder bestimmte Blöcke auf das Feld rechts neben dem Mülleimer gezogen werden.



Mit der Löschen-Taste wird das derzeitige Programm gelöscht.



Verlassen Sie die Programmieroberfläche durch die Escape-Taste (ESC) oder durch einen Klick auf das Verlassen-Symbol

Es empfiehlt sich für mehrmaliges Testen des Programms, an dessen Ende den Block einzufügen, der Chip wieder an seinen Startpunkt zurücksetzt. So kann das Programm getestet und angepasst werden, während Chip immer wieder von seiner Ursprungsposition startet. Bleibt Chip dann an einer Stelle des Labyrinths hängen, wird er automatisch wieder an seine Ausgangsposition zurückkehren. Dadurch wird das Debugging durch Codeveränderungen erleichtert.

## Durchführung Modul 5 Patching

Patching besteht aus vier Untermodulen. Es wird mit einfachen Übungen angefangen, die ein analoges Programmieren simulieren. Über das Gestalten von eigenen Problemstellungen wird zu anspruchsvolleren Programmierungen übergeleitet.

Die Untermodule bauen zwar aufeinander auf, können aber auch ohne Vorwissen einzeln bearbeitet werden. Es ist wichtig, notwendige Zeit für die Grundlagen von Minetest und das Kennenlernen der Chip-Bots einzuplanen.

### 5.1 Analoges Programmieren

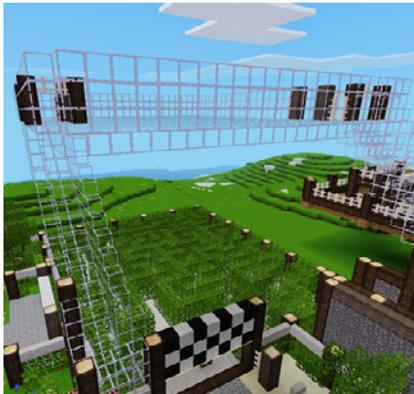
Beim analogen Programmieren durchläuft ein:e Schüler:in das Labyrinth und wird auf Zuruf der Mitschüler:innen gesteuert. Aus diesen Erfahrungen heraus wird in Gruppen eine Sprache entwickelt, mit der Schüler:innen das Labyrinth effizienter durchqueren können.

Wenn die Schüler:innen den Level zum ersten Mal betreten, erscheinen sie im Lehrbereich. Diesen können Sie nutzen, um grundlegende Regeln des virtuellen Miteinanders zu besprechen und die Grundlagen der Steuerung zu vermitteln.

Kommunikation gar nicht funktioniert, können Sie die Person im Labyrinth über die Lehrerkonsole EduTest (siehe S. 8) zu sich teleportieren. Im besten Fall einigen sich die Schüler:innen ohne Absprache auf Befehle, die der Person im Labyrinth dabei helfen, den richtigen Weg zu finden.

Rufen Sie die Schüler:innen zum Lehrbereich zurück und erörtern Sie mit ihnen, wie die Person im Labyrinth angesprochen werden muss, um den effizientesten Weg zu finden. Sie können dabei über die Befehle und die Sprache diskutieren, welche die Schüler:innen verwendet haben. Wieso war diese effizient oder nicht?

Glasplattform und Labyrinth



Aus der Gruppe kann ein:e Schüler:in ausgewählt werden, welche:r das Labyrinth durchqueren soll. Dazu geht die ausgewählte Person die Treppe herunter und gelangt in den Labyrinthbereich. Die übrige Gruppe geht die Treppe hoch in den Glasbereich, von dem aus man das Labyrinth und den korrekten Weg im Blick hat. Die Gruppe im Glasbereich kann sich nun kurz mit der Struktur des Labyrinths vertraut machen. Nun wird durch Zuruf die Person im Labyrinth durch eben dieses gelotst. Falls die

Hierbei gibt es viele mögliche Antworten, denen aber die Klarheit und Unmissverständlichkeit der Befehle eigen sein sollte. Im nächsten Schritt sollen die Schüler:innen in Gruppen eigene Sprachen entwickeln. Dazu können Sie in den Baubereich umziehen und die Schüler:innen ihren jeweiligen Baugebieten zuweisen. Die Schüler:innen können beispielsweise Schilder verwenden, um ihre Ideen aufzuschreiben. Sie können die Klasse dabei unterstützen, indem sie darauf hinweisen, welche Anweisungen für die erste Durchquerung des Labyrinthes besonders hilfreich waren.

Nach dieser Arbeitsphase sollten die Ergebnisse gemeinsam besprochen werden. Um die Sprachen zu evaluieren, bietet es sich an, einige Schüler:innen anhand der entwickelten Sprache durch das Labyrinth zu schicken und die verwendeten Befehle zu diskutieren.

## Verlaufsplan | 5.1 Analoges Programmieren

Dauer	Phase und Unterrichtsform	Inhalt	Material und Technik
10min	Einstieg	SuS und Lehrkraft betreten die Welt und machen sich mit der Steuerung vertraut.	
5 min	Einstieg, Unterrichtsgespräch	Gruppeneinteilung	freiwillig/Zufall
10 min	Erarbeitung I, Gruppenarbeit	Ausgewählte:r Schüler:in wird von Gruppe gesteuert → findet Labyrinthausgang → findet Labyrinthausgang nicht (Teleport zur Gruppe)	Evtl. Timer benutzen
20min	Problematisierung, Unterrichtsgespräch	Im Klassenzimmer oder Diskussionsplattform: Wie muss die Person angesprochen werden, damit sie möglichst effizient den Ausgang erreicht?	
20 min	Reflexion, Gruppenarbeit	SuS sollen in Gruppen eine effiziente Programmiersprache für die Labyrinthdurchquerung entwickeln. Es dürfen später nur exakt die entwickelten Befehle gegeben werden.	Schilder und Bücher
25 min	Sicherung, Schülerpräsentation	SuS der Gruppen werden durch das Labyrinth geschickt, Programmiersprache der jeweiligen Gruppe wird diskutiert	

## Anknüpfungspunkte Informatik | 5.1 Analoges Programmieren

Informatik	Modul 1
(Programmier-)Sprache	Natürliche Sprachen, Plansprachen, etc.
Syntax	Die Grammatik einer (Programmier-)Sprache. Wird diese verletzt, kann das Programm nicht richtig ausgeführt werden

## 5.2 Chip (Bot) programmieren

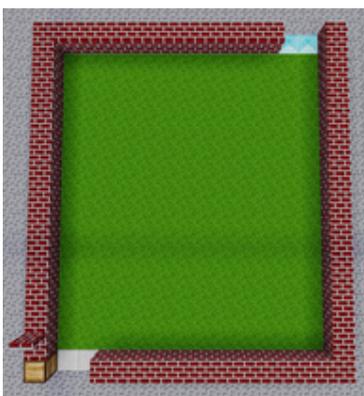
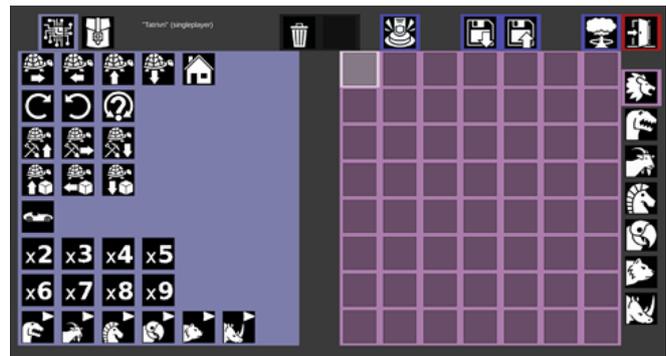
Die Schüler:innen werden Gruppen zugeordnet und sollen auf vordefinierten Feldern erste Erfahrungen in der Programmierung der Bots machen. Innerhalb dieser

Felder darf beliebig umgestaltet werden, um die Ziele zu erreichen.



Die Schüler:innen sollten sich mit jeweils einem Chip (Bot) ausstatten. Diese sind im Baubereich in einem Lagerraum verfügbar. Ein Bot kann, wie ein normaler Block, auf dem Boden platziert und dann über Rechtsklick programmiert werden.

Erklären Sie an der Tafel im Baubereich, wie die Bots programmiert werden können und welche Funktionen zur Verfügung stehen. Die Schüler:innen können bei Problemen jederzeit an der Tafel nachschauen, um sich die Syntax zu vergegenwärtigen.



Die Gruppen sollen sich zunächst auf dem freien Feld mit den möglichen Befehlen vertraut machen und Chip einfache Anweisungen geben. Dann soll dieser so programmiert werden, dass er sich von der silbernen Startfläche bis zur türkisgrünen Endfläche bewegt.

Sobald die Steuerung der Chip-Bots gut funktioniert, können Sie die Gruppen anweisen, Hindernisse aufzubauen, die Chip überwinden muss. Es empfehlen sich leicht-

te Hindernisse, die etwa einen Block hoch sind.

An der Tafel kann anschließend das Konzept der Wiederholungen (Schleifen) erklärt werden, mit der Chip eine Anweisung mehrfach ausführt. Als Differenzierung bieten sich außerdem die Unterprogramme an, die ihrerseits mit Anweisungen befüllt und aus dem Hauptprogramm heraus mehrmals gestartet werden können. Dazu können die jeweiligen Programmreiter mit Befehlen gefüllt und dann aus dem Hauptprogramm mit dem jeweiligen Programmblock aufgerufen werden.

Der zuvor aufgebaute Hindernisparcours soll nun erneut durchlaufen werden. Zuvor sollen die Schüler:innen die Anweisungen für Chip so optimieren, dass möglichst wenige Programmierblöcke verwendet werden (Refactoring). Am Ende der Arbeitsphase dürfen die Schüler:innen ihre Ergebnisse präsentieren und erläutern, wie sie den Programmcode vereinfacht haben.

### Verlaufsplan | 5.2 Chip (Bot) programmieren

Dauer	Phase und Unterrichtsform	Inhalt	Material und Technik
5-10 min	Einstieg	Grundlegende Erklärung der Bots	
15 min	Impuls, Gruppenarbeit	SuS erproben Bots: Einfache Strecke zurücklegen	Chip-Bot Material aus Truhe
20 min	Erarbeitung I, Gruppenarbeit	Aufgabe: Baut für Chip ein Hindernis auf, das er abbauen muss, um zum Ziel zu gelangen → Differenzierung: Baut einen Abgrund, den Chip mit einem Block ausfüllt, so dass Ihr den Abgrund überqueren könnt.	
10 min	Impuls II, Unterrichtsgespräch	Schleifen (Wiederholungen) gemeinsam an der virtuellen Tafel erarbeiten	Virtuelle Tafel
15 min	Problematisierung, Gruppenarbeit	Aufgabe: Optimiert Chip so, dass Ihr möglichst wenige Programmierblöcke verwenden müsst.	
10 min	Schülerpräsentation	SuS präsentieren Chips Programmierung und erläutern, wo sie den Code optimiert haben	

## Anknüpfungspunkte Informatik | 5.2 Chip (Bot) programmieren

Informatik	Modul 5.2
Schleifen	Wiederholen einer Anweisungen, um diese nicht jedes Mal explizit nennen zu müssen
Refactoring	Vereinfachen und Entschlacken des Programmcodes

### 5.3 Eigene Problemstellungen definieren

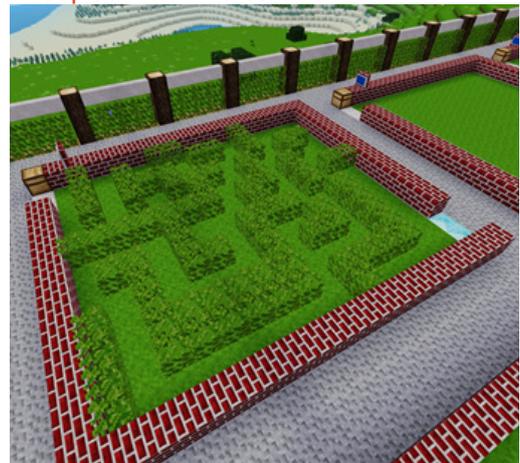
Die Schüler:innen sollen innerhalb der Gruppen selbst Labyrinth gestalten. Dazu werden vordefinierte Bauplätze bereitgestellt. Am Anfang sollen die Wände des Labyrinths nur einen Block hoch sein, damit die Programmierung von Chip leichter ausfällt.

Stellen Sie den Schüler:innen vorab individualisiertes Material zur Verfügung oder lassen Sie sie das Standardmaterial aus den Kästen benutzen, die am Eingang jeder Bauplätze stehen. Die Gruppen sollen nun gemeinsam herausfordernde Labyrinth gestalten. Als Differenzierungsaufgabe können die Gruppen später ihre Strukturen in die Höhe ausbauen. Auch hier sollte als letzte Anweisung der „Teleportiere dich nach Hause“-Block in der Programmierung stehen, damit das Testen von Chip möglichst einfach wiederholt werden kann.

Besprechen Sie gemeinsam mit der Gruppe die Entwürfe. Ausgehend von den Rückmeldungen können die Schüler:innen nun Verbesserungen umsetzen.

Am Ende der Phase stellen die Schüler:innen ihre Labyrinth vor und nennen selbst einige Herausforderungen, die bei der Programmierung von Chip beachtet werden müssten. Diese können unregelmäßige Abstände und häufige Richtungs- oder Höhenwechsel sein.

Bauplatz in Minetest



## Verlaufsplan | 5.3 eigene Problemstellungen definieren

Dauer	Phase und Unterrichtsform	Inhalt	Material und Technik
5 min	Einstieg	SuS erhalten Aufgabe und Material für Labyrinthbau	
20 min	Erarbeitung I, Gruppenarbeit	SuS sollen kollaborativ ein herausforderndes Labyrinth konstruieren	
20 min	Reflexion, Unterrichtsgespräch	Die Labyrinthwürfe werden gemeinsam besprochen und Verbesserungsvorschläge und Anregungen eingebracht.	
25 min	Erarbeitung II, Gruppenarbeit	SuS überarbeiten ihre Laybrinthe mit den Rückmeldungen und stellen diese fertig →Differenzierung: Mehrere Blöcke hohes Labyrinth konstruieren, dessen Gänge jeweils 1 Block Durchmesser haben	
15 min	Reflexion, Schülerpräsentation/ Unterrichtsgespräch	SuS stellen ihr Labyrinth vor und nennen Herausforderungen für Chip	

## 5.4 Chip (Bot) durchs Labyrinth schicken

Den Schüler:innen werden Bots zur Verfügung gestellt, die sie so programmieren sollen, dass diese durch das Labyrinth navigiert werden können.

Zuerst sollten sie sich Notizen machen und vorhersehen, welche Bewegungen auszuführen sind. Dann können sie die ersten Programmierversuche wagen.

Zum Abschluss sollen die Gruppen ihre Ergebnisse präsentieren.

In diesem Untermodul können die Gruppen entweder ihre eigenen oder die Labyrinth anderer Gruppen bearbeiten. Erläutern Sie je nach Kenntnisstand kurz hilfreiche Funktionen der Bots (Wiederholungen/Unterprogramme/nach-hause-Teleportieren). Die Schüler:innen sollen nun gemeinsam die Programmierung ihrer Chip-Bots planen. Dabei können sie sich zuerst mit den Schildern aus der Truhe Notizen machen und so die Aufgabe in Teilschritte zerlegen. Aber auch ein direktes Loslegen und Debuggen von Chip ist ein möglicher Weg. Als Differenzierungsaufgabe können die Gruppen ihre Programmierungen vereinfachen (Refactoring) oder mit einem zweiten Bot einen eventuellen zweiten möglichen Weg finden, der schneller ist.

Zum Abschluss des Moduls können die Gruppen ihre Ergebnisse vorstellen und erläutern, welche Punkte bei der Programmierung besonders komplex in der Umsetzung waren. Indem Chip losgeschickt wird, ist der Programmcode auch für die restlichen Schüler:innen nachvollziehbar. Je nach Art der Herangehensweise der einzelnen Gruppen können die Vorzüge eines planenden Vorgehens und dem iterativen Ansatz besprochen werden.

Bauplatz mit Hindernissen und Chip



## Verlaufsplan | 5.4 Chip (Bot) durchs Labyrinth schicken

Dauer	Phase und Unterrichtsform	Inhalt	Material und Technik
10 min	Einstieg	Bots bereitstellen und Aufgabenstellung erläutern	
35 min	Erarbeitung, Gruppenarbeit	SuS programmieren Bots →Differenzierung: Programmiere Chip möglichst effizient und verringere so die Anzahl der benutzten Programmierblöcke	Chip-Bots Material aus Truhe
15 min	Reflexion	Anhand von 2-3 Gruppen wird die bisherige Programmierung erläutert; etwaige offene Fragen im Plenum beantwortet	
25 min	Reflexion, Schülerpräsentation	Die Gruppen stellen ihre Ergebnisse vor und erläutern ihre Herangehensweise an die Programmierung	

## Varianten/Abänderungen

- Anstatt eines Labyrinths, lassen sich auch Hindernisparcours aufbauen, die von Chip absolviert werden müssen. Chip kann auch fliegen und muss daher keinen Boden unter den Beinen haben. So lässt sich eine motivierende Geschicklichkeitsaufgabe nutzen, in der Chip beispielsweise einen fehlenden Block platzieren muss, bevor die Schüler:innen den Parcours durchqueren können.
- Die bereitgestellte Karte enthält Rennbahnen, die mit Hindernissen befüllt werden können. Die Schüler:innen können dann in Gruppen ihre Chip-Bots so programmieren, dass diese möglichst schnell die Zielinie erreichen.

## Hilfestellungen

Modul 5 Handbuch (separat)



Ein Projekt der

**LFK** • Die Medienanstalt für  
• Baden-Württemberg

Gefördert durch das



**Baden-Württemberg**

MINISTERIUM FÜR KULTUS, JUGEND UND SPORT



In Kooperation mit

