
MCreator: Erstelle deine eigene Mod Alpaka-Ball!

KidsLab.de gGmbH - MatzE, KingBBQ



Oktober 2023

Übersicht

Eigene Minecraft-Mod erstellen: Alpaka-Ball!	3
Einführung	3
Was brauche ich dafür?	3
Vorbereitung	4
Online-Version mit Links zu weiteren zugehörigen Dokumenten	4
Source-Code und weitere Dateien	4
Einführung in MCreator & unser Projekt	4
Unser Spiel: Alpaka-Ball	4
Mods, MCreator, Java - eine kurze Einführung	5
Was ist MCreator?	5
Mods?	5
Vorbereitung: Ein neues Projekt in MCreator erstellen	6
Übersicht über die Entwicklungsumgebung (IDE)	8
Nützliche Einstellungen im Spiel	9
Level 1: Der Ball für unser Spiel	10
Verhalten anpassen	15
Level 2: Tor! Oder: Da muss der Ball rein	19
Anzeige des Spielstandes - Wie viele Tore wurden geschossen?	22
Wie werden die Tore gezählt? Auslöser!	22
Was ist mit der gegnerischen Mannschaft?	25
Level 3: Tore / Punkte zählen	28
Punkte merken	28
Tore erkennen	29
Prozedur 1: Punkte zurücksetzen	31
Prozedur 2: Tore zählen	31
Jetzt nochmal für Orange	33
Level 4: Erweiterte Punkte-Anzeige / Punkte zurücksetzen	35
Level 5: Die Alpaka-Ball-Schläger	36
Erstelle einen neuen Gegenstand	36
Textur für den Schläger pixeln	39
Eigenschaften des Schlägers	40

Rückstoß	40
Level 6: Das Spielfeld: Die Arena!	42
Beispiel - so sieht unsere Arena aus	43
Level 7: Exportieren & Installieren der Mod	44
JAR-Datei exportieren	44
Mod Installieren	45



Abbildung 1: Jugend hackt schriftzug

Eigene Minecraft-Mod erstellen: Alpaka-Ball!

Minecraft ist nicht nur ein außergewöhnlich kreatives Computerspiel – es ist auch eines der erfolgreichsten Computerspiele der Welt und bei Groß und Klein gleichermaßen beliebt. Diese Begeisterung kann man nutzen, um Neues zu lernen: Erstelle deine eigene Mod, also eine Erweiterung für Minecraft, und damit dein “Spiel im Spiel”: Alpakaball!

Die Open-Source Entwicklungsumgebung MCreator (<https://mcreator.net/>) bietet dazu alles Nötige: ein einfacher Start, die Möglichkeit zum Pixeln der Blöcke und eine einfache Programmierung. Die ganze Mod kann man in 8 einzelnen Workshops (1-2h) oder als Tages-Workshops erstellen. **Lizenz** CC BY 4.0 KidsLab

Zielgruppe ab 10

Kategorie Coding, Gaming, Design

Einführung

“MCreator ist eine Open-Source-Software, mit der Minecraft Java Edition Mods, Bedrock Edition Add-Ons und Data Packs über eine intuitive, leicht zu erlernende Oberfläche oder mit einem integrierten Code-Editor erstellt werden können. Sie wird weltweit von Minecraft-Spielern, Mod-Entwicklern, in der Ausbildung und in MINT-Workshops eingesetzt.” (Quelle: <https://mcreator.net/>)

In diesem Video lernst du die Basics - in 30 Minuten ist MCreator installiert und die erste eigene Mod fertig: <https://youtu.be/QeViHXM0aFQ>

Was brauche ich dafür?

Software:	MCreator
Geräte:	Computer pro Teilnehmer*in & Mentor*in, Beamer
Internet:	MCreator braucht Internet
Kein Account:	MCreator Mods können auch ohne Minecraft Account getestet / gespielt werden
Personenzahl:	10 Teilnehmer*innen + 2 Mentor*innen

Vorbereitung

- MCreator auf allen Computern installieren (<https://mcreator.net/install>)
- Nach dem Installieren an jedem Computer ein neues MCreator Projekt erstellen. Beim ersten Mal werden weitere Dateien für MCreator heruntergeladen. Der initiale Download aller Bibliotheken dauert sonst weiterer 15-30 Minuten im Workshop.

Online-Version mit Links zu weiteren zugehörigen Dokumenten

Du findest diese Anleitung - zusammen mit allen Texturen und sonstigen Inhalten auch online:
<http://kidslab.de/handbuch/mcreator/index.html>

Source-Code und weitere Dateien

Alle Quell-Dateien und auch die fertige Mod aus diesem OER findest du auf GitHub: https://github.com/KidsLabDe/alpaka_ball_oer

Einführung in MCreator & unser Projekt

Unser Spiel: Alpaka-Ball

Aber was machen wir hier eigentlich? Mit Mods lässt sich Minecraft individuell erweitern, denn Minecraft lebt vom Prinzip “Spiel im Spiel” – also, dass man eigene Spiele in Minecraft spielt, die vom ursprünglichen Spielprinzip stark abweichen können. Gute Beispiele sind Bedwars oder Builder-Challenge.

Wir haben uns für unsere Spielidee an dem Sport [Quidditch \(aus Harry Potter\)](#) orientiert, der von Zauberern und Hexen gespielt wird. Um das Ganze perfekt in den [Jugend hackt](#)-Kontext einzubetten, haben wir es AlpakaBall genannt: Das [Maskottchen](#) von Jugend hackt ist ein Alpaka :)

Mods, MCreator, Java - eine kurze Einführung

Wenn du gerne Minecraft spielst, ist dir bestimmt schon aufgefallen, dass es verschiedene Versionen gibt: Es gibt eine Java- und eine Bedrock-Version. Die Java-Version ist die ursprüngliche Minecraft-Version, die "Notch", der Erfinder von Minecraft, 2009 programmiert hat.



Wusstest du schon...

Notch hat die erste Version in nur wenigen Tagen programmiert: Nach nur 6 Tagen, am 16. Mai 2009, erblickte die erste wirkliche Minecraft-Version 0.0.9a das Licht der Welt!

Was ist MCreator?

"MCreator ist eine Open-Source-Software, mit der Mods für die Minecraft Java Edition, Add-Ons für die Bedrock Edition und Data Packs über eine intuitive, leicht zu erlernende Oberfläche oder mit einem integrierten Code-Editor erstellt werden können. Sie wird weltweit von Minecraft-Spielern, Mod-Entwicklern, in der Ausbildung und in MINT-Workshops eingesetzt." (Quelle: <https://mcreator.net/>)

Mods?

Minecraft verdankt seinen großen Erfolg nicht nur den Entwicklern von Mojang – schon sehr früh konnten externe Programmierer eigene Erweiterungen für Minecraft hinzufügen, sogenannte Modifikationen oder kurz "[Mods](#)".

Mods gibt es nur in der Java-Version. Sie sind grundsätzlich kostenlos, und es gibt jede Menge davon: Auf [CurseForge](#) sind aktuell über 100.000 Mods gelistet!

So hat die Community das Spiel für andere Spieler immer weiterentwickelt und die verrücktesten Ideen umgesetzt – ganz kostenlos gibt es so immer neuen Spielspaß ohne jegliche In-App-Käufe oder ähnliches!

Die Mods in Minecraft Java sind jeweils eine sogenannte JAR-Datei (JAR, englisch für "Krug" oder Java ARchive). Wenn sie an der richtigen Stelle liegen, ist dein Minecraft gemoddet!



Lust mal eine Mod zu installieren?

Wir würden dir dazu den [ATLauncher](#) empfehlen – er macht es viel einfacher, mit den verschiedenen Minecraft-Versionen umzugehen und erleichtert die Installation von Mods. In diesem Video wirst du Schritt für Schritt durchgeführt – und danach kannst du es mal **richtig** krachen lassen :) [So installierst du deine erste Mod! \(WorldEdit\) – mit Knalleffekt :\)](#)

Vorbereitung: Ein neues Projekt in MCreator erstellen

Um loszulegen, öffne zuerst MCreator. Unabhängig von deinem Betriebssystem (Windows, macOS oder Linux) ist MCreator eine normale Anwendung, die wie jede andere gestartet werden kann.

Als erstes erstellst du nun ein neues Projekt:



Abbildung 2: Neues Projekt erstellen - Schritt 1

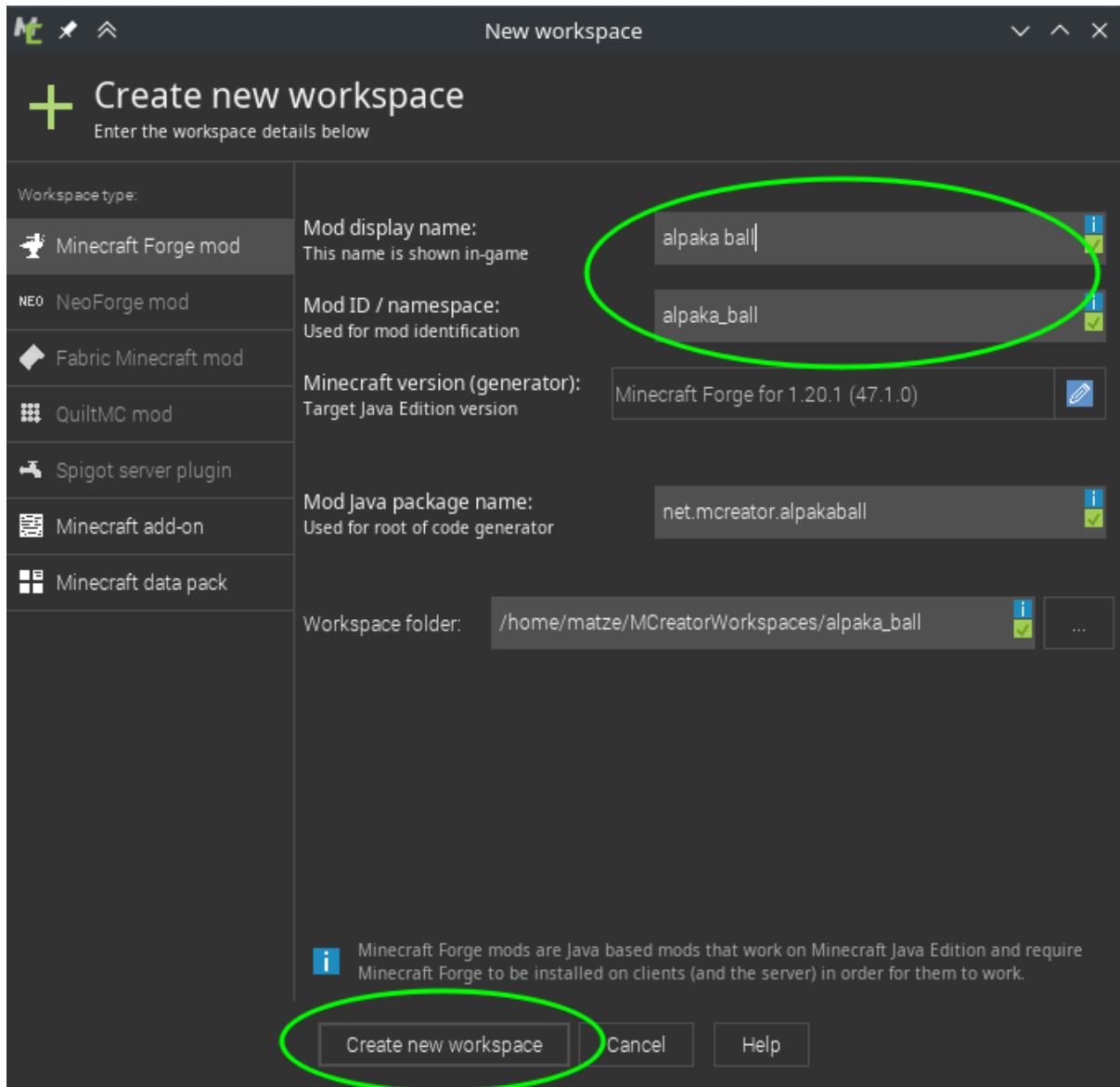


Abbildung 3: Neues Projekt erstellen - Schritt 2

Nun füllst du folgende Felder aus:

- **Mod-Anzeigename** - Wie möchtest du deine Mod nennen?
- **Mod-ID** - Interne Name (ohne Leerzeichen oder Sonderzeichen)
- **Minecraft-Version** - Mods müssen zur Minecraft-Version passen. Für dieses Tutorial wähle bitte "Minecraft Forge for 1.19.2".

Klicke dann auf "Neue Arbeitsumgebung erstellen". Es öffnet sich eine Webseite, die um eine Spende bittet.

Der Download und der erste Build können je nach Internetverbindung und Computer etwas Zeit in Anspruch nehmen. Dieser Schritt nimmt jedoch viele Aufgaben ab, die wir sonst manuell erledigen müssten. Also, geduldig warten :)

**Dauert es zu lange?**

Manchmal sieht es so aus, als ob der Vorgang ewig dauert, obwohl er bereits abgeschlossen ist. Wenn du eine Erfolgsmeldung in der Konsole siehst und nichts mehr passiert, kannst du das Popup-Fenster schließen, und alles sollte bereits installiert sein.

Übersicht über die Entwicklungsumgebung (IDE)

**IDE... was ist das?**

IDE steht für *Integrierte Entwicklungsumgebung* und ist ein Programm, das alle erforderlichen Tools für das Programmieren in einer einzigen Umgebung zusammenführt. Das ist unsere Arbeitsumgebung für dieses Tutorial.

Hier siehst du die Startseite der IDE und alle wichtigen Elemente:

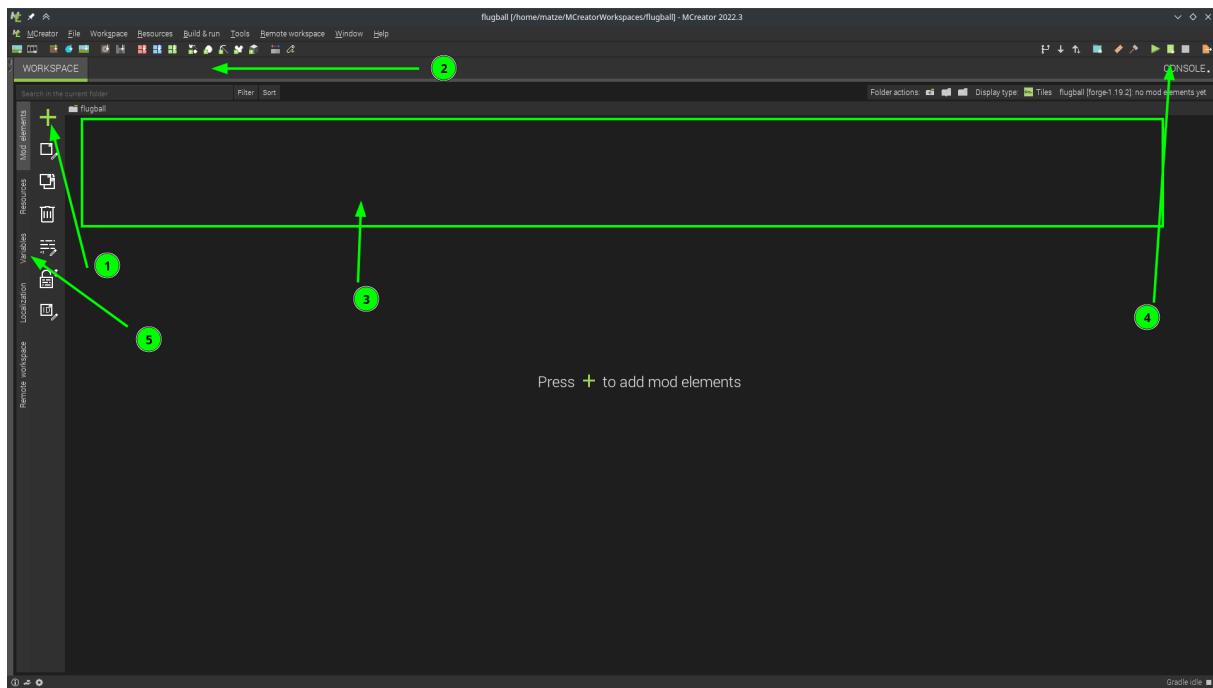


Abbildung 4: Übersicht der IDE: Die wichtigsten Elemente

1. Mit dem + Symbol kannst du neue Elemente erstellen, z.B. Blöcke, Tiere, oder Gegenstände.
2. In dieser Leiste siehst du, welche Elemente gerade zur Bearbeitung geöffnet sind.
3. Im großen, noch leeren Bereich bekommst du eine Übersicht über alle erstellten Elemente.
4. Hier sind einige Schaltflächen wichtig:
 - Wenn Symbole ausgegraut sind, können sie derzeit nicht verwendet werden.
 - Das grüne Dreieck startet Minecraft mit der neu erstellten Mod.
 - Das rote Viereck stoppt aktuelle Vorgänge, z.B. wenn Minecraft bereits gestartet ist.
 - Die orangefarbene Datei mit dem Pfeil exportiert die Mod für die Verwendung mit anderen Mod-Launchern.
5. Hier kannst du Variablen erstellen.

Nützliche Einstellungen im Spiel

Wenn du in Minecraft längere Bauarbeiten durchführst, können einige Einstellungen hilfreich sein. Dazu gibt es Befehle in Minecraft. Um einen Befehl auszuführen, öffne das Spiel mit **T**, um den Chat zu öffnen, und gib / gefolgt von dem gewünschten Befehl ein.

- Zeit auf Tag setzen und auf Tag belassen:
- ```
/time set day & /gamerule doDaylightCycle false
```

- Wetter auf klaren Himmel setzen und Wetter so belassen:  
`/weather clear & /gamerule doWeatherCycle false`

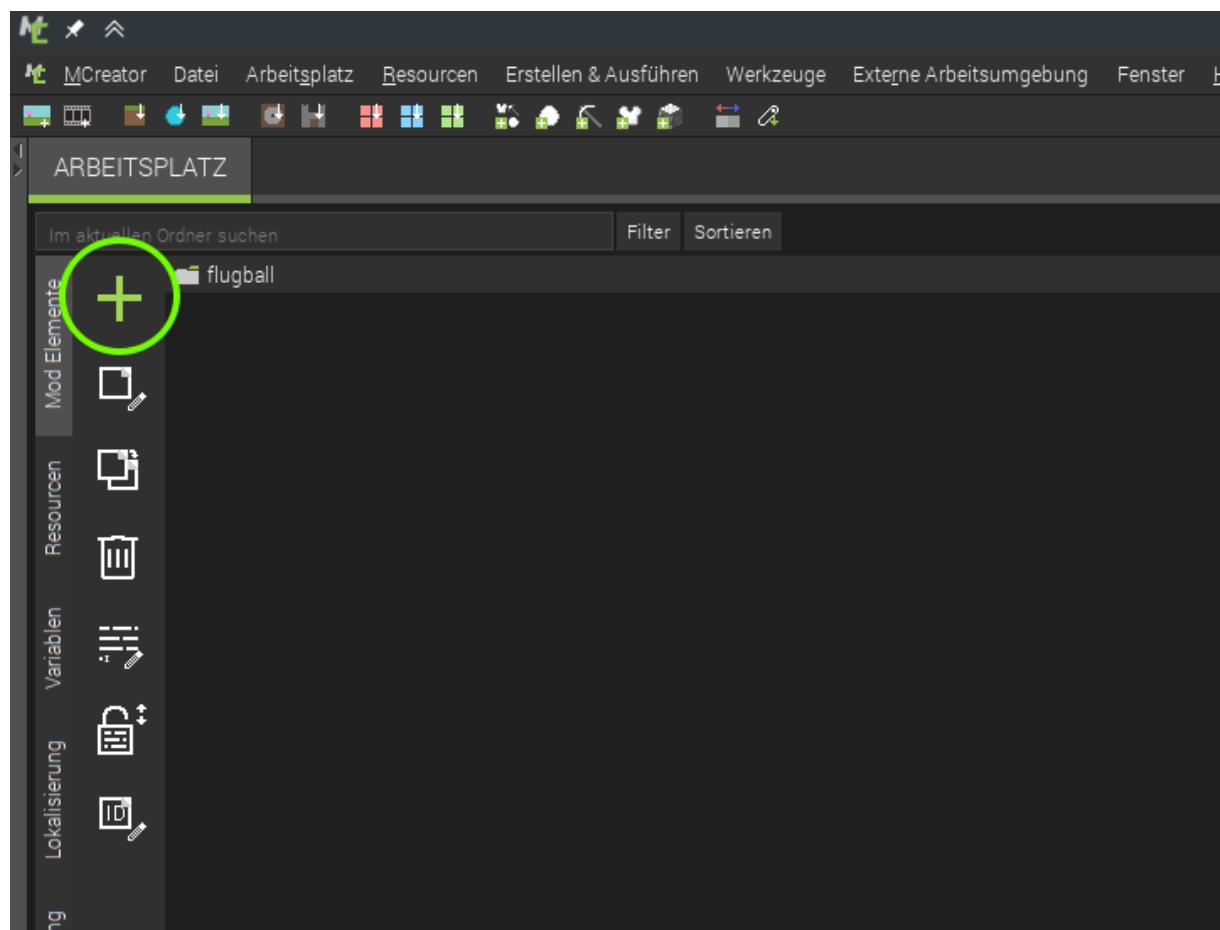


### Minecraft Wiki

Es gibt noch viele weitere nützliche [gamerules](#). Hier findest du eine Liste: [Minecraft Wiki Gamerules](#)

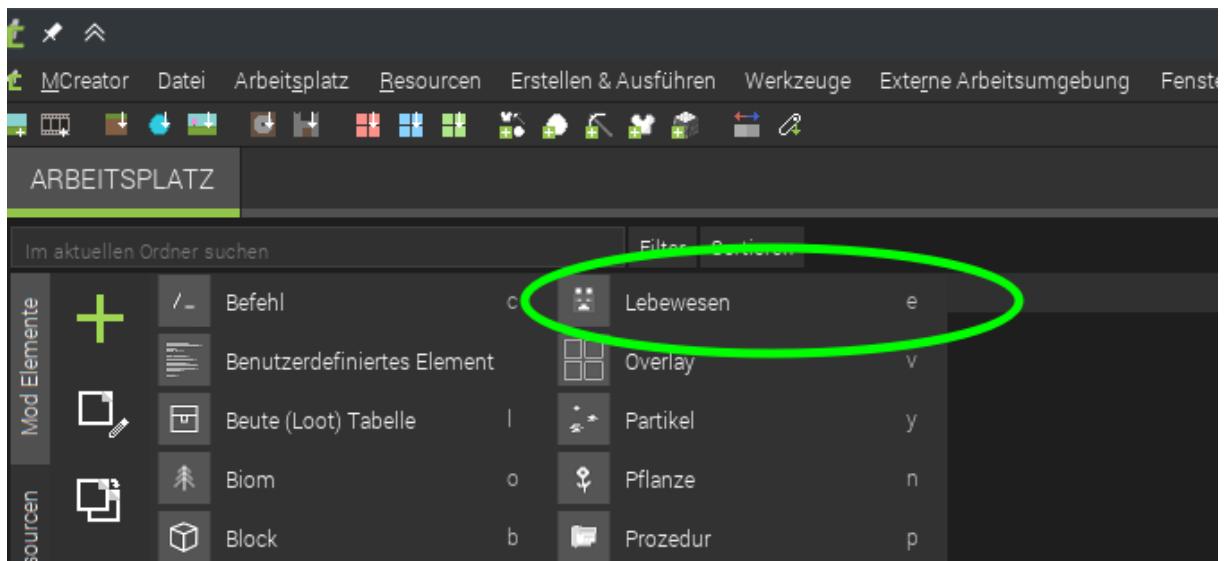
## Level 1: Der Ball für unser Spiel

Zuallererst benötigen wir einen passenden Ball. Er wird das Zentrum unseres Spiels sein. In Minecraft gibt es viele verschiedene “Dinge” - Blöcke, Pflanzen, Mobs. Wir haben uns für ein **Lebewesen** als Ball entschieden. Warum? Das erfährst du gleich...



**Abbildung 5:** Starte die IDE und erstelle ein neues Objekt

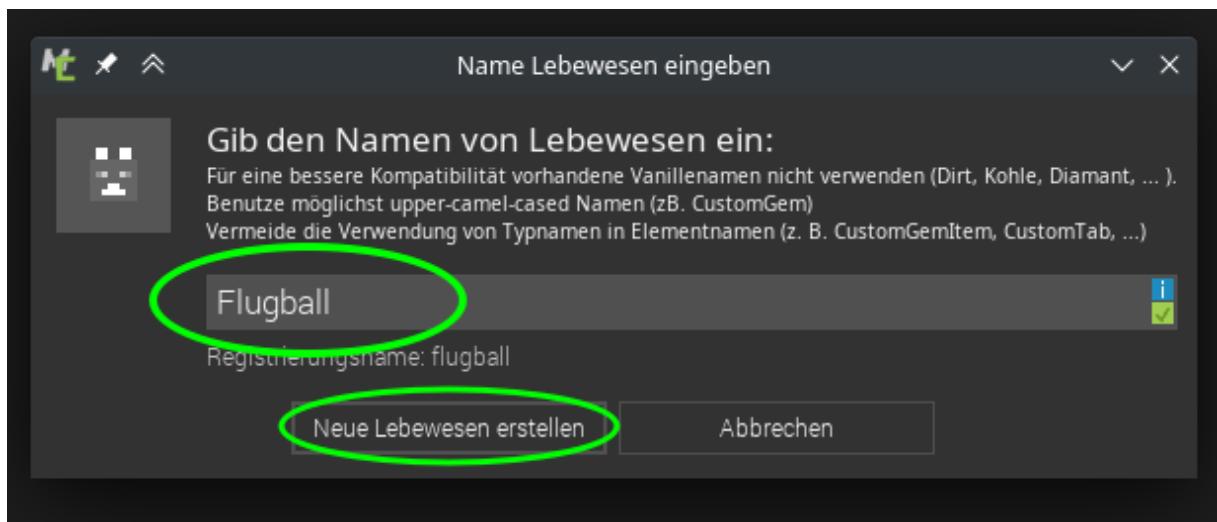
Du kannst neue Objekte über das “+”-Symbol hinzufügen. Wähle dann “Lebewesen” aus:



**Abbildung 6:** Wähle “Lebewesen”

Lebewesen sind “Entities”, die sich anhand von Regeln bewegen, die du festlegen kannst.

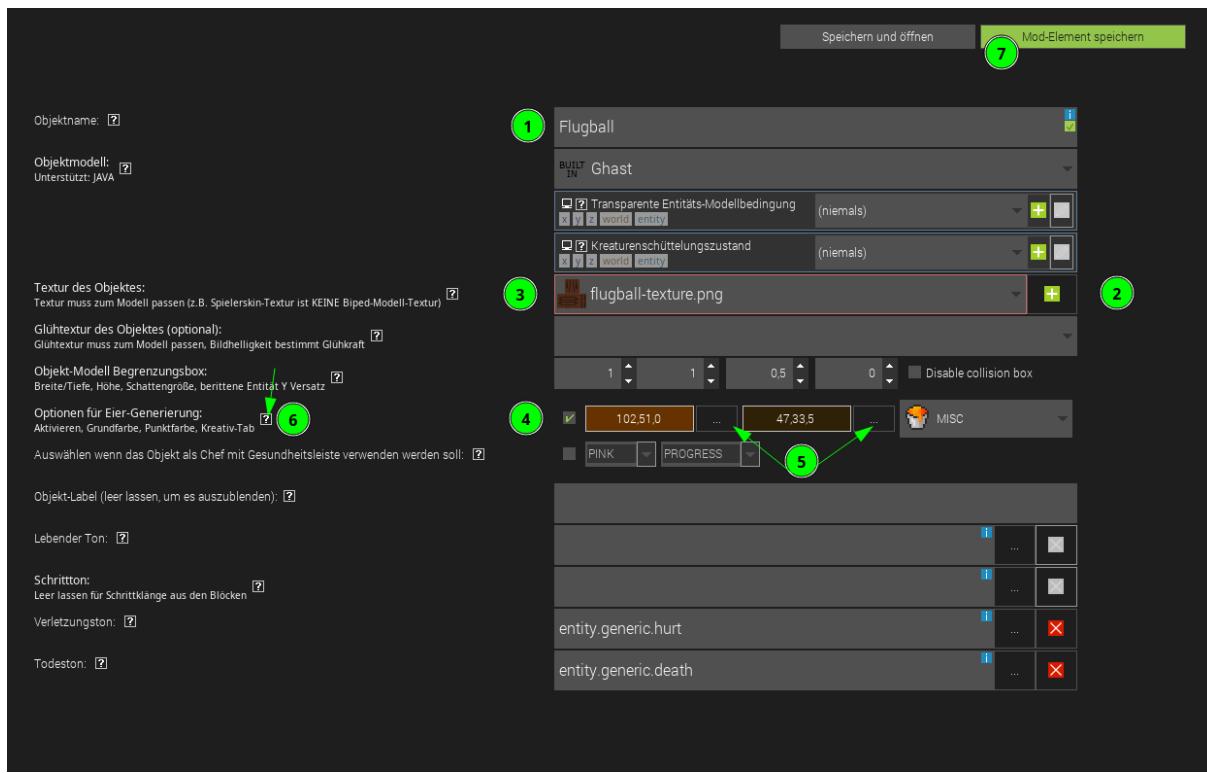
Natürlich musst du einen Namen vergeben:



**Abbildung 7:** Name des Lebewesens: “Flugball”

Jetzt öffnet sich das Fenster, in dem du alle Eigenschaften des neu erstellten Lebewesens festlegen kannst. In der Bearbeitung eines Objekts gibt es verschiedene Kategorien (wählbar in der unteren Leiste).

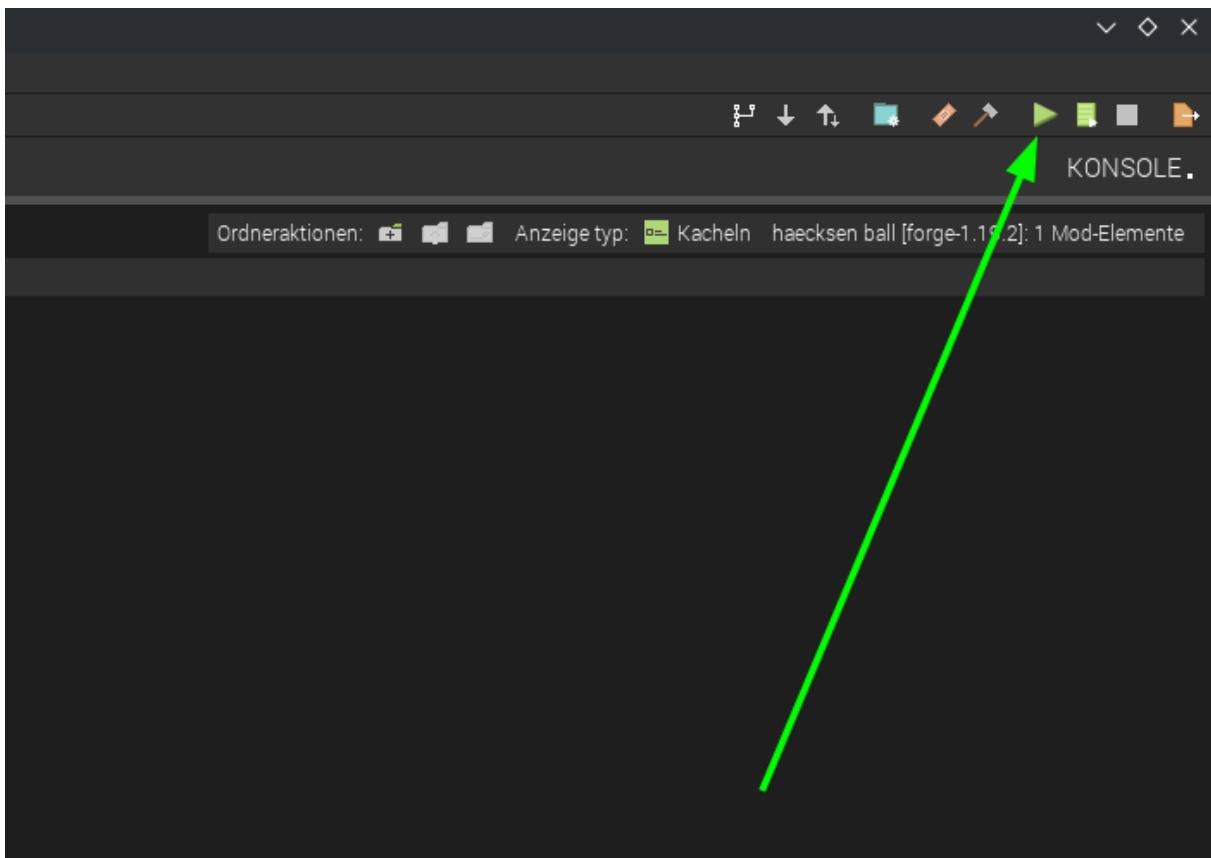
Bei *Bild und Ton* sind folgende Schritte wichtig für unseren Ball:



**Abbildung 8:** Lebewesen - Bild und Ton anpassen

- Der **Objektname** wird aus dem vorherigen Dialog übernommen.
- Die Textur bestimmt das Aussehen deines Balls. Du fügst die Textur mit dem +-Symbol hinzu. Du kannst selbst eine Textur erstellen oder diese als Beispiel verwenden: [Textur herunterladen](#)
- Danach kannst du die Textur im Dropdown auswählen.
- Wir lassen uns auch ein Spawning-Ei generieren.
- Für das Ei können wir bei den 3 Punkten neben dem Farbfeld die Farben auswählen, die das Ei haben soll.
- Zu jedem Attribut, das man hier auswählen kann, gibt es bei dem Fragezeichen einen Hilfetext.
- Vergiss nicht, mit **Mod-Element speichern** zu speichern!

Nach dem Speichern dauert es eine Weile, bis MCreator die Mod daraus erstellt hat. Wenn das Play-Dreieck oben rechts grün leuchtet, kannst du mit dem Play-Dreieck Minecraft direkt mit der neuen Mod starten:



**Abbildung 9:** Starte Minecraft mit deiner Mod: der Play-Knopf



#### Was passiert da im Hintergrund?

Da passiert im Hintergrund eine Menge - unter **Konsole** kannst du dir anschauen, was MCreator alles tun muss, um die Mod zu kompilieren.

Sobald Minecraft gestartet ist, erstelle eine Welt im **Kreativ-Modus**, um sofort auf alle Gegenstände zugreifen und den Ball am einfachsten testen zu können.

Das erstellte Spawn-Ei findest du im Kreativ-Inventar:



**Abbildung 10:** Inventar: Hier findest du dein erstelltes Lebewesen, bzw. dessen Ei

1. Wähle den Tab “**Verschiedenes**”.
2. Scrolle ganz nach unten.
3. Voilà! :)

**Geschafft: Der Ball fliegt!**

Voilà! So sieht es bisher aus:

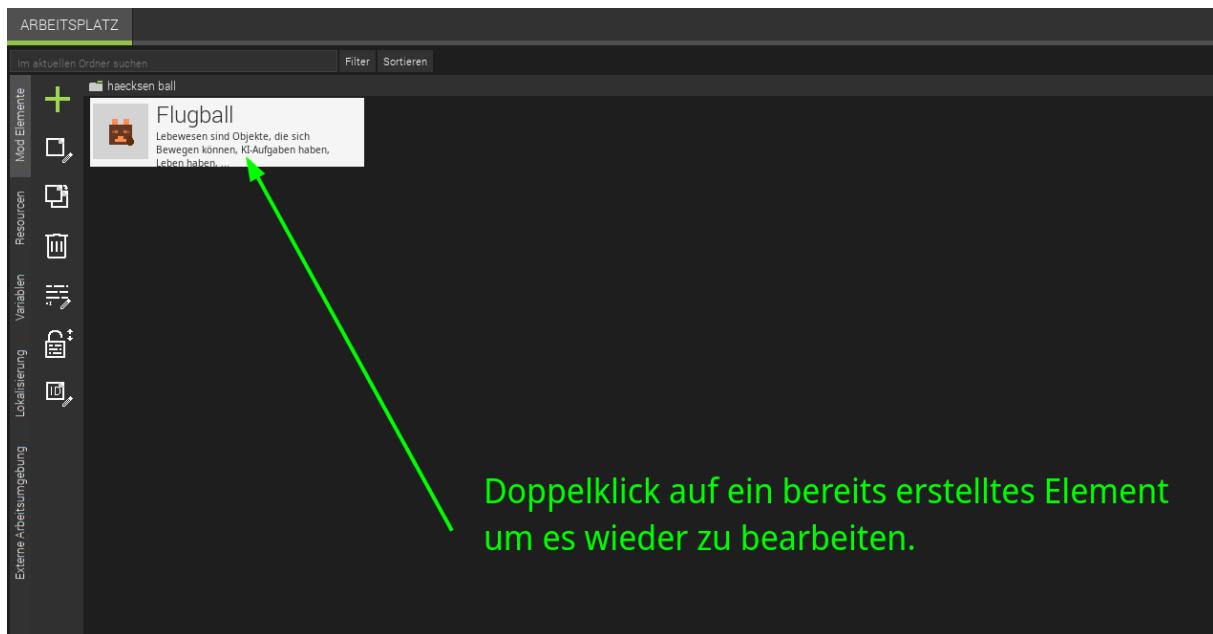


- Der Ball fällt nicht mehr, sondern fliegt.
- Der Ball bewegt sich nicht mehr von alleine.
- Der Ball stirbt nicht mehr, wenn man ihn schlägt. :)

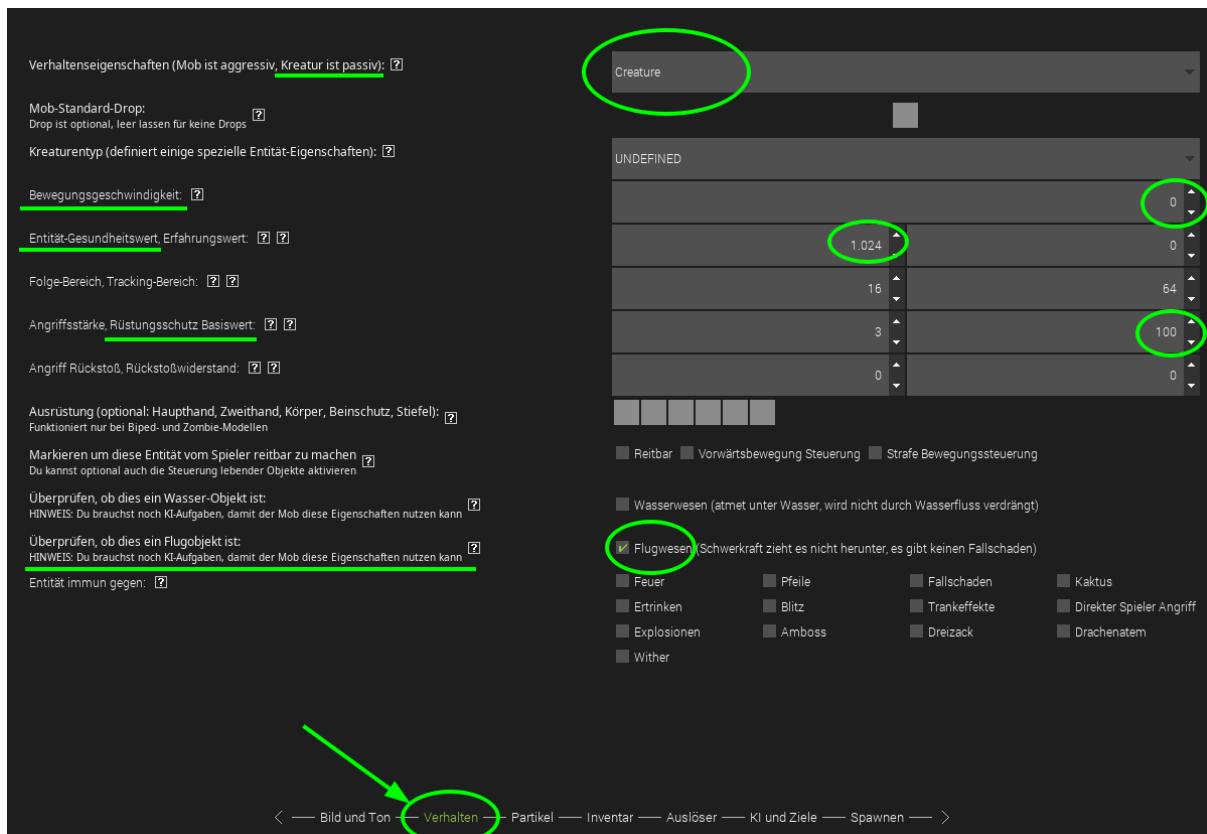
**Verhalten anpassen**

Nun wollen wir unserem Ball beibringen, sich wie ein Ball zu verhalten.

Schließe dazu Minecraft und kehre zu MCreator zurück. Du kannst ein Element durch Doppelklick erneut bearbeiten:



**Abbildung 11:** Bearbeiten der Eigenschaften: Verhalten ändern



**Abbildung 12:** Hier kannst du alles zum Verhalten anpassen

Um das Verhalten zu ändern, wechsle unten zum Verhaltens-Reiter. Jetzt kannst du die Werte anpassen:

- **creature:** Damit der Ball nicht angreift.
- **Bewegungsgeschwindigkeit:** 0, damit er nicht von alleine wegfliegt. Unser Ball soll nicht mehr sterben, wenn man ihn schlägt:
  - **Gesundheitswert:** 1024 (höchstmöglicher Wert, 2er-Potenz)
  - **Rüstungsschutz:** 100 (ebenfalls höchstmöglicher Wert)

**Ball fertig - yeah!**

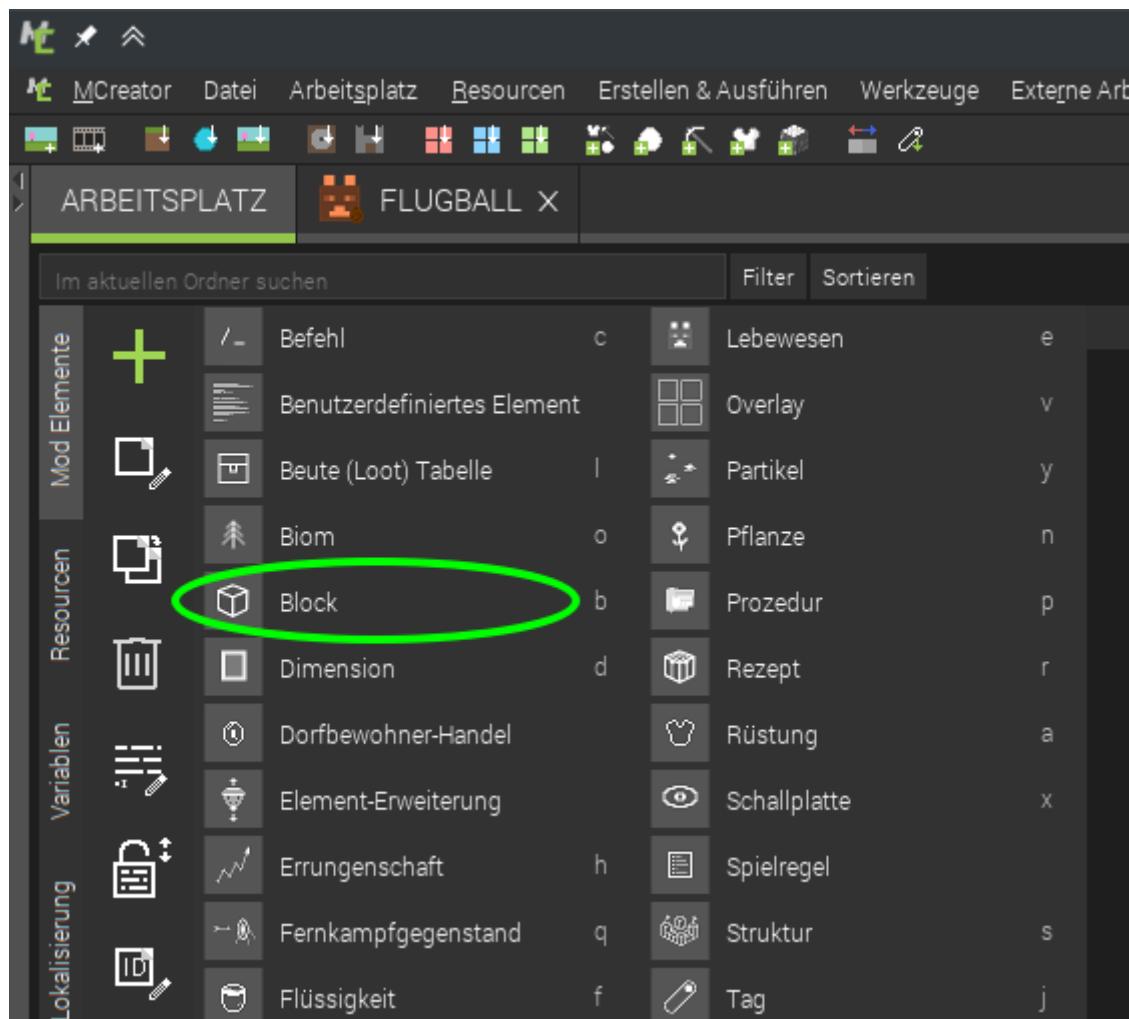
Das hast du bereits erreicht:

**Abbildung 13:** Stand Level 1 - Bälle sind geschafft!

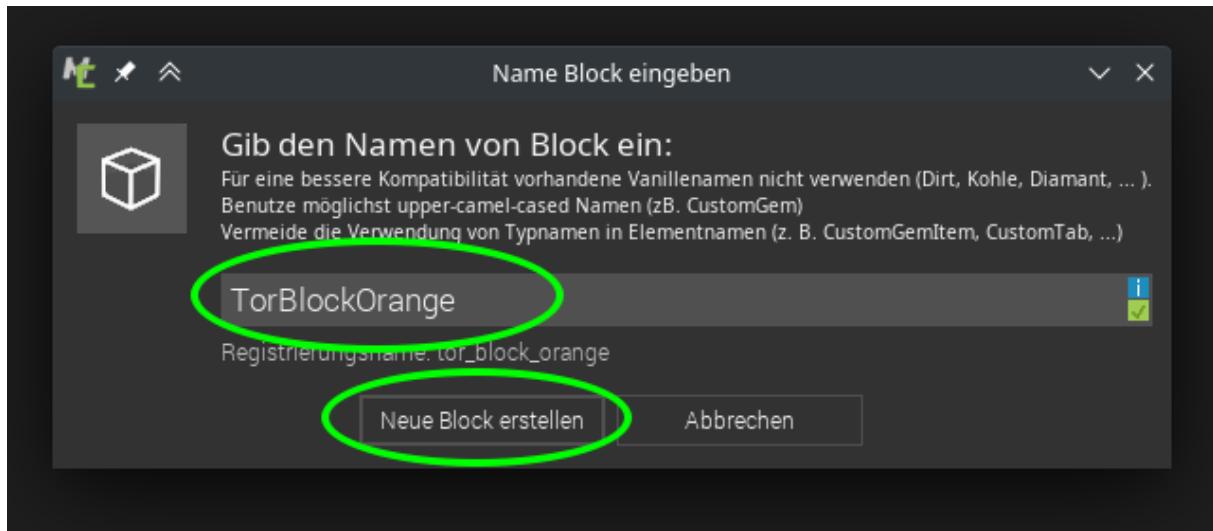
- Der Ball fällt nicht mehr, sondern fliegt.
- Der Ball bewegt sich nicht mehr von alleine.
- Der Ball stirbt nicht mehr.
- Es macht bereits Spaß, den Ball mit Schwert und Bogen zu schubsen (die Verzäuberungen "Schlag" oder "Rückstoß" helfen dabei ;D).

## Level 2: Tor! Oder: Da muss der Ball rein

Unser Tor soll aus Blöcken gebaut werden können: In diesem Level erstellst du also einen eigenen Block, mit dem dann das Tor im Spiel gebaut werden kann.

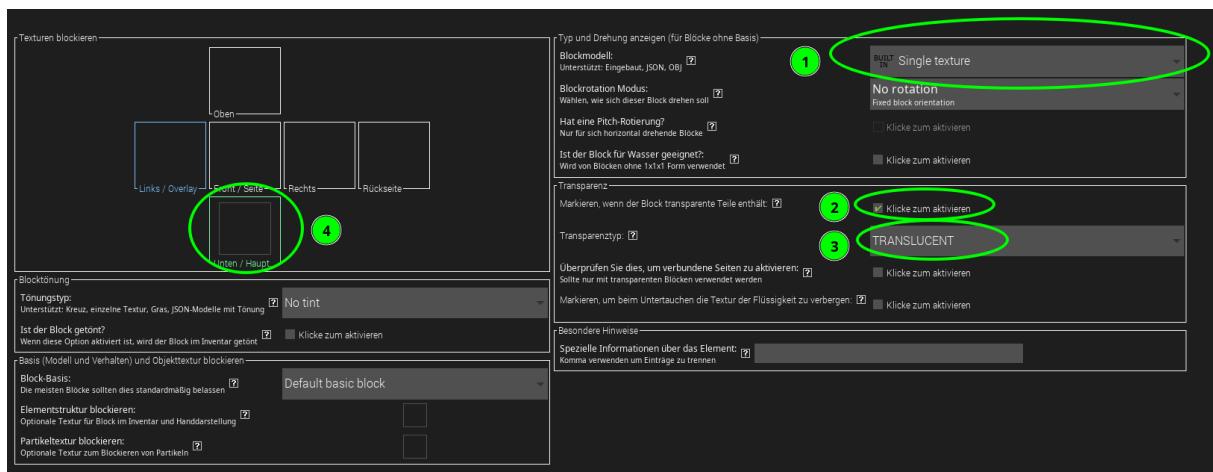


**Abbildung 14:** Erstelle einen neuen Block



**Abbildung 15:** Gib dem neuen Block einen (sinnvollen) Namen

Jetzt kannst du die Eigenschaften des Blocks anpassen - das kennst du schon aus unserem ersten Level, dort haben wir die Eigenschaften des "Lebewesens" Balls angepasst.

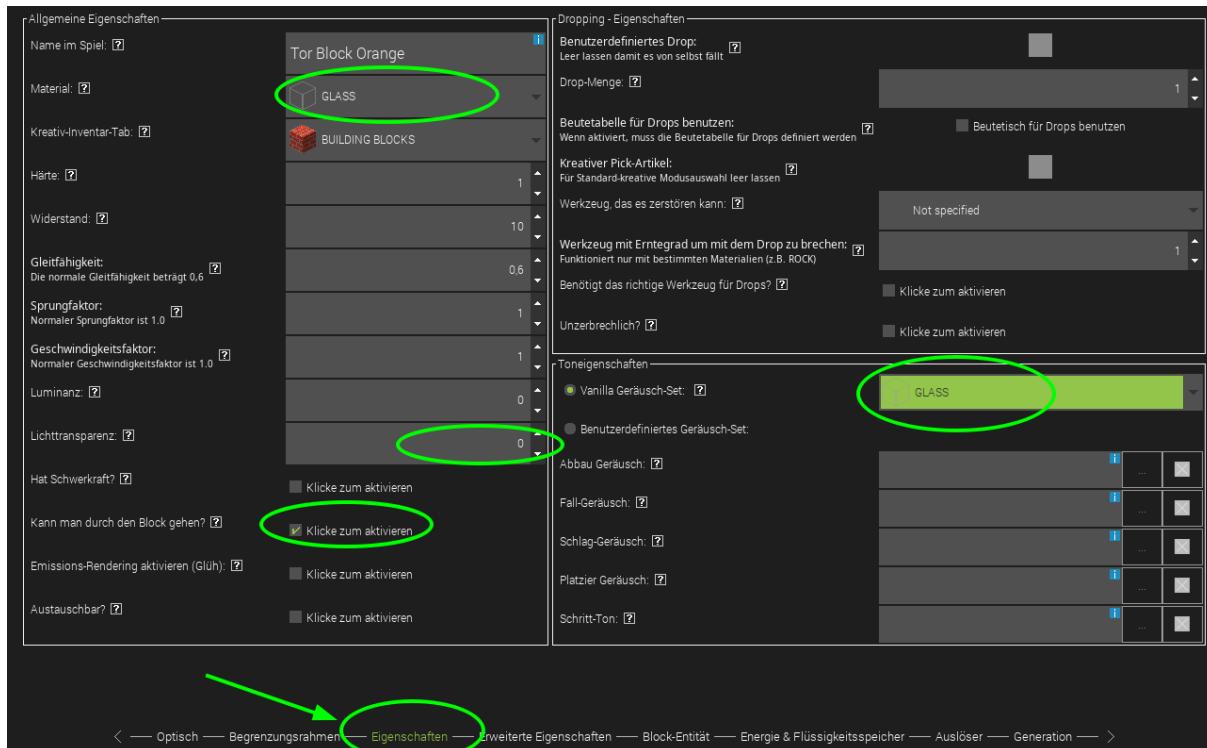


**Abbildung 16:** Die optischen Eigenschaften des Blocks

1. Wähle hier aus, dass auf allen Seiten die gleiche Textur verwendet werden soll, das spart viel Arbeit.
2. Die Beispieltextur (Schritt 4) ist halbtransparent, deshalb wähle hier aus, dass der Block transparente Teile enthält.
3. Setze den Transparenztyp auf **Translucent**.

4. Hier kannst du die Textur für den Block importieren und auswählen. Als Beispieltextur für das Tor kannst du diese Vorlage verwenden: [Gimp-Projekt "Tor\\_block\\_rahmen" Von GitHub herunterladen](#) Natürlich kannst du auch selbst kreativ werden und deine eigene Textur erstellen!

Gib dem Block jetzt glasähnliche Eigenschaften:



**Abbildung 17:** Eigenschaft: Glass-ähnlich

Wir haben verschiedene Möglichkeiten getestet und uns entschieden, den Block so zu konfigurieren, dass man durch ihn hindurchlaufen kann.



### Probier es aus!

Beim Programmieren ist es immer gut, zwischendurch immer mal wieder zu testen und auszuprobieren. Jetzt ist ein guter Zeitpunkt: Drücke den grünen Play-Knopf (rechts oben) und teste das Ganze!

## Anzeige des Spielstandes - Wie viele Tore wurden geschossen?

Wie könnten wir jetzt die erzielten Tore anzeigen? Wir haben uns an die Boss-Bar erinnert: In Minecraft Vanilla (ohne Mods) kann man mit bordeigenen Mitteln eine Bossbar hinzufügen.

Dafür gibt es diese Befehle:

1. Neue Bossbar erstellen

```
/bossbar add orange {"text":"Team Orange"}
```

2. Farbe festlegen

```
/bossbar set orange color yellow
```

3. Maximalwert der BossBar (12 bedeutet, wer 12 Tore hat, hat gewonnen)

```
/bossbar set orange max 12
```

4. Stil auswählen (schön, passend zum Maximalwert)

```
/bossbar set orange style notched_12
```

5. ... und jetzt noch anzeigen:

```
/bossbar set orange players @a
```



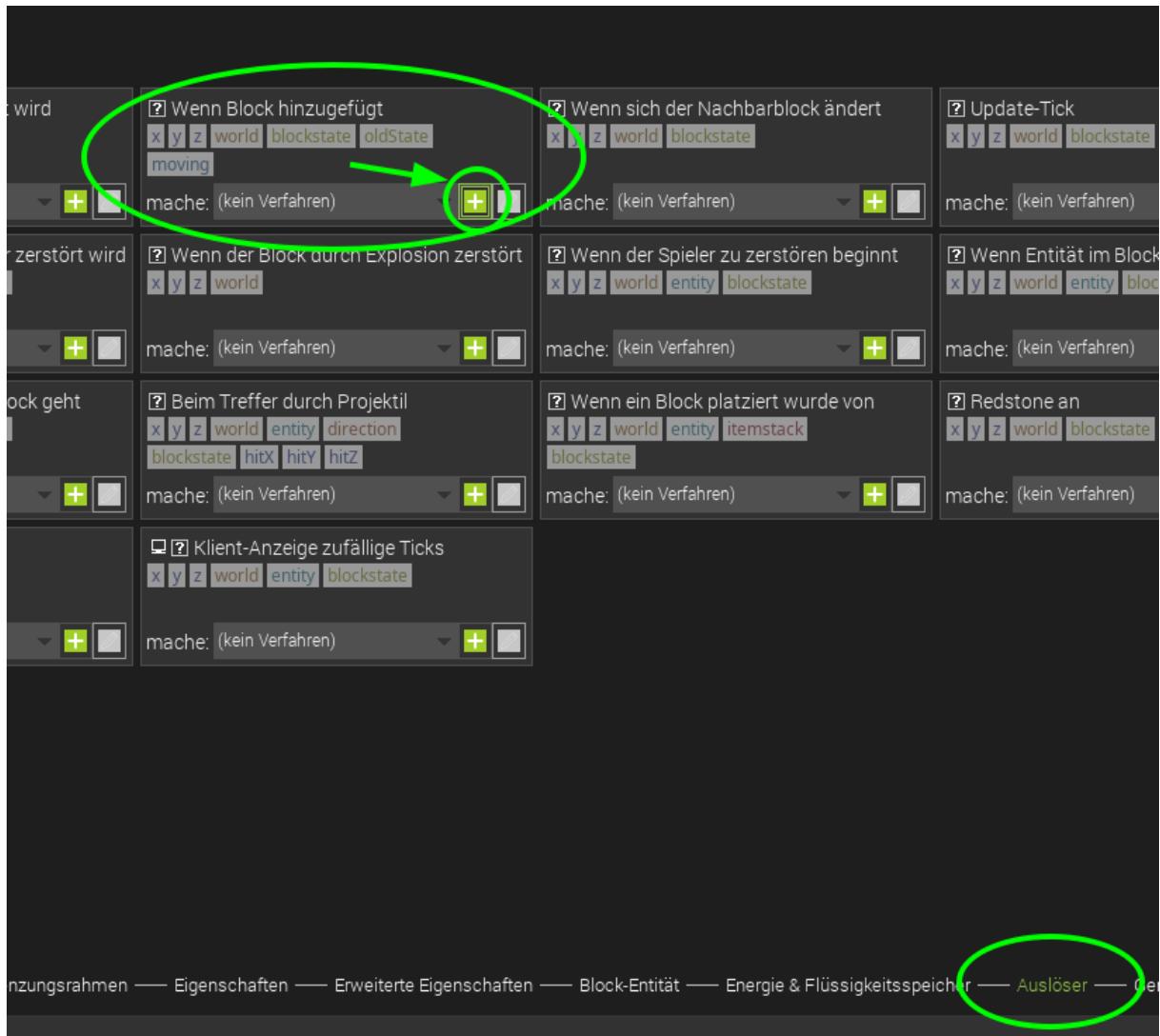
### Probiere es aus!

Öffne Minecraft (am einfachsten wieder über MCreator, grüner Knopf...) und starte eine Kreativ-Welt. Öffne mit **T** die Chat-Zeile. Gib die Befehle wie oben nacheinander in dieser Zeile ein und bestätige jeden mit Enter. Klappt es? Dann solltest du jetzt eine Boss-Bar sehen!

## Wie werden die Tore gezählt? Auslöser!

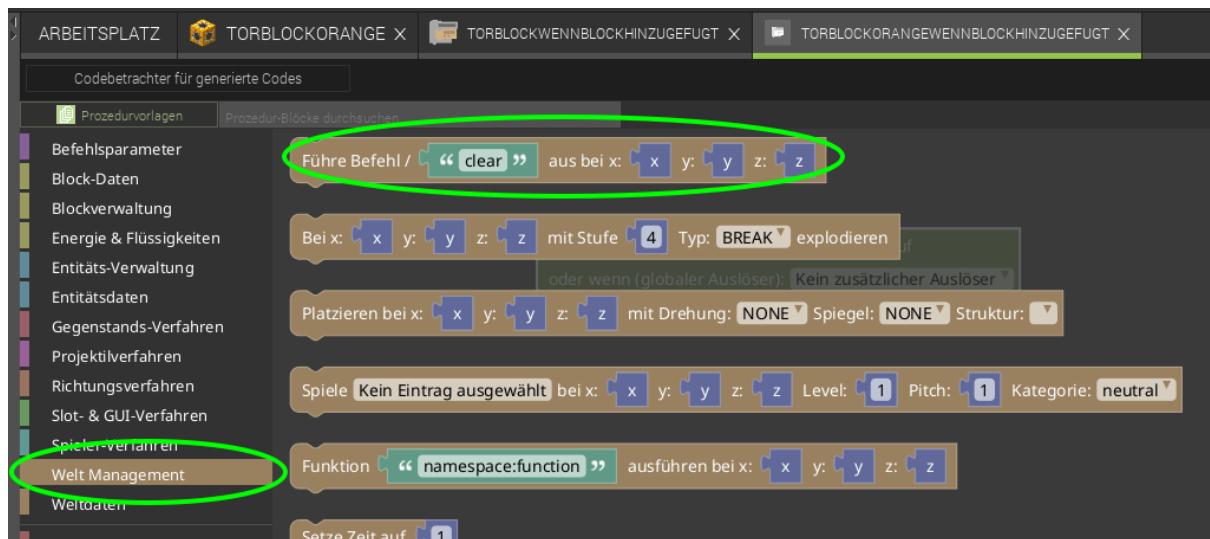
Jetzt muss das Tor ja wissen, dass ein Ball reinflog. Wie könnten wir das am besten anstellen? **Auslöser** bieten die Möglichkeit, auf gewisse Ereignisse in Minecraft zu reagieren und unseren Code auszuführen, z.B. wenn ein Block gesetzt wird. Gehe noch mal in die Eigenschaften des Tor-Blocks und wähle unten "Auslöser" aus. Jetzt schreibst du deinen ersten Code für die Mod: Wähle im Bereich **Wenn Block hinzugefügt** aus, dass du eine neue Prozedur erstellen möchtest - wie gewohnt mit dem **+** **Plus-** Knopf.

Wir erstellen also eine Prozedur, ein Stück Code, das ausgeführt werden soll, wenn der Block eine Entität (Entity) berührt:



**Abbildung 18:** Auslöser hinzufügen - erzeugt ein Ereignis

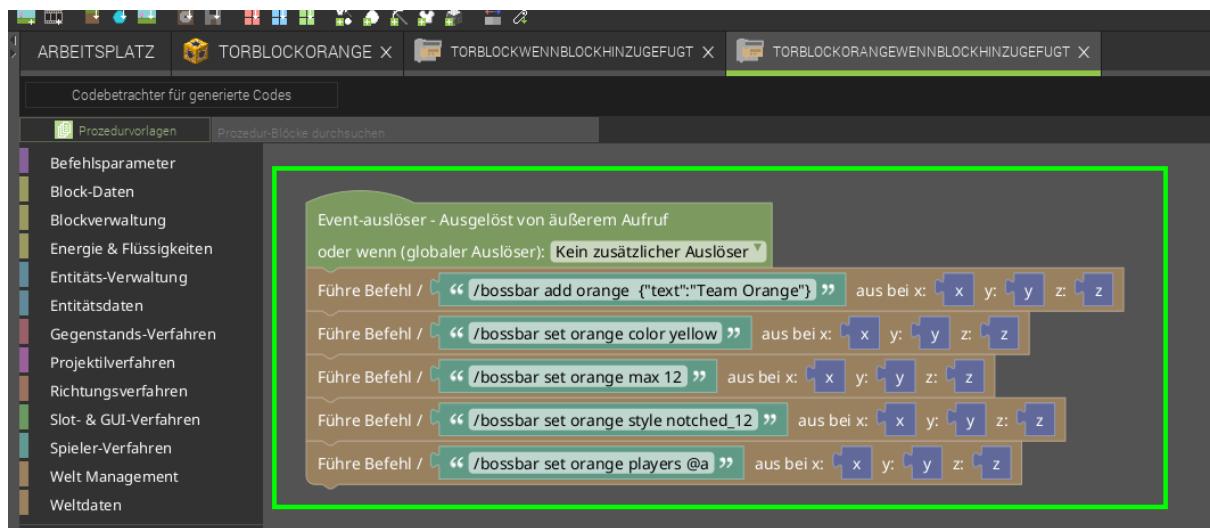
Der Name der Prozedur wird automatisch generiert (den kannst du so lassen). Jetzt sollten die 4 Minecraft-Befehle von oben ausgeführt werden, wenn das Tor gebaut wird. Also suchen wir den Baustein, der uns ermöglicht, einen Minecraft-Befehl auszuführen.



**Abbildung 19:** Wie können Minecraft-Befehle ausgeführt werden?

- Wähle links die Kategorie *Welt Management*.
- Nimm den Programmier-Block ganz oben: **Führe Befehl ... aus ...**
- Ziehe ihn in die Programmier-Oberfläche und docke ihn unter dem Event-Auslöser an.
- Füge jeweils in den Anführungszeichen die Befehle von oben ein.

So sieht es aus, wenn die Befehle von oben alle ausgeführt werden sollen:



**Abbildung 20:** Minecraft Befehle in MCreator für Bossbar

Vergiss nicht zu speichern, indem du rechts oben auf **Mod-Element speichern** klickst.

Gerne mal testen. Falls du es nochmal testen möchtest, kannst du die Bossbar mit `/bossbar remove orange` wieder entfernen vor dem neuen Test.



### Mal nachschauen

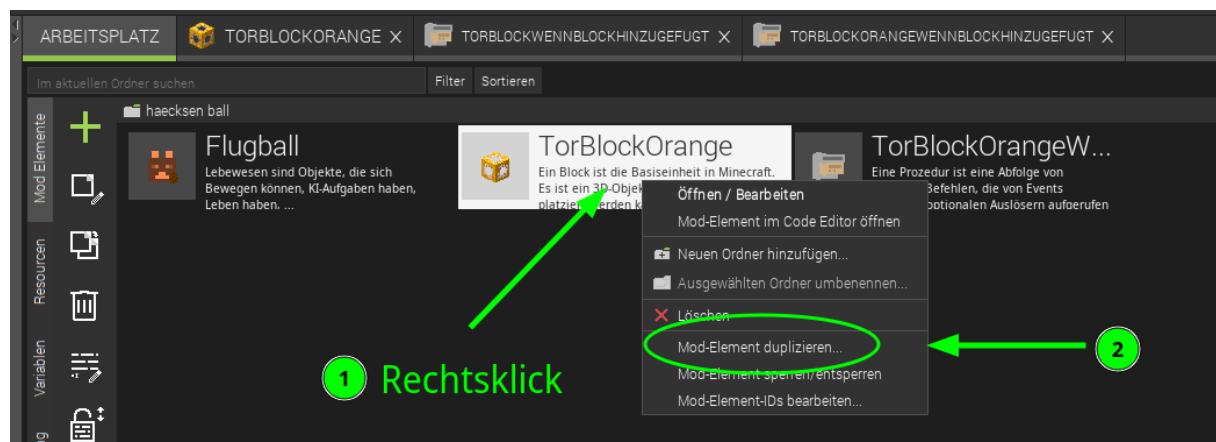
Jetzt sollte es so aussehen, nachdem du einen orangenen Torblock gesetzt hast:



**Abbildung 21:** Bossbar für Team Orange

### Was ist mit der gegnerischen Mannschaft?

Jetzt natürlich das Ganze noch für das zweite Team: Team Blau. Weil wir nicht alles nochmal neu machen wollen, können wir die bestehenden Elemente kopieren:



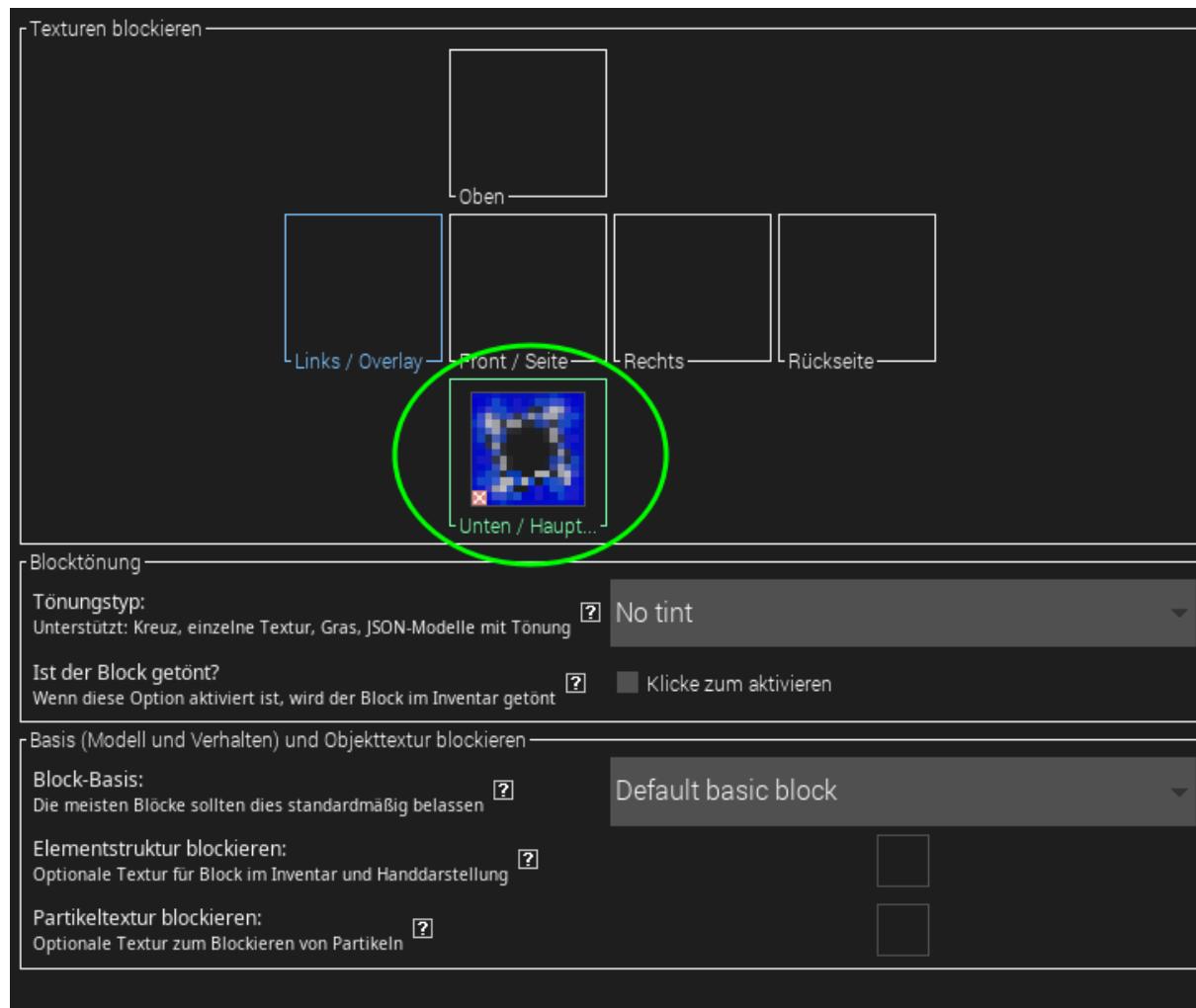
**Abbildung 22:** Arbeit sparen: so kannst du bestehende Elemente duplizieren

Du musst jetzt einen neuen Namen vergeben, z.B. `TorBlockBlue`.

Erstelle jetzt - wie vorher beim ersten Block - einen neuen Auslöser und eine neue Prozedur:

- Rechtsklick
- *Mod-Element duplizieren*
- Neuer Name: *TorBlockBlueWennBlockHinzugefügt*

Und in beiden Elementen musst du die IDs und farbrelevanten Eigenschaften anpassen:



**Abbildung 23:** Tausche die Textur des Blocks aus

- Tausche die Blocktextur aus.
- In der Kategorie *Eigenschaften* von dem Block solltest du noch den Namen von *Tor Block Orange* auf *Tor Block Blau* ändern.
- Und natürlich den Auslöser auch anpassen, damit die Funktion für die blaue Bossbar aufgerufen wird:



**Abbildung 24:** Tausche die Farbe aus

- Tausche den Code für die Farbe aus: *orange* war hier die ID der Bossbar und *yellow* war die Farbe. Beides kann zu *blue* geändert werden.



### Geschafft!

Wow - du hast schon eine Menge geschafft: Bälle gibt's, Tore - und sogar die Tor-Anzeige!



## Level 3: Tore / Punkte zählen

So, wir haben Tore, einen Ball - aber aktuell zählen wir die Tore noch nicht. Das packen wir in diesem Level an!



### Variablen - eine kurze Einführung

Was genau ist eine Variable? Eine Variable hört sich an wie „variabel“, also veränderbar. Am besten kannst du dir eine Variable wie einen dieser Klebezettel vorstellen. Die sind super, um sich Sachen zu merken!

### Hintergrund: Variablen-Typen

Es gibt verschiedene Arten von Variablen, je nachdem, was du ausdrücken möchtest:

- Ja / Nein – Boolean
- Text – String
- Einfache Zahlen (1, 2, 3...) – Integer
- Kommazahlen (3,1415) – Float

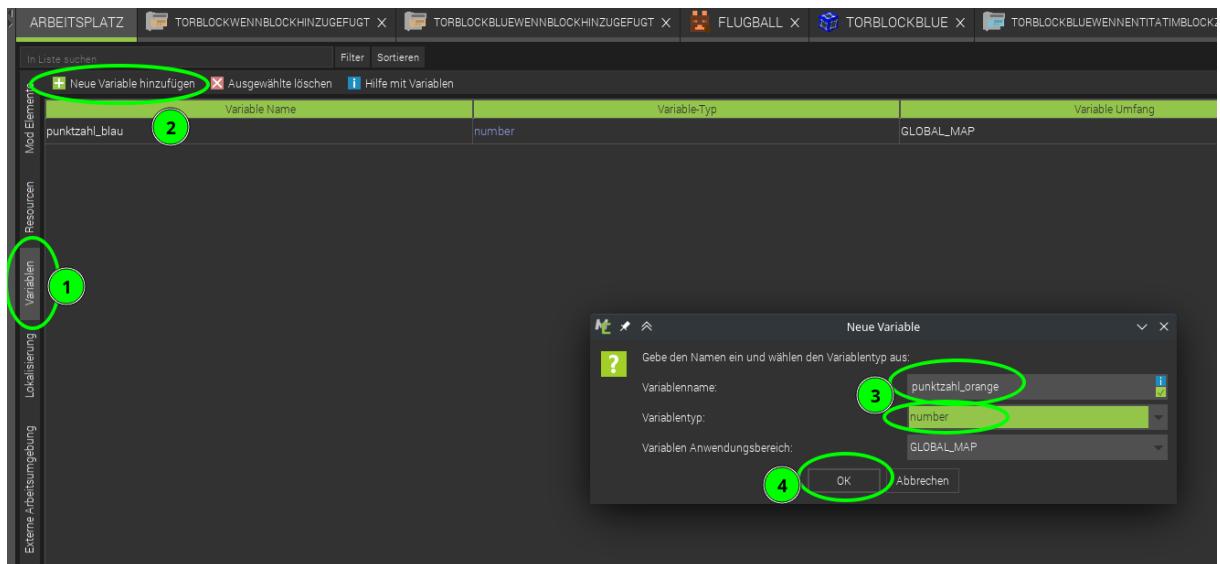
Und noch viele andere...

### Achtung: Gute Namen sind gute Helfer

Variablennamen können aus einem einzelnen Buchstaben oder aus ganzen Wörtern bestehen. Wähle also weise: Gib den Variablen Namen, die andere (und auch du nach 2 Wochen) noch verstehen. Dann findest du dich viel leichter im eigenen Programm zurecht.

## Punkte merken

Wir erstellen also eine Variable - In dieser merken wir uns jeweils die Tore / Punkte pro Team.



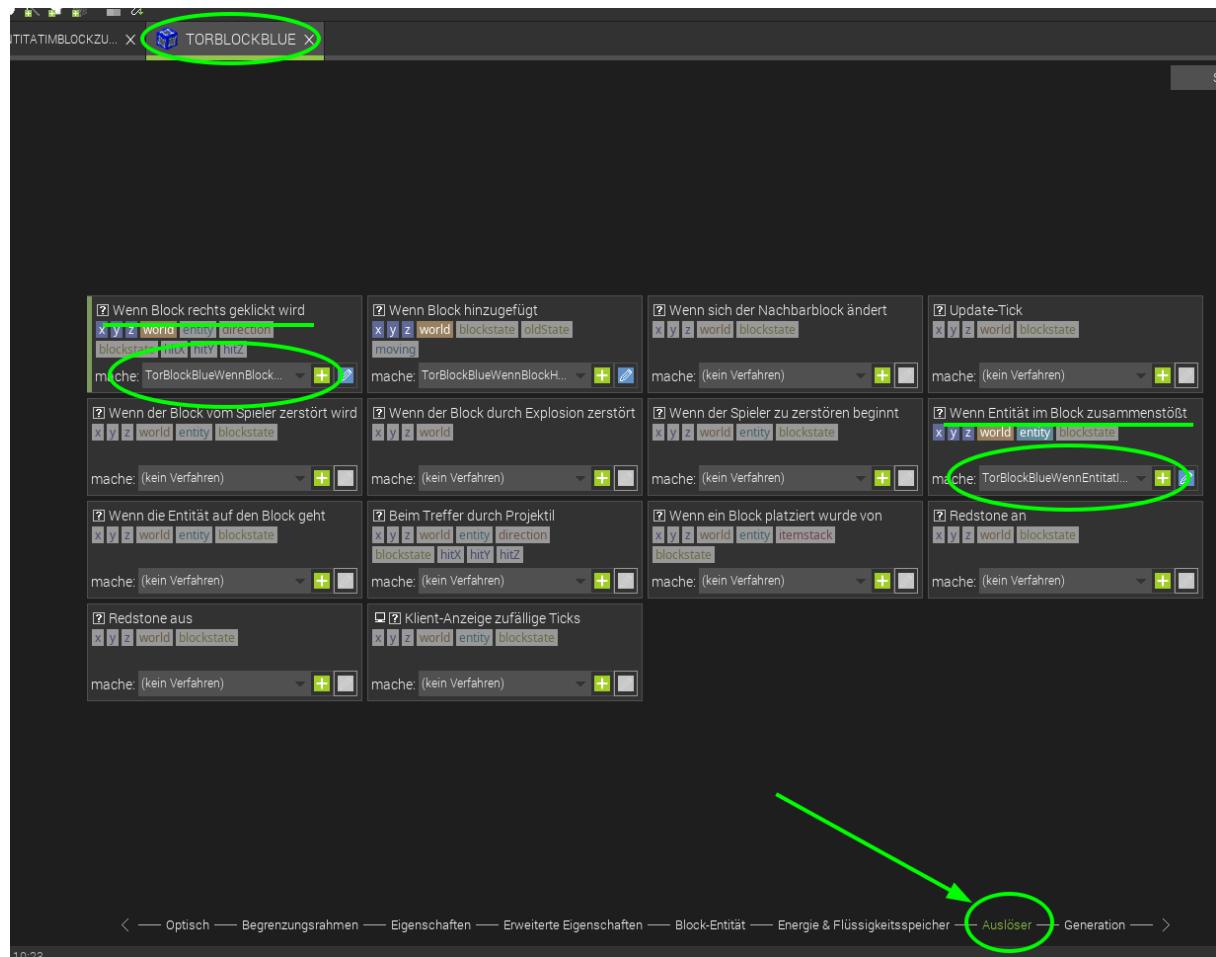
**Abbildung 25:** So erstellst du eine neue Variable

Dazu erstellen wir eine Variable für jedes Team:

1. Wähle in der seitlichen Leiste *Variablen* aus.
2. Füge eine *Neue Variable* hinzu.
3. Vergib einen *Variablennamen* und setze den *VariablenTyp* auf *number*.
4. Speichere mit *Ok*.

## Tore erkennen

Jetzt müssen wir nur noch erkennen, wenn ein Tor geschossen wurde. Genau - da gab's ja was - die "Auslöser". Wir suchen also einen Auslöser, der erkennt, wenn ein Tor geschossen wurde.

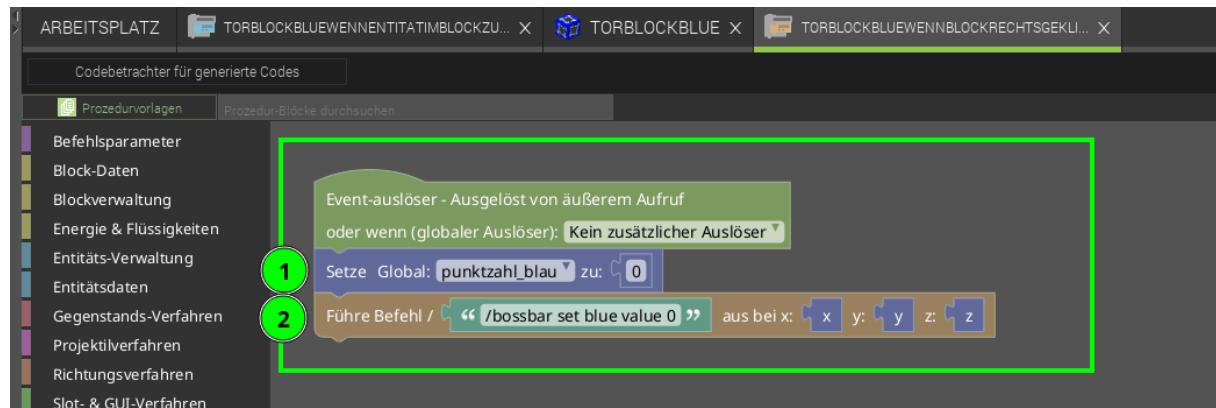


**Abbildung 26:** Auslöser: So erkennt das Tor, was gerade passiert

Wir fügen 2 Auslöser hinzu:

- Einen *Wenn Block rechts geklickt wird*, um Punkte zurückzusetzen.
- Einen *Wenn Entität im Block zusammenstößt*, um Punkte hochzuzählen und den Ball zu entfernen.

## Prozedur 1: Punkte zurücksetzen



**Abbildung 27:** Code, um die Tore zurückzusetzen

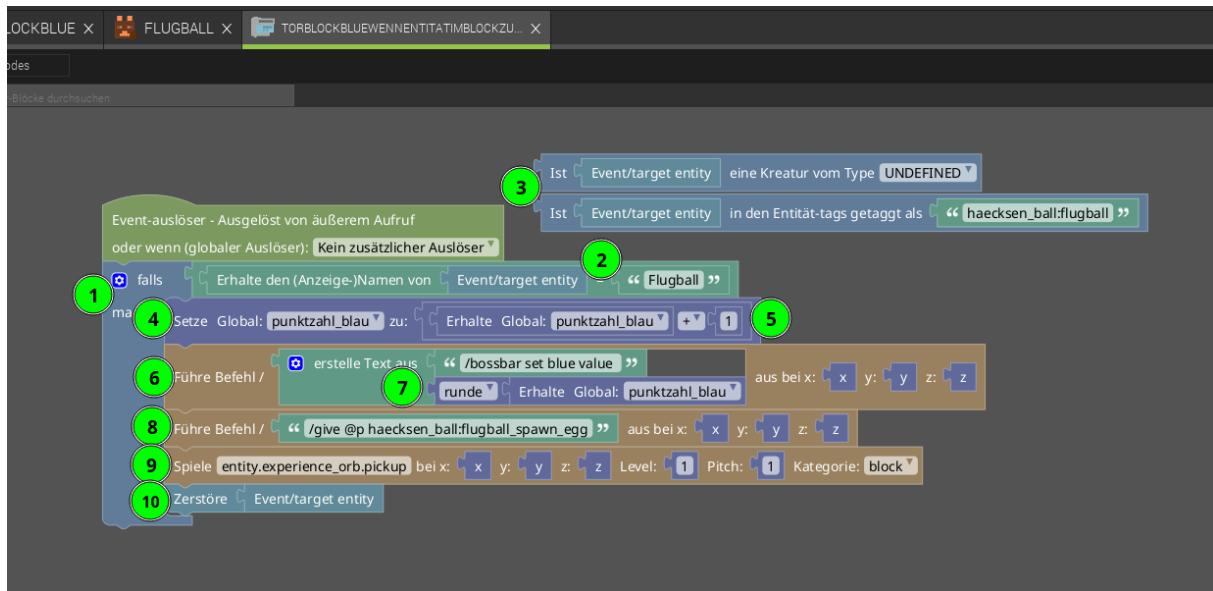
In diesem Fall möchten wir, dass die Punkte des Teams auf “0” gesetzt werden und dass die Bossbar auch zurückgesetzt wird. Dafür verwenden wir folgende Befehle:

1. Setze die globale Variable zum Punkte Zählen auf 0 zurück.
2. Führe `/bossbar set blue value 0` in Minecraft aus, um die Bossbar wieder auf 0 zu setzen.

## Prozedur 2: Tore zählen

Um die Tore zu zählen, haben wir einen Auslöser erstellt, der immer startet, wenn er vom Ball berührt wird.

Die Prozedur für den Zusammenstoß mit einer Entity ist länger und sieht so aus:



**Abbildung 28:** Code, um die Tore zu zählen

1. Tore sollen nur bei Ball-Kontakt gezählt werden: Als erstes prüft das Programm, ob es sich bei der Entität um einen Ball handelt.
2. Dazu "holt" es sich den Anzeigenamen der "Berührenden Entity" mit `Erhalte den Anzeigenamen von Event/target entity` und prüft, ob er mit dem String `Flugball` übereinstimmt (muss übereinstimmen mit dem Objektnamen bei [Bild und Ton von Ball erstellen](#)).
3. Diese beiden Zeilen, die vom Hauptcodeblock abgelöst sind, werden nicht ausgeführt. Sie dienen nur als Hinweis darauf, was wir ausprobiert haben: Bei der (3) wollten wir auf diese beiden Arten testen, ob es sich um einen Ball handelt. Das hat leider nicht funktioniert.
4. Erhöhe die Punktzahl von Blau:
  - Dazu holt es sich die aktuelle `punktzahl_blau`.
  - Addiert die Zahl `1` hinzu.
  - Speichert das Ergebnis mit `Setze Global: punktzahl_blau` erneut in der Punktzahl für Blau ab.
5. Wir führen den Minecraft Befehl für die Bossbar aus - den kennst du bereits aus Level 3.
6. Dazu kombinieren wir den Text `/bossbar set blue value` (**Achtung! Leerzeichen am Ende sind wichtig, damit es funktioniert**) mit der `punktzahl_blau` und runden diese, da Minecraft keine Dezimalzahlen erkennt.
7. Wir geben dem Spieler, der am nächsten steht (@p), ein neues Spawn-Ei:  
`/give @p alpaka_ball:flugball_spawn_egg`
8. Wenn ein Tor fällt, muss man das natürlich auch hören: Mit diesem Befehl können wir einen Ton

abspielen lassen, wenn ein Tor geschossen wurde. Hier z.B. der gleiche Ton, den man hört, wenn man XP aufsammelt: `entity.experience_orb.pickup`.

9. Am Schluss wollen wir den Ball "los werden" - er soll verschwinden und mit einem neuen Ball geht es weiter.

- Dazu verwenden wir am besten den Befehl [Zerstöre](#).
- Achte darauf, dass die Minecraft-Befehle `kill @e[...]`, `tp @e[...] ~ -500 ~` und MCreator [Lösche](#) nicht funktionieren.

#### 10. **Vergiss nicht zu speichern ;)**



##### **Blau bekommt einen Punkt?**

Vielleicht wunderst du dich, warum wir dem blauen Team einen Punkt geben, wenn ein Ball ins blaue Tor geht - müsste es nicht andersherum sein? Aber sobald mehr als zwei Teams mitspielen, würde es nicht mehr funktionieren. Mit 12 Punkten hat man also verloren.



##### **Alternative: Punkte abziehen**

Wenn du möchtest, kannst du am Anfang auch 12 Punkte vergeben und mit jedem Tor im blauen Torblock 1 Punkt von Blau abziehen. Experimentiere gerne mit den Zahlen rum ;)



##### **Geschafft! Tore zählen - ich habe fertig!**

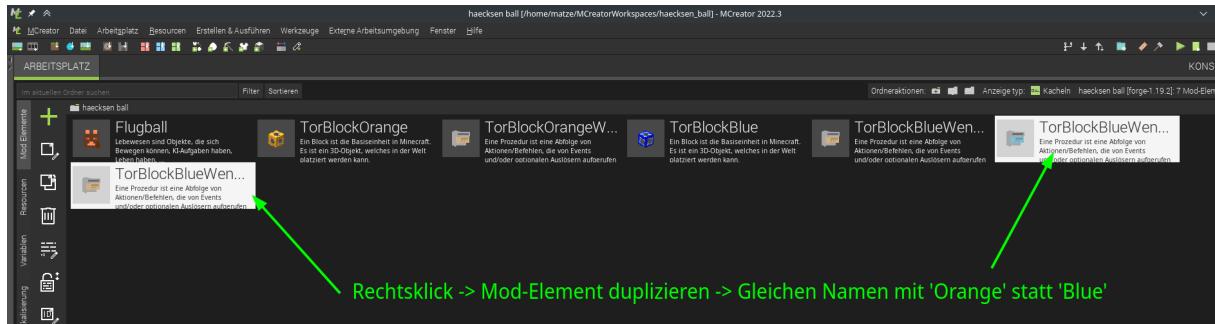
Jetzt kannst du im Spiel testen. Wenn ein Ball in einen blauen Torblock fliegt, sollte ein Ton erklingen, die Punktzahl in der Bossbar erhöht werden, der Ball verschwinden und du erhältst ein neues Spawn-Ei.

#### **Jetzt nochmal für Orange**

Die Variable `punktezahl_orange` für Orange haben wir bereits erstellt. Jetzt fehlen nur noch die passenden Prozeduren, die wir vom Team Blau kopieren können.

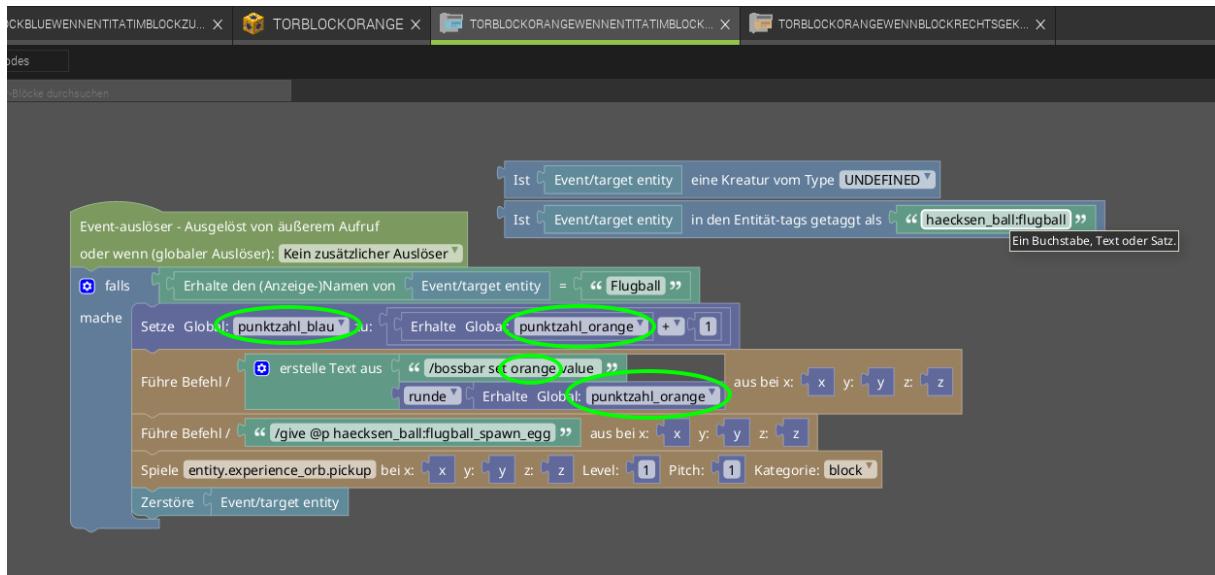
- Gehe zum **Arbeitsplatz**.
- Rechtsklicke auf die Prozedur und wähle **Mod-Element duplizieren**.

- Benenne die neuen Prozeduren um, z.B. ...Orange... statt ...Blau...

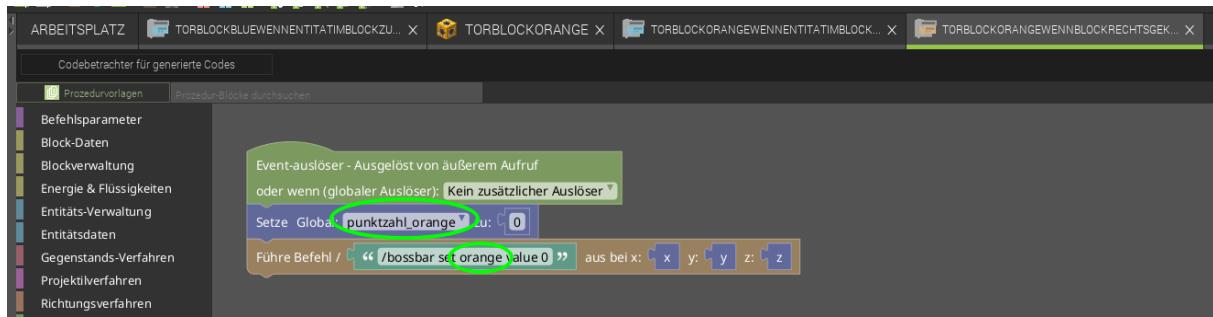


**Abbildung 29:** Faulheit siegt, Arbeit sparen: Dupliziert den bestehenden Code

Jetzt musst du natürlich alle Stellen, an denen "Blau" steht, durch "Orange" ersetzen, sowohl in den beiden Codedateien als auch im Auslöser:



**Abbildung 30:** Neuer Code - Zusammenstoß für Orange...



**Abbildung 31:** ... und der Rechtsklick-Code für Orange.

Vergiss nicht, auch den Auslöser anzupassen!



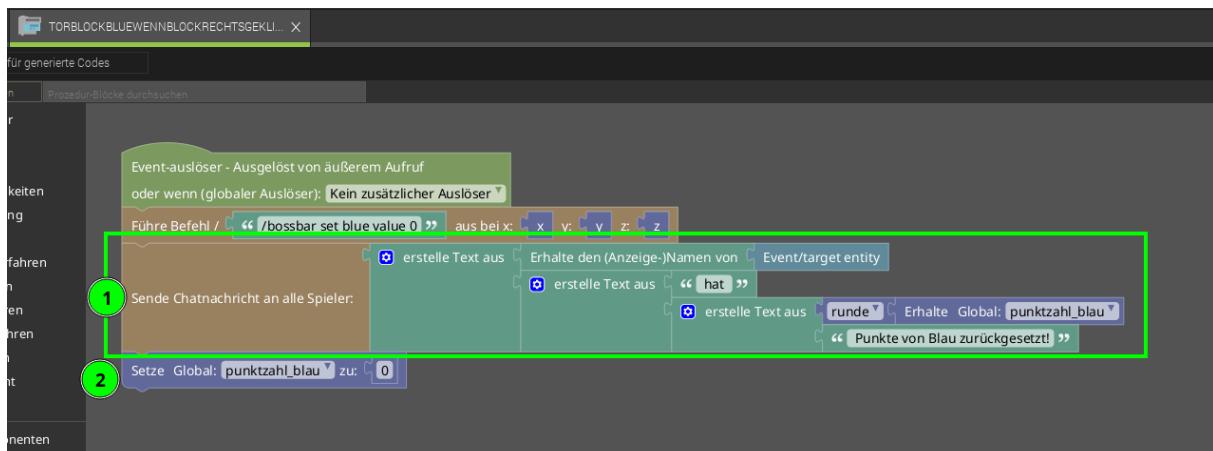
### Funktioniert es?

Teste alles noch einmal ;)

Wenn etwas nicht funktioniert, gehe die Schritte oben Schritt für Schritt durch, um sicherzustellen, dass nichts vergessen wurde.

## Level 4: Erweiterte Punkte-Anzeige / Punkte zurücksetzen

Beim Testen und Spielen ist uns etwas aufgefallen: Jeder kann relativ einfach dem anderen Team Punkte klauen, indem er einfach zum Tor schleicht und rechts klickt. Deswegen haben wir uns Folgendes überlegt: Wenn jedes Mal eine Nachricht an alle Spieler gesendet wird, wenn Punkte zurückgesetzt werden, wäre der Betrug schnell enttarnt. Schau dir mal den folgenden Programm-Code an:



**Abbildung 32:** Erweiterung des Codes zum Zurücksetzen

1. Wir setzen aus vielen Bauteilen die Nachricht für den Chat zusammen:
  - Spielername mit *Erhalte den (Anzeige-)Namen von > Event/target entity*
  - ” hat ” (Leerzeichen nicht vergessen)
  - *runde ]> Erhalte Global punktzahl\_blau*
  - ” Punkte von Blau zurückgesetzt!”
2. Achte darauf, dass die Punkte erst danach auf 0 gesetzt werden, ansonsten wird im Chat immer 0 angezeigt.

Das Gleiche musst du auch in dem Code für Orange durchführen, damit es für beide Farben passt.



#### Aufgepasst! Platz lassen

Wenn wir mehrere Textbausteine aneinander hängen, müssen wir darauf achten, wo die Leerzeichen stehen, sonst wird der Text schwer lesbar. Zum Beispiel vor und nach *hat* ist jeweils ein Leerzeichen.



#### Testen schadet nie ;)

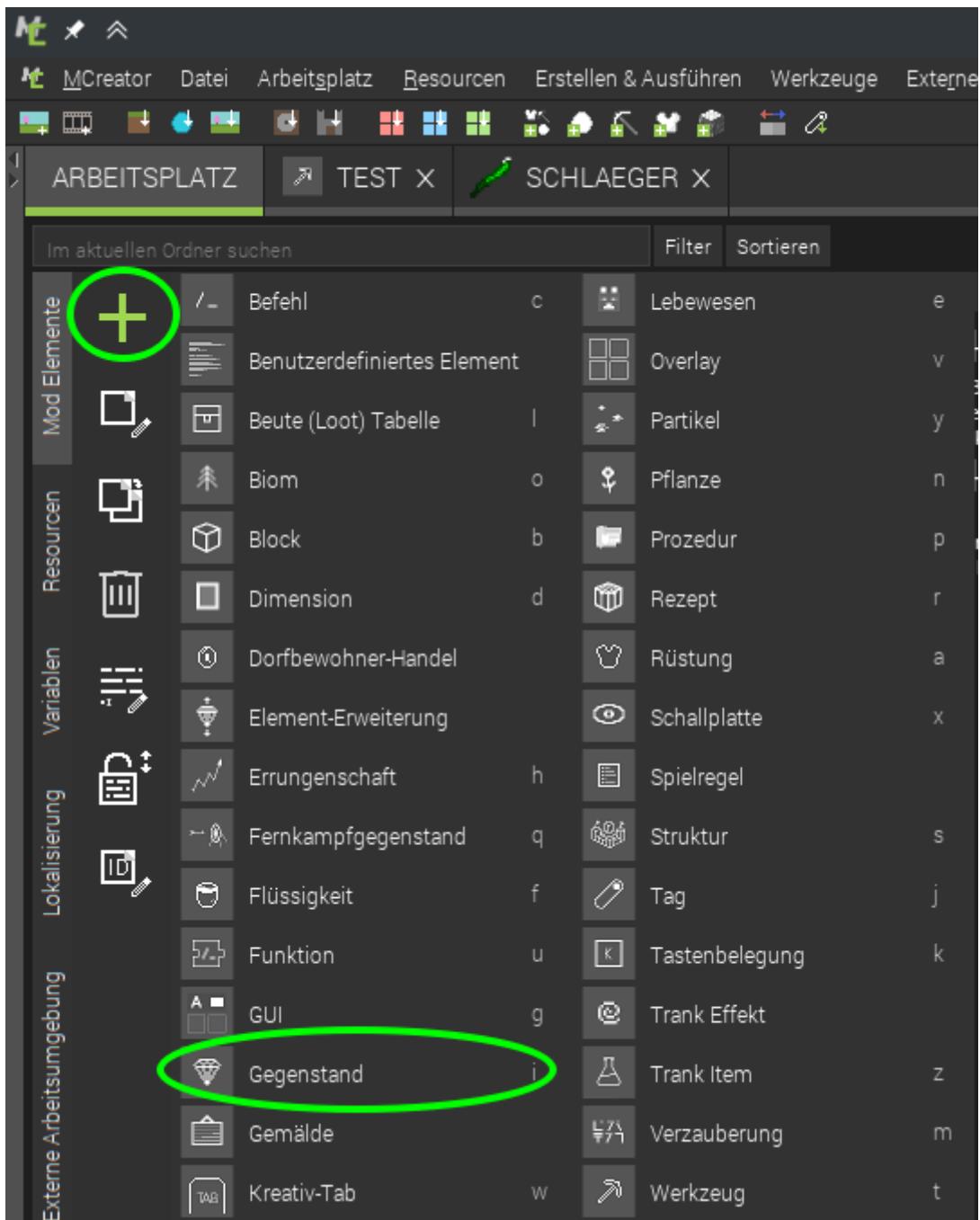
Starte dein Programm und teste alles noch einmal! Funktioniert alles? Für beide Farben?

## Level 5: Die Alpaka-Ball-Schläger

Stell dir vor, du spielst Eishockey, Golf oder Tennis - ohne Schläger. Hört sich komisch an, oder? Also brauchen wir noch den passenden Schläger: Da macht das Spiel noch mehr Spaß!

### Erstelle einen neuen Gegenstand

- Wechsle zum Arbeitsplatz
- Klicke auf “+” und wähle “Gegenstand” aus



**Abbildung 33:** Gegenstand erstellen

Der neue Gegenstand braucht natürlich einen Namen - "Schlaeger" zum Beispiel.

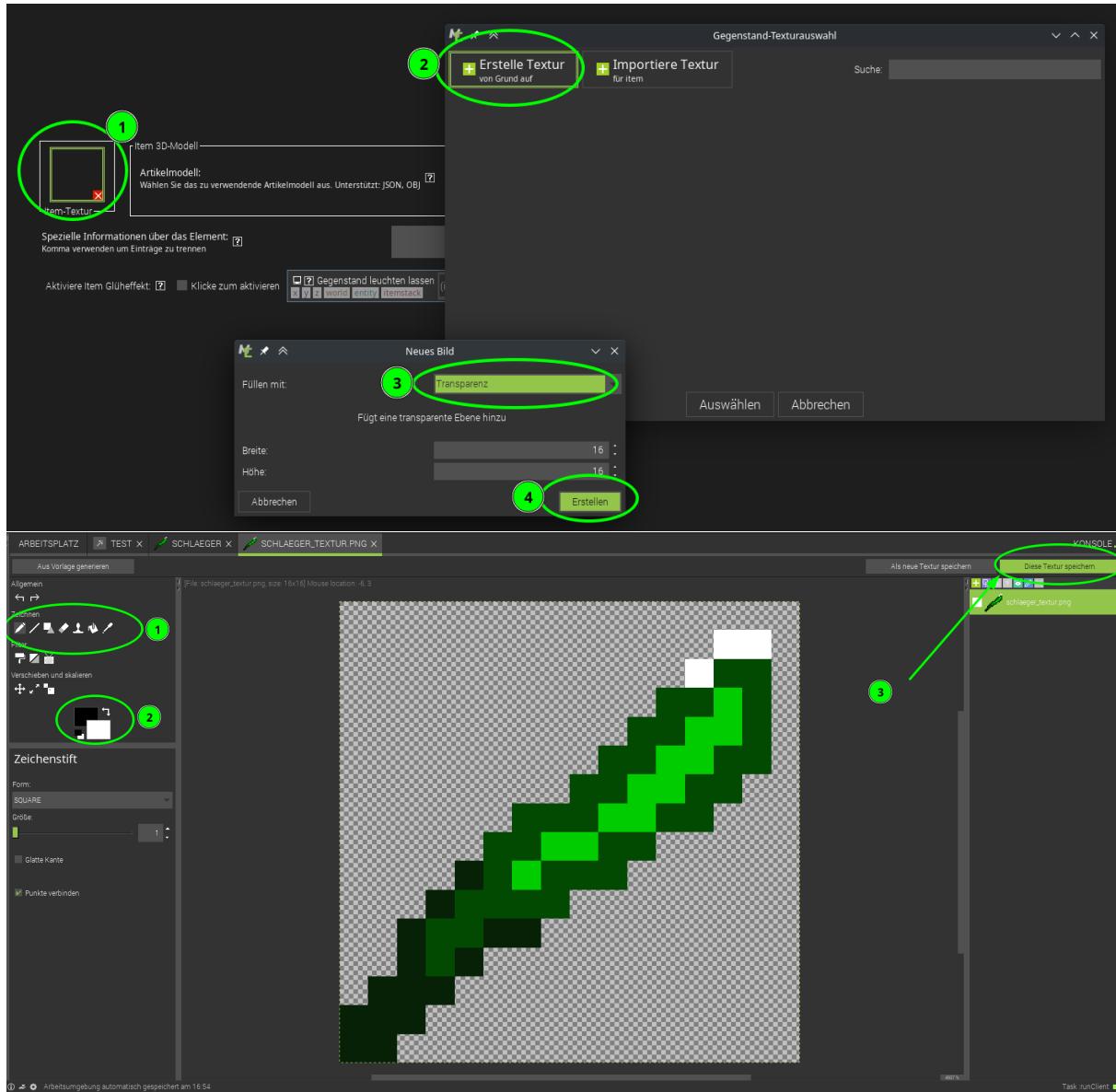


### **Umlaute und Programmieren**

Probier mal den Schläger auch so zu nennen mit Ä statt AE. Wenn du den Gegenstand angelegt hast, fällt dir vielleicht auf, dass er dann in MCreator “Schlager” heißt. Schlager? Komisch!

Das liegt daran, dass Computer an ganz vielen Stellen noch nicht mit Umlauten und erweiterten Zeichen wie Emoticons oder anderen Schriftarten umgehen können. Darum macht MCreator automatisch aus dem **Ä** ein **A**. Darum haben wir uns gleich für den Namen **Schlæger** entschieden.

## Textur für den Schläger pixeln

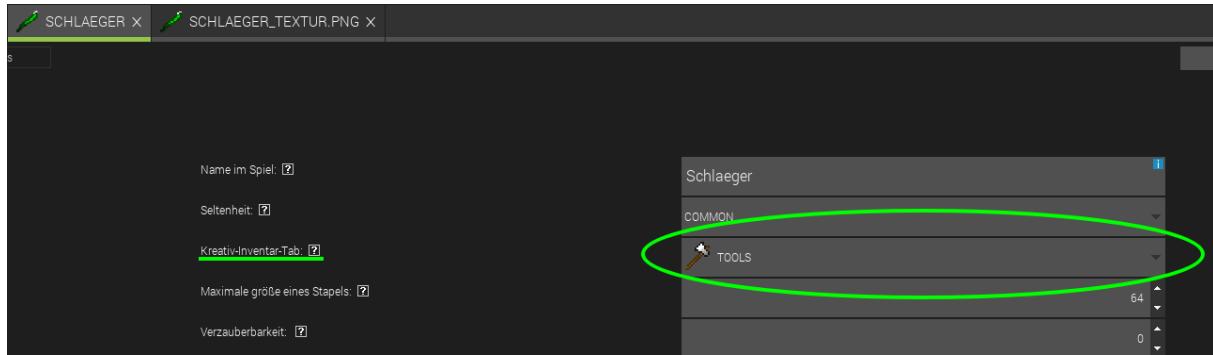


1. Wähle das Zeichentool aus
2. Wähle eine Farbe aus
3. Speichere es als **Gegenstandtextur!**

Dann wähle beim Gegenstand die gerade erstellte Textur aus.

## Eigenschaften des Schlägers

Jetzt müssen wir noch die Eigenschaften des Schlägers bearbeiten, damit auch alles so ist, wie wir das von einem Schläger erwarten. Wechsel dazu zu den *Eigenschaften*.



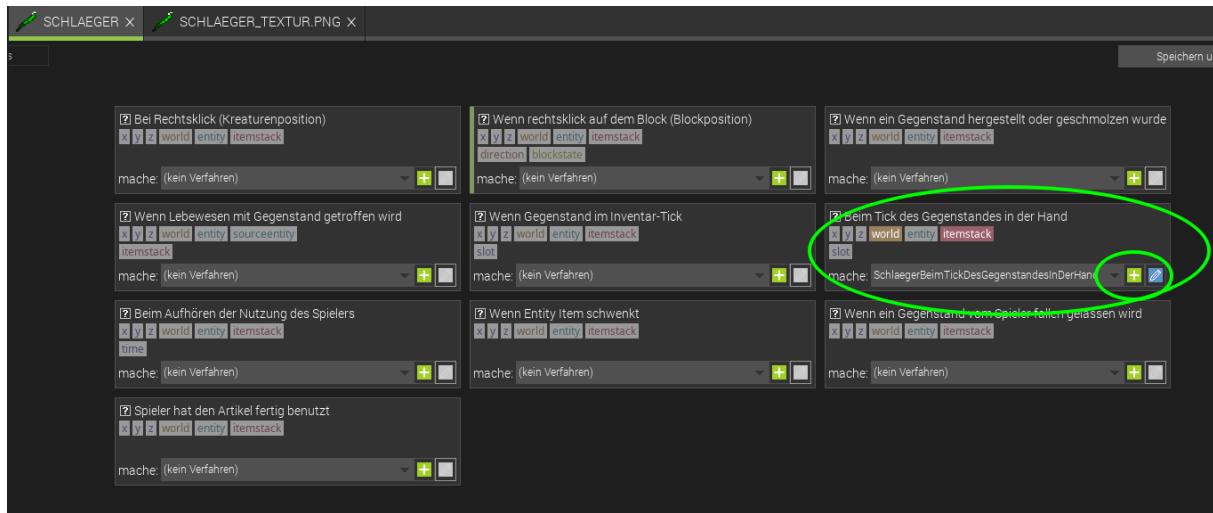
**Abbildung 34:** Gegenstand Eigenschaften

Als Inventar-Tab haben wir *Tools* ausgewählt, damit es bei den anderen Werkzeugen im Inventar auftaucht.

## Rückstoß

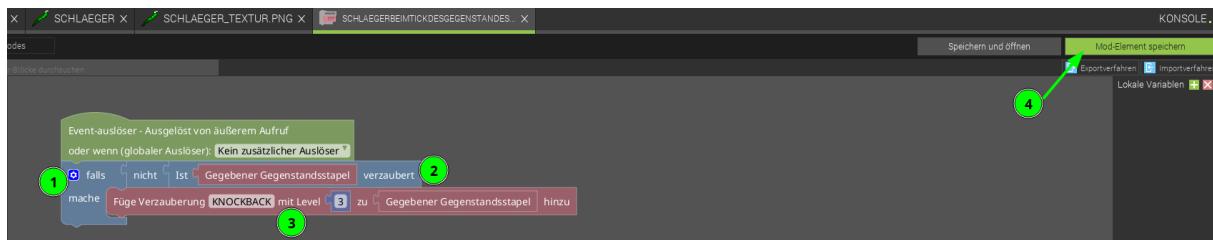
Der Schläger soll besondere Eigenschaften bekommen, wie ein echter Sport-Schläger. Da es aber in den normalen Eigenschaften keine Einstellung dafür gibt, wie viel Rückstoß der Schläger haben soll, behelfen wir uns anders.

Die Auslöser kann man nutzen, um auf Ereignisse im Spiel zu reagieren. Du nutzt jetzt einen neuen Auslöser, der immer läuft, wenn man ihn in der Hand hält:



**Abbildung 35:** Auslöser für den Schläger - Bei jedem Tick

Um den Rückschlag zu erstellen, “verzaubern” wir den Schläger einfach. Der Auslöser läuft “immer”, solange ein Spieler den Schläger in der Hand hat. Deswegen musst du etwas mehr programmieren:



**Abbildung 36:** Code zum Verzaubern des Schlägers

1. Überprüfe, ob der Gegenstand noch nicht verzaubert ist - sonst würde er jeden Tick neu verzaubert werden.
2. Dazu verneinen wir die Antwort auf die Frage *Ist Gegebener Gegenstandsstack verzaubert*. Das bedeutet, mit dem Befehl **nicht** drehen wir das Ergebnis von **ist ... verzaubert** um und erhalten **wahr**, wenn er nicht verzaubert ist.
3. Füge dann die Verzauberung **KNOCKBACK** (auf Deutsch “Rückstoß”) mit dem **Level \*3\*** hinzu.



### Ausprobieren!

Probiere es gleich aus! Klappt es? Dann müsste sich der Gegenstand sofort verzaubern, wenn du ihn in die Hand nimmst. Das siehst du im Spiel - der Schläger auf Slot 1 ist verzaubert (blau...), der auf Slot 2 nicht.



**Abbildung 37:** 1 ist in der Hand - und verzaubert!



### Alternative Schläger

Man kann auch mit einem Bogen (eventuell mit Schlag verzaubert) sehr gut spielen, wenn der Ball weiter weg ist.

Probiere es doch mal aus!

## Level 6: Das Spielfeld: Die Arena!

Du bist schon fast fertig! Schläger, Ball, Tore - alles da!

Jetzt fehlt eigentlich nur noch das komplette Spielfeld! Das bauen wir jetzt.

Los gehts!

1. Starte Minecraft mit deiner Mod.
2. Erstelle eine neue Welt.
  1. Spielmodus: Kreativ.
  2. Welttyp: Flachland.
  3. Spielregeln bearbeiten: "Kreaturen erzeugen **aus**" (sonst hüpfen überall die Schleime rum).

### Beispiel - so sieht unsere Arena aus



**Abbildung 38:** Beispiel Alpaka-Ball-Arena



#### Tipp: Schneller bauen

Keine Lust lange rumzubuddeln?

Minecraft Wiki: /fill - Schau dir mal den Fill-Befehl an!

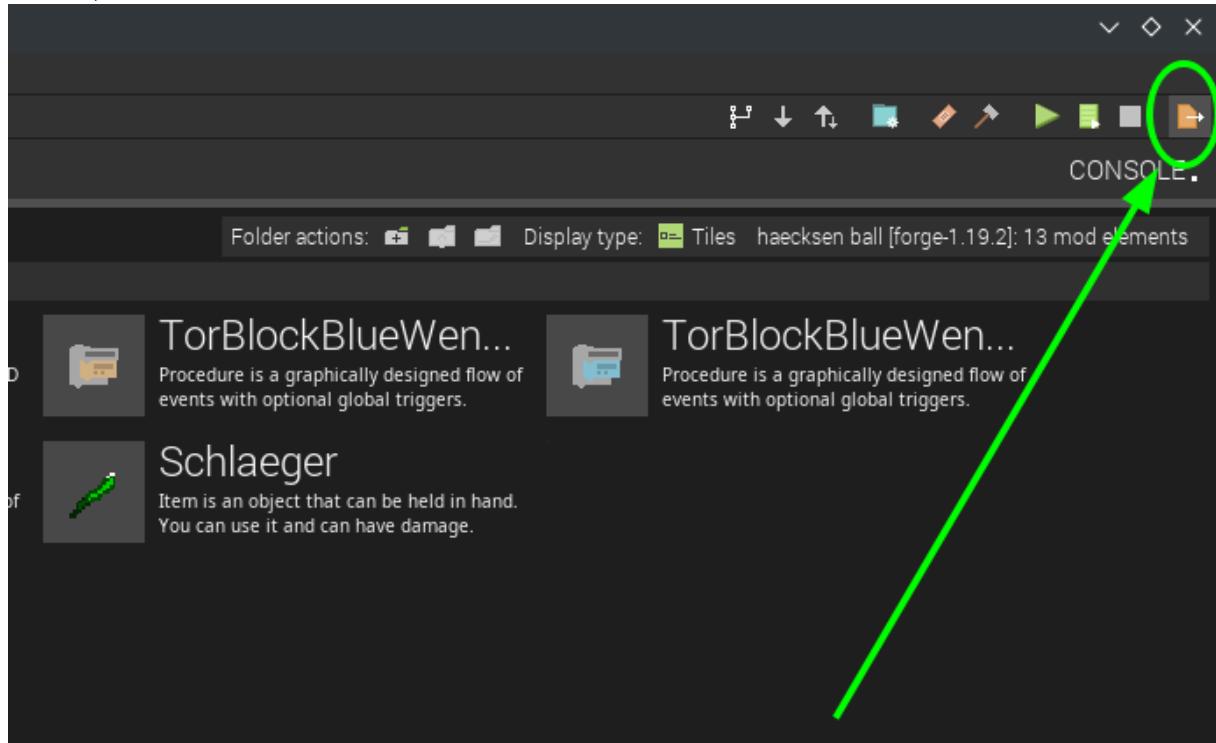
## Level 7: Exportieren & Installieren der Mod

So, alles ist fertig - jetzt wird es Zeit, die Mod zu exportieren und zu veröffentlichen, damit auch deine Freund\*innen die Mod herunterladen und installieren können.

### JAR-Datei exportieren

Um die Mod in Minecraft ohne MCreator zu spielen, müssen wir eine [.jar](#)-Datei erstellen.

Hier ist, wie du das machst:



Nachdem Klick auf das Exportieren-Symbol wird die [.jar](#)-Datei generiert, und MCreator fragt nach einem Speicherort:



Diese .jar-Datei kann wie jede andere Mod installiert werden.

## Mod Installieren

Die einfachste Möglichkeit, Mods zu installieren, ist die Verwendung eines erweiterten Launchers wie dem “[ATLauncher](#)”:

1. Lade den ATLauncher herunter und installiere ihn: [ATLauncher-Website](#)
2. Erstelle eine neue Minecraft-Instanz:
  1. Klicke auf “Vanilla Packs”.
  2. Wähle die richtige Version - 1.19.2 in unserem Fall.
  3. Wähle den “**Forge**”-Loader (wichtig!).
  4. Klicke auf “Create Instance”.
3. Wechsle zum Reiter “**Instances**”.
4. Klicke auf “**Edit Mods**”.
5. Klicke auf “**Add Mod**” und wähle die zuvor erstellte JAR-Datei aus.



## Mods & Server

Möchtest du gemeinsam auf einem Server spielen? Die einfachste Möglichkeit ist, das Spiel in Minecraft von einem Rechner aus freizugeben:

### ***Minecraft über LAN-Server spielen***

1. Starte das Spiel **Minecraft** wie gewohnt.
2. Wähle “Singleplayer” und starte die gewünschte Welt oder erstelle eine neue.
3. Klicke im Spiel auf die “Escape-Taste” und wähle anschließend “Für **LAN** öffnen”.
4. Passe die Einstellungen an und klicke auf “OK”.

Wenn du auf einem öffentlichen Server im Internet gemeinsam spielen möchtest, muss dieser ebenfalls Forge verwenden und die Mod installiert haben.