

---

## **MCreator: Erstelle deine eigene Mod HaecksenBall!**

KidsLab.de gGmbH - MatzE, KingBBQ

2022-12-22

## Übersicht

<b>Eigene Minecraft-Mod erstellen: Haecksen-Ball!</b>	<b>4</b>
Einführung . . . . .	4
Was brauche ich dafür? . . . . .	4
Vorbereitung . . . . .	5
Online-Version mit Links zu weiteren zugehörigen Dokumenten . . . . .	5
 <b>Einführung in MCreator &amp; Unser Projekt</b>	 <b>5</b>
Unser Spiel: Haecksen-Ball . . . . .	5
Mods, MCreator, Java - eine kleine Einführung . . . . .	5
Was ist MCreator? . . . . .	6
Mods? . . . . .	6
 <b>Vorbereitung: Neues Projekt in MCreator anlegen</b>	 <b>7</b>
Achtung . . . . .	9
Übersicht über die IDE . . . . .	9
Minecraft Wiki . . . . .	11
 <b>Level 1: Der Ball für unser Spiel</b>	 <b>11</b>
Verhalten bearbeiten: . . . . .	16
 <b>Level 2: Tor! Oder: da muss der Ball rein</b>	 <b>19</b>
Anzeige des Spielstandes - wieviele Tore wurden geschossen? . . . . .	22
Wie werden die Tore gezählt? Auslöser! . . . . .	23
“Mal Nachschauen:” . . . . .	27
Was ist mit der gegnerischen Mannschaft? . . . . .	27
“Geschafft!” . . . . .	30
 <b>Level 3: Tore / Punkte zählen</b>	 <b>30</b>
Punkte merken . . . . .	30
Tore erkennen . . . . .	31
Punkte abziehen . . . . .	35
Erfolg . . . . .	35
Jetzt nochmal für Orange . . . . .	35
“Funktioniert?” . . . . .	38
 <b>Level 5: Erweiterte Punkteanzeige / Punkte zurücksetzen</b>	 <b>38</b>
“Aufgepasst! Platz lassen” . . . . .	39
“Testen schadet nie ;)” . . . . .	39

<b>Level 5: Die Haecksenball-Schläger</b>	<b>39</b>
Kleiner Tipp . . . . .	44
<b>Level 6: Das Spielfeld: Die Arena!</b>	<b>44</b>
<b>Level 7: Exportieren &amp; Veröffentlichen der Mod</b>	<b>44</b>



**Abbildung 1:** Jugend hackt schriftzug

## Eigene Minecraft-Mod erstellen: Haecksen-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 in Minecraft - und damit dein “Spiel im Spiel”: Haecksenball!

MCreator (<https://mcreator.net/>), eine Open-Source Entwicklungsumgebung, bietet dazu alles nötige: vom einfachen Start bis pixeln der Blöcke, einfache Programmierung. Das ganze Spiel 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

---

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

---

## Vorbereitung

- MCreator auf allen Computern installieren (<https://mcreator.net/install>)
- Leere Arbeitsumgebung installieren (der initiale Download aller Bibliotheken dauert sonst noch mal 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>

## Einführung in MCreator & Unser Projekt

### Unser Spiel: Haecksen-Ball

Aber was machen wir hier eigentlich? Mit Mods lässt sich Minecraft ja individuell erweitern. Da Minecraft ja vom Prinzip "Spiel im Spiel" lebt - also, dass man eigene Spiele in Minecraft spielt, die nichts mit dem ursprünglichen Prinzip zu tun haben. Gute Beispiele sind zum Beispiel Bedwars oder Builder-Challenge.

Wir haben uns für unsere Spiel-Idee an dem Sport [Quidditch](#) orientiert, der ja von Zauberern und Hexen gespielt wird. Um das ganze noch perfekt in den Hacker-Kontext einzubetten, haben wir es (Haeckse-)[<https://de.wikipedia.org/wiki/Haeckse>]Ball genannt: eine Haeckse ist ein weiblicher Hacker :)

### Mods, MCreator, Java - eine kleine Einführung

Wenn Du gerne Minecraft spielst, ist dir bestimmt schon mal 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, erblickt die erste wirkliche Minecraft-Version 0.0.9a das Licht der Welt!

### Was ist MCreator?

“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/>)

### Mods?

Minecraft hat seinen großen Erfolg nicht nur den Entwicklern von Mojang zu verdanken - schon sehr früh konnten externe Programmierer eigenen Erweiterungen zu Minecraft hinzufügen: sogenannte Modifikationen oder kurz “[Mods](#)”

Mods gibt es nur in der Java-Version. Mods 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 dann jeweils eine sog. JAR Datei (JAR, englisch für “Krug” - oder Java ARchive). Wenn die an der richtigen Stelle liegen, ist dein Minecraft gemodded!



Lust mal ne Mod zu installieren? Ich würde 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 :\)](#)

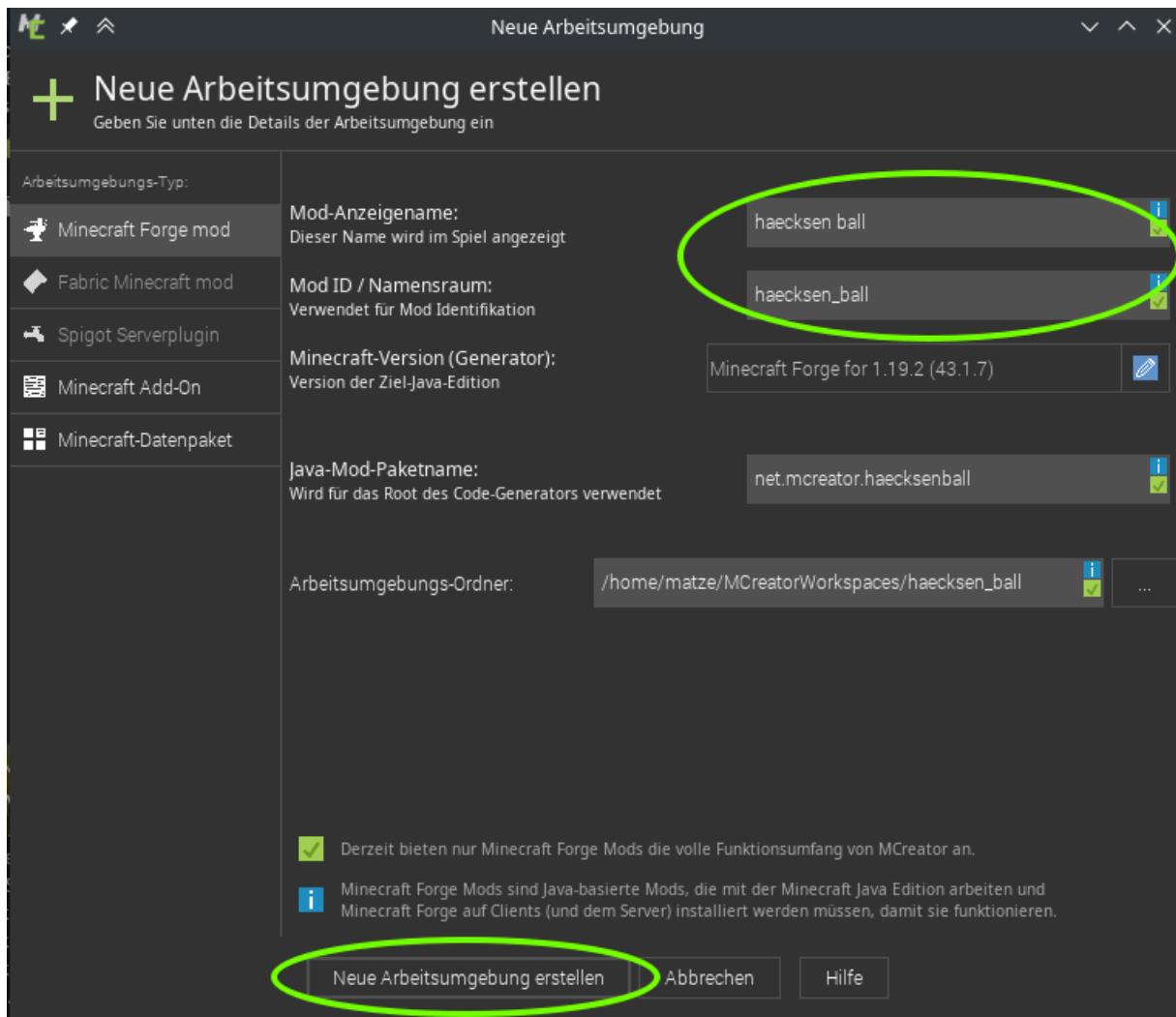
## Vorbereitung: Neues Projekt in MCreator anlegen

Öffne im ersten Schritt MCreator. Unter allen Betriebssystemen (Windows, macOS und Linux) ist MCreator eine normale Anwendung und kann wie jede Andere gestartet werden.

Als erstes erstellst Du jetzt ein neues Projekt:



**Abbildung 2:** Neues Projekt anlegen - 1



**Abbildung 3:** Neues Projekt anlegen - 2

Folgende Felder füllst Du nun aus: - **Mod-Anzeigename** - wie soll deine Mod heißen? - **Mod-ID** - Interner Name, der muss ohne Leerzeichen etc. sein - **Minecraft-Version** - Mods müssen passend zur Minecraft Version sein - hier wählst Du bitte für dieses Tutorial “Minecraft Forge for 1.19.2”

Klicke dann auf “Neue Arbeitsumgebung erstellen” - es öffnet sich eine Internetseite die um eine Spende bittet.

Der Download und erste Build dauert je nach Internet und Computer ein bisschen. Dadurch wird uns aber viel Arbeit abgenommen die wir sonst von Hand machen müssten, also warten wir gerne :)

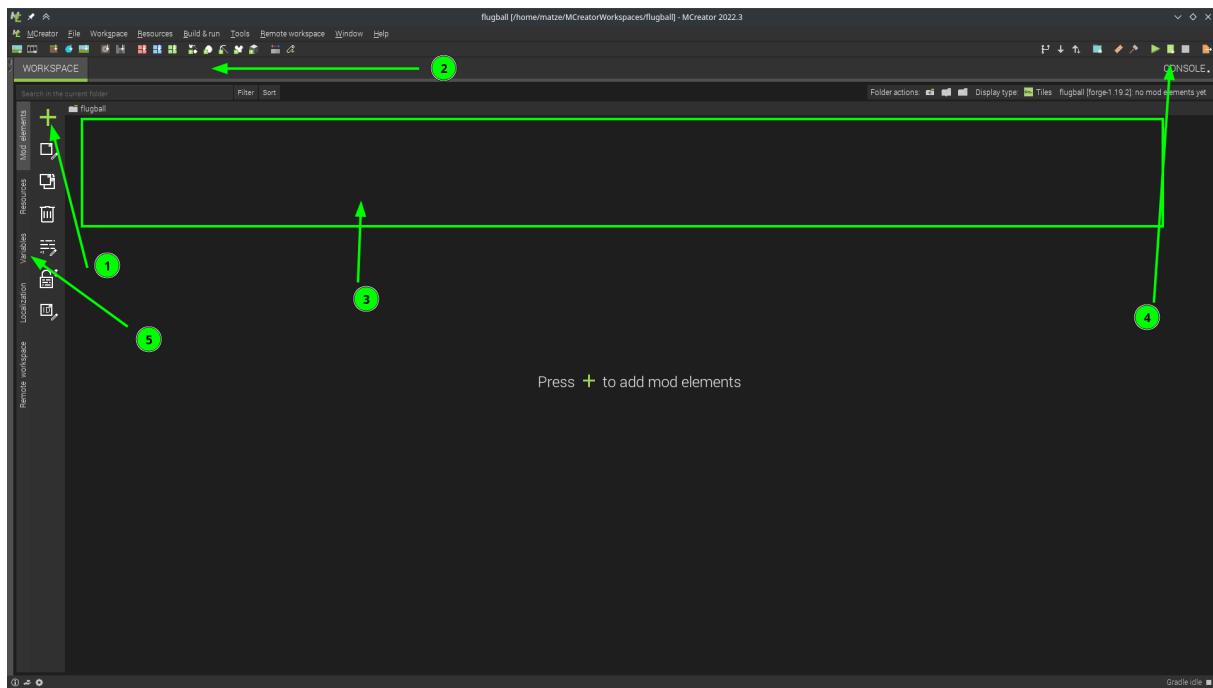
**Achtung**

Manchmal wartet man ewig und es war eigentlich schon fertig. Wenn man in der Konsole eine erfolgsnachricht sieht, und nichts mehr passiert, kann es sein, dass man das Popup fenster schließen kann und trotzdem alles fertig installiert ist.

**Übersicht über die IDE**

**ID.... was?** [IDE](#) steht für Integrated Development Editor oder Integrierte Entwicklungs-umgebung. Das ist ein Programm, dass alle nötigen Tools zum Programmieren zusammenfasst - unsere Arbeitsumgebung für dieses Tutorial.

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



**Abbildung 4:** IDE Übersicht: das sind die wichtigsten Elemente

1. mit dem + kann man neue Elemente erstellen, z.B. Blöcke, Tiere, Gegenstände ...
2. in dieser Leiste sieht man welche Elemente gerade zur Bearbeitung geöffnet sind.
3. in dem großen (noch leeren) Bereich sieht man dann eine Übersicht über alle erstellten Elemente.
4. hier sind mehrere kleine Knöpfe wichtig:  
sind manche Symbole ausgegraut, können diese aktuell nicht geklickt werden.
  - das grüne Dreieck: startet Minecraft mit der neu erstellten Mod
  - das rote Viereck: stoppt was gerade läuft, also z.B. wenn man Minecraft noch gestartet hat
  - die orangene Datei mit dem Pfeil: exportiert die Mod zur Verwendung mit anderen Mod-Launchern.
5. hier kann man Variablen erstellen, das brauchen wir später auch noch :)

**{}< notice mentor >}}** **Nützliche Einstellungen im Spiel** Vanilla-Einstellungen, also Dinge die ohne Mod funktionieren:

- Zeit auf Tag setzen und auf Tag lassen  
`/time set day & /gamerule doDaylightCycle false`
- Wetter auf schön setzen und wetter so lassen `/weather clear & /gamerule doWeatherCycle false`

...

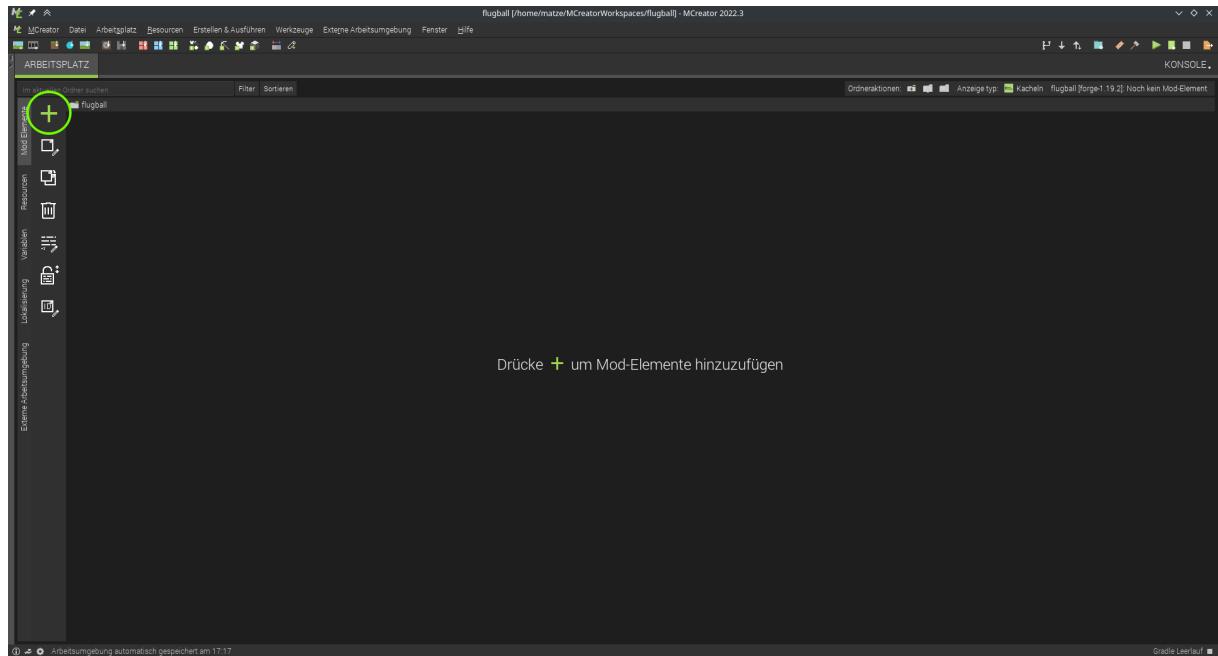


## Minecraft Wiki

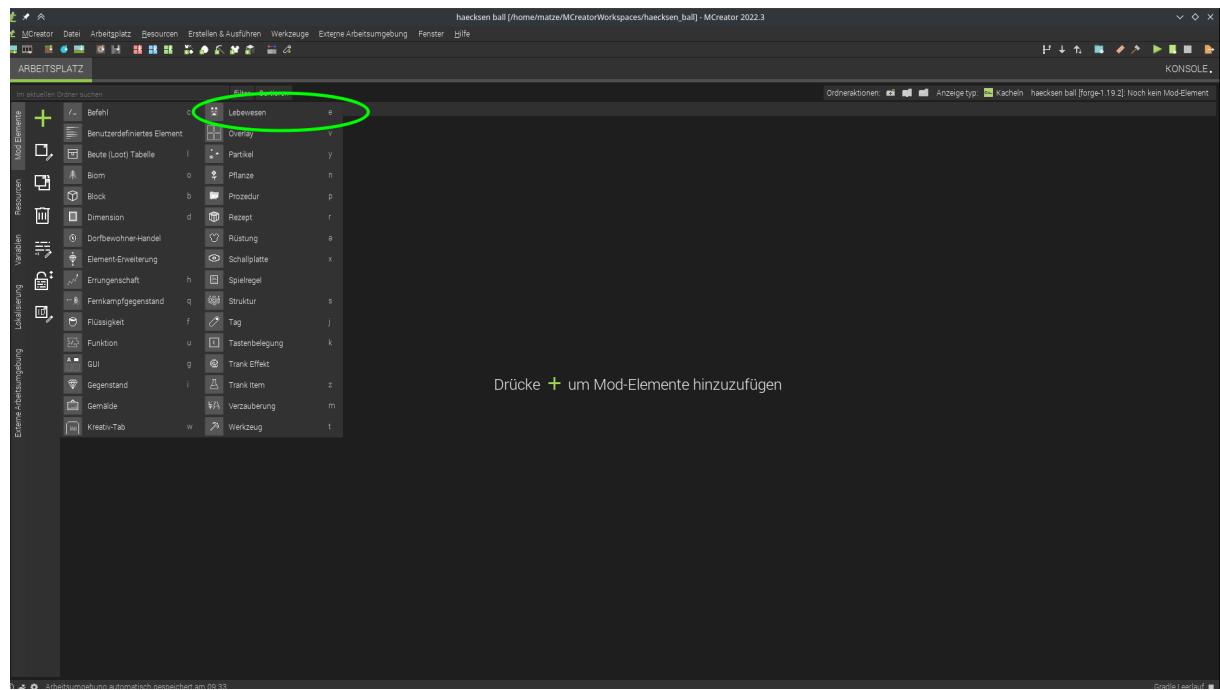
Es gibt noch sehr viel weitere praktische [gamerules](#): [Minecraft Wiki Gamerules](#)

## Level 1: Der Ball für unser Spiel

Als erstes brauchen wir einen passenden Ball, er ist das Zentrum unseres Spiels. Es gibt in Minecraft ja viel verschiedene “Sachen” - Blöcke, Pflanzen, Mobs. Für den Ball haben wir uns für ....

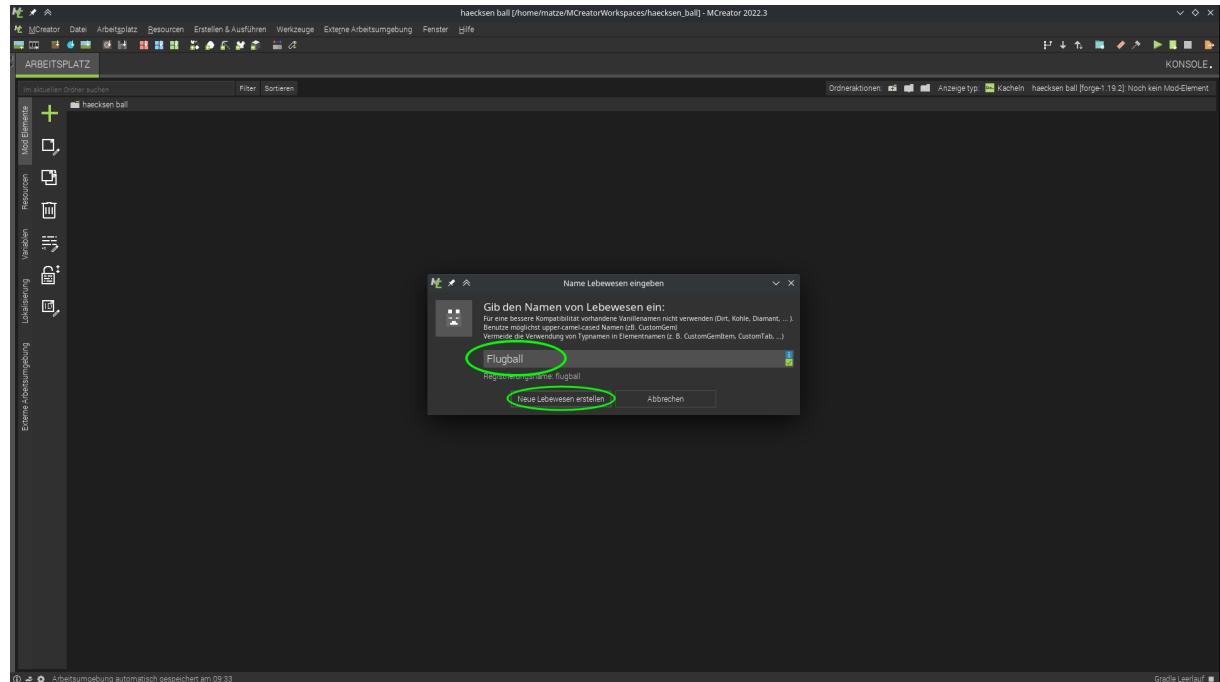


Du kannst neue Objekte über das “+” Plus Zeichen hinzufügen. Wähle dann “Lebewesen” aus:



Lebewesen sind “Entities”, die sich bewegen können - anhand von Regeln und KI, die Du festlegen kannst - das passt doch perfekt!

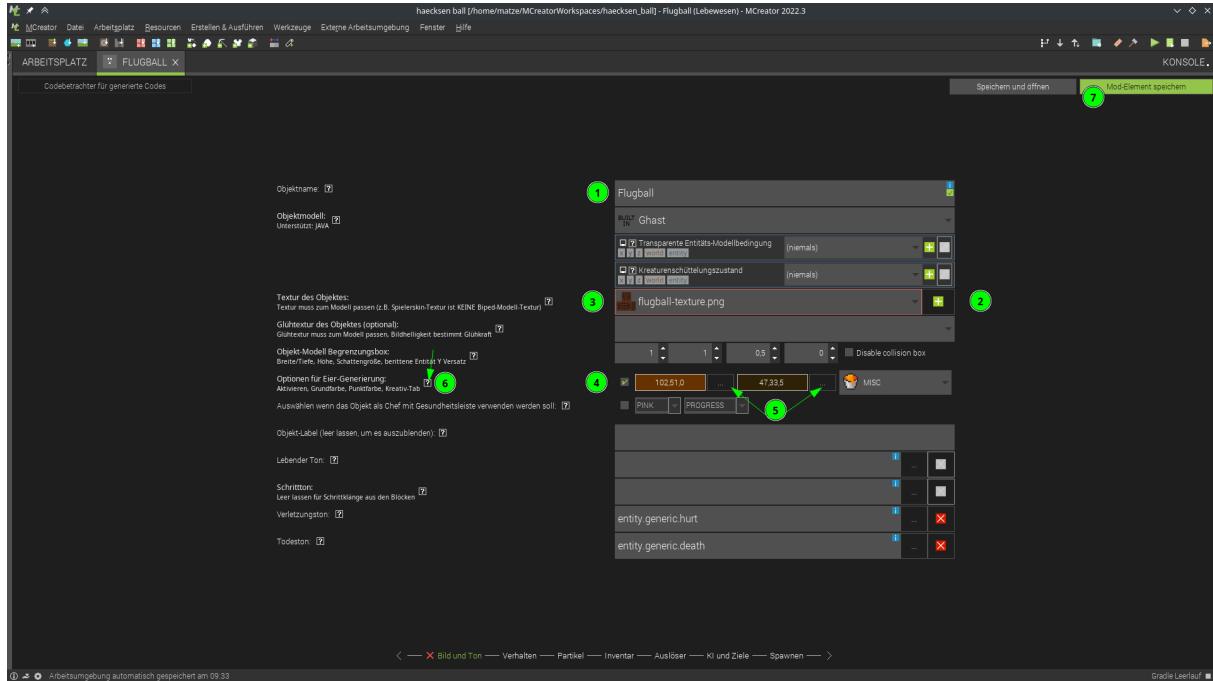
Natürlich musst Du einen Namen vergeben:



Jetzt öffnen sich das Fenster, in dem Du alle Eigenschaften des neuen erstellten Lebewesens festlegen kannst. Bei der Bearbeitung von einem Objekt gibt es verschiedene Kategorien (in der Unteren Leiste

auswählbar).

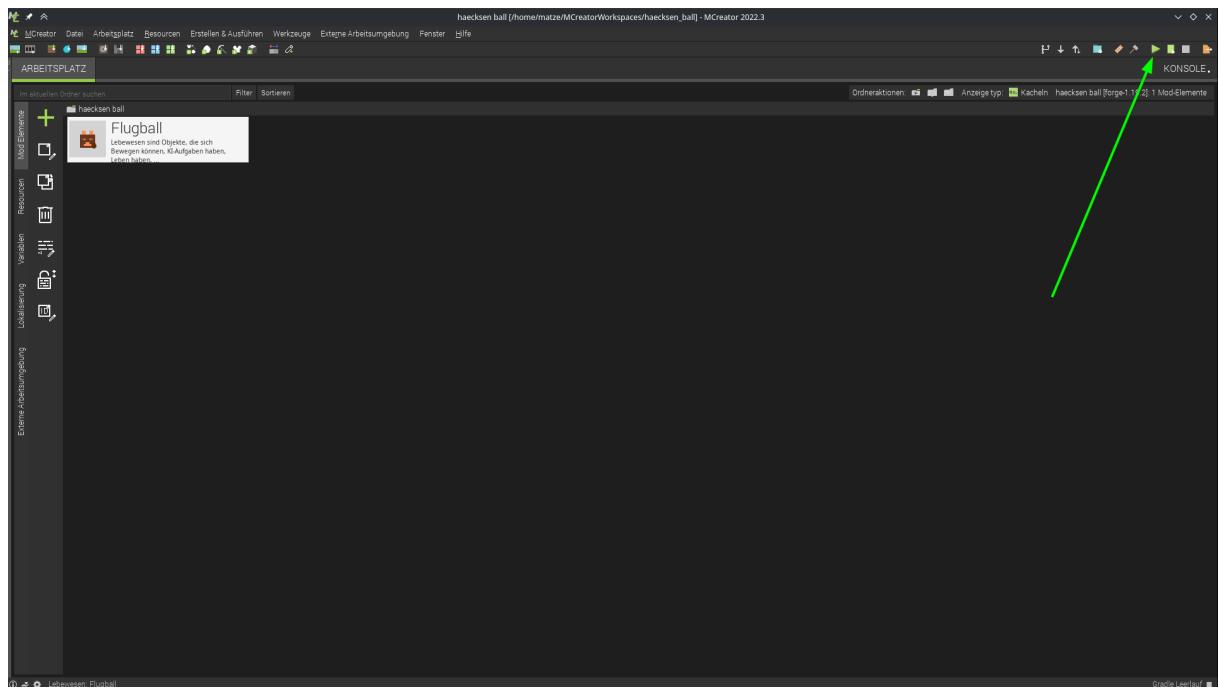
Bei *Bild und Ton* ist folgendes wichtig für unseren Ball:



**Abbildung 5:** Erstelle ein neues Lebewesen

1. Name wird übernommen aus vorigem Dialog
2. Die Textur bestimmt das Aussehen Deines Balls - Du fügst die Textur mit dem **+ Plus** hinzu. Du kannst selbst eine Textur erstellen oder diese als Beispiel benutzen:
3. Erst danach kannst Du die Textur im Dropdown auswählen
4. Wir lassen uns auch ein Spawn-Ei generieren
5. Für das Ei können wir bei den 3 Punkten neben dem Farbfeld die Farben auswählen die das Ei haben soll.
6. Zu jedem Attribut, das man hier auswählen kann gibt es bei dem fragezeichen einen Hilfetext
7. Speichern nicht vergessen!

Nach dem speichern dauert es eine kleine Weile, bis MCreator die Mod daraus erstellt hat, wenn dann das Play-Dreieck oben rechts grün ist kann man mit dem Play-Dreieck dann Minecraft direkt mit der neuen Mod starten:



**Abbildung 6:** Starte Minecraft mit deiner Mod: der Play-Knopf



Da passiert ganz viel im Hintergrund - unter **Konsole** kannst Du Dir anschauen, was MCreator alles machen muss, um die Mod zu kompilieren.

Wenn Minecraft gestartet ist, erstelle eine Welt im [Kreativ-Modus](#) - damit hast Du direkt Zugriff auf alle Gegenstände und kannst den Ball am einfachsten testen.

Das erstellte Spawn-Ei versteckt sich im Kreativ Inventar:



**Abbildung 7:** Inventar: hier findest Du dein erstelltes Lebewesen, bzw. dessen Ei

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



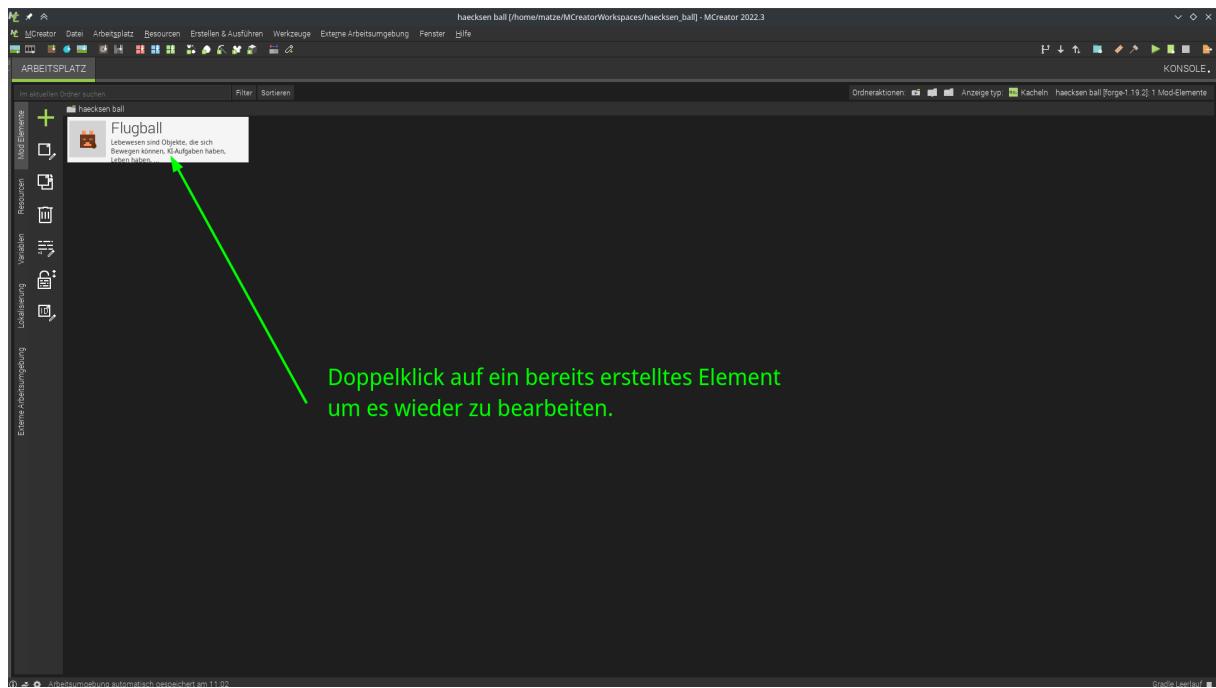
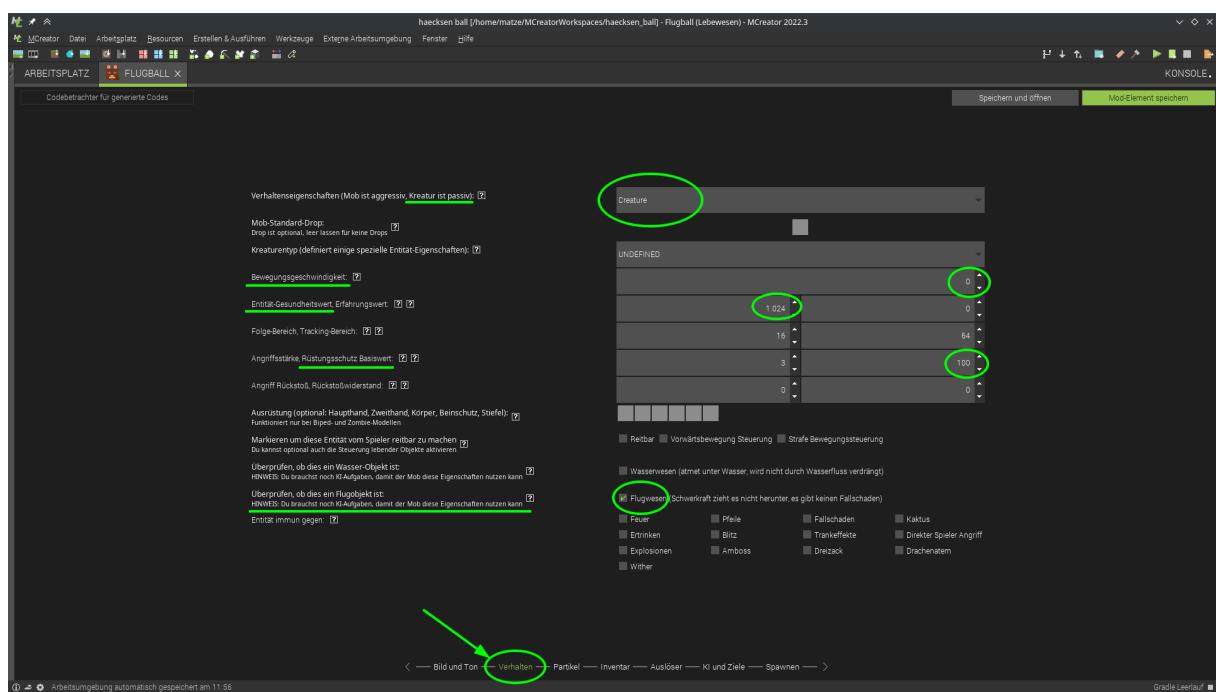
Tadaaa! So sieht es bis jetzt aus:



- der Ball fällt und fliegt noch nicht - der Ball läuft selbst noch herum - Ball kann noch sterben werden wenn man ihn schlägt :(

### Verhalten bearbeiten:

Jetzt wollen wir aber dem Ball noch beibringen, dass er sich wie ein Ball verhalten soll. Schließe Minecraft und gehe dazu wieder in MCreator. Mit Doppelklick auf ein Element kann dieses wieder bearbeitet werden:

**Abbildung 8:** Bearbeiten der Eigenschaften: Verhalten ändern**Abbildung 9:** Hier kannst Du alles zum Verhalten anpassen

Um das Verhalten zu verändern, wechselst Du unten auf den Verhaltens-Reiter. Jetzt kannst Du die Werte ändern:

- **creature** damit der Ball nicht angreift
- **bewegungsgeschwindigkeit** = 0, damit er uns nicht von alleine weg fliegt
- unser Ball soll nicht sterben wenn man ihn schlägt
  - **Gesundheitswert** = 1024 (höchst möglicher wert, (2er-Potenz ;))
  - **Rüstungsschutz** = 100 (auch höchst möglicher wert)



Das hast Du schon geschafft:



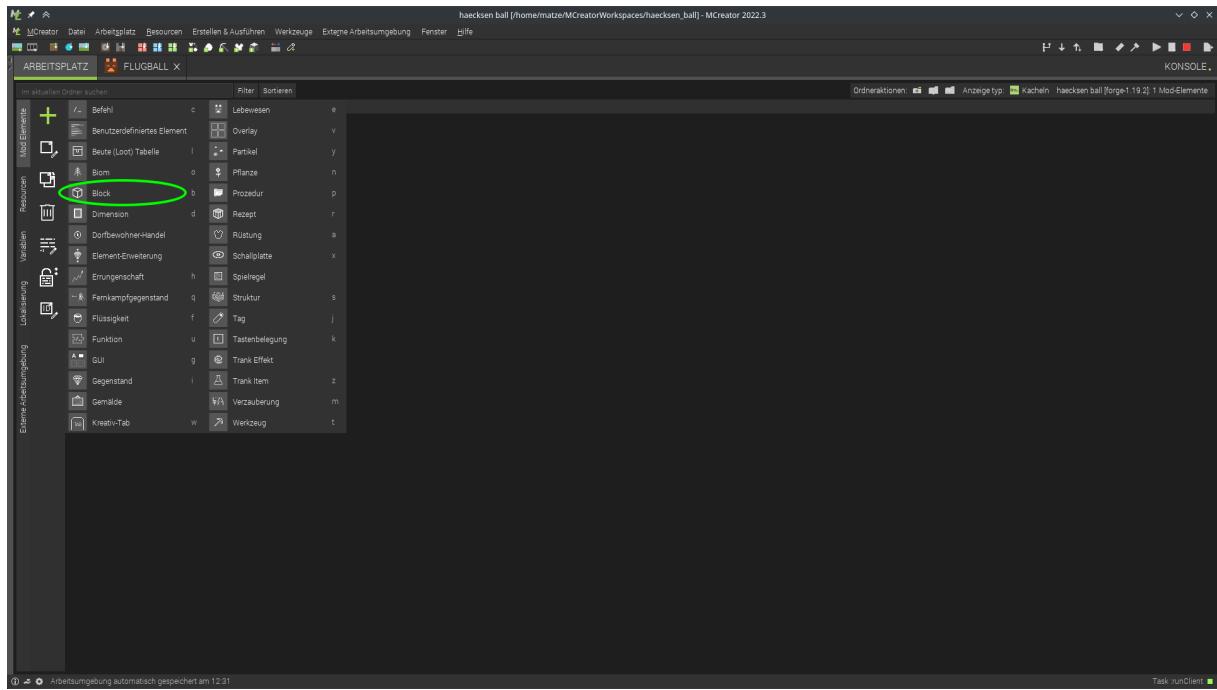
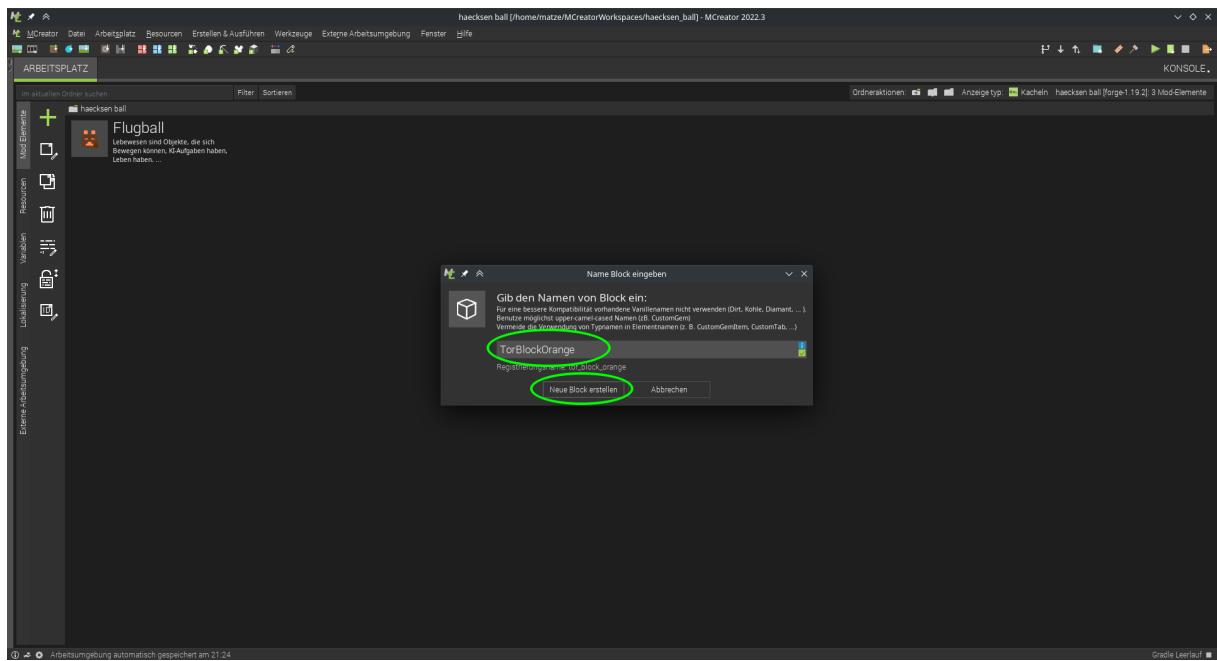
**Abbildung 10:** Stand Level 1 - Bälle sind geschafft!

- der Ball fällt nicht mehr, sondern fliegt
- der Ball läuft nicht mehr herum
- Ball stirbt nicht mehr ;)
- macht jetzt schon Spaß den Ball mit Schlag oder Rückstoß verzauberten Schwert und Bogen zu schubsen ;D

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

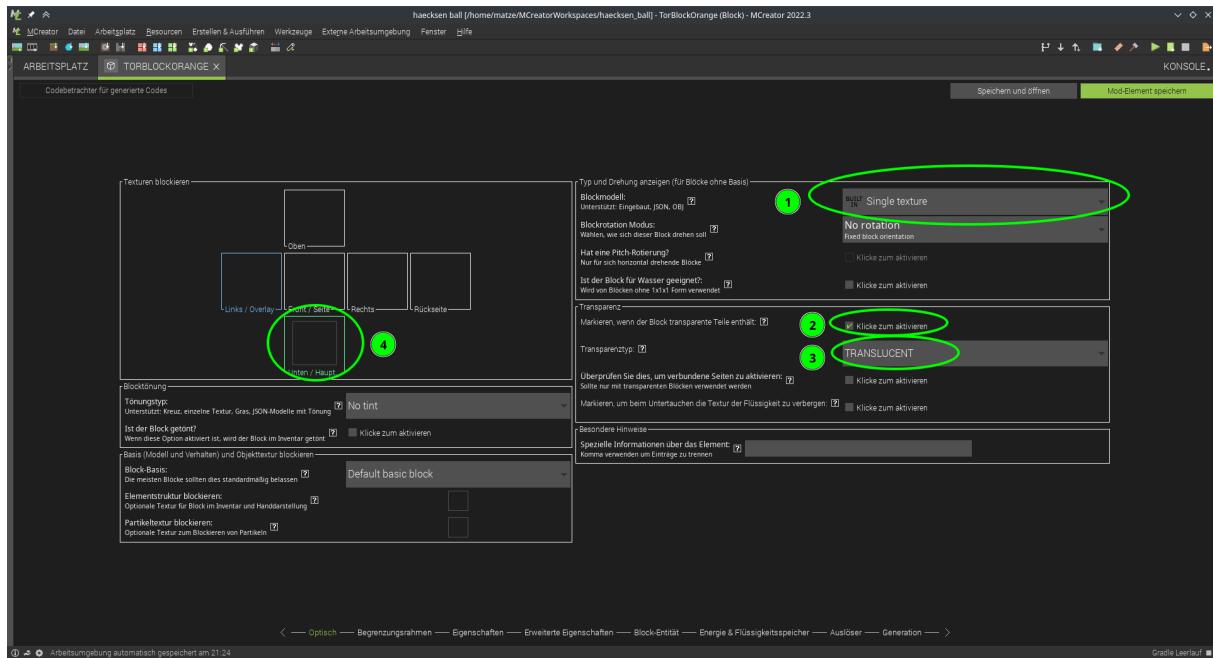
Wir haben uns überlegt, wie man das Tor am besten darstellen könnte. Blöcke erschienen uns am passendend: Unser Tor soll aus Blöcken gebaut werden können.

In diesem Level erstellst Du also einen eigene Block, mit dem dann das Tor im Spiel gebaut werden kann.

**Abbildung 11:** Erstelle einen neuen Block**Abbildung 12:** Gib dem neuen Block einen (sinnvollen) Namen

Jetzt kannst Du die Eigenschaften des Blocks anpassen - das kennst Du ja schon aus unserem ersten

Level, dort haben wir die Eigenschaften des “Lebewesens” Balls angepasst.



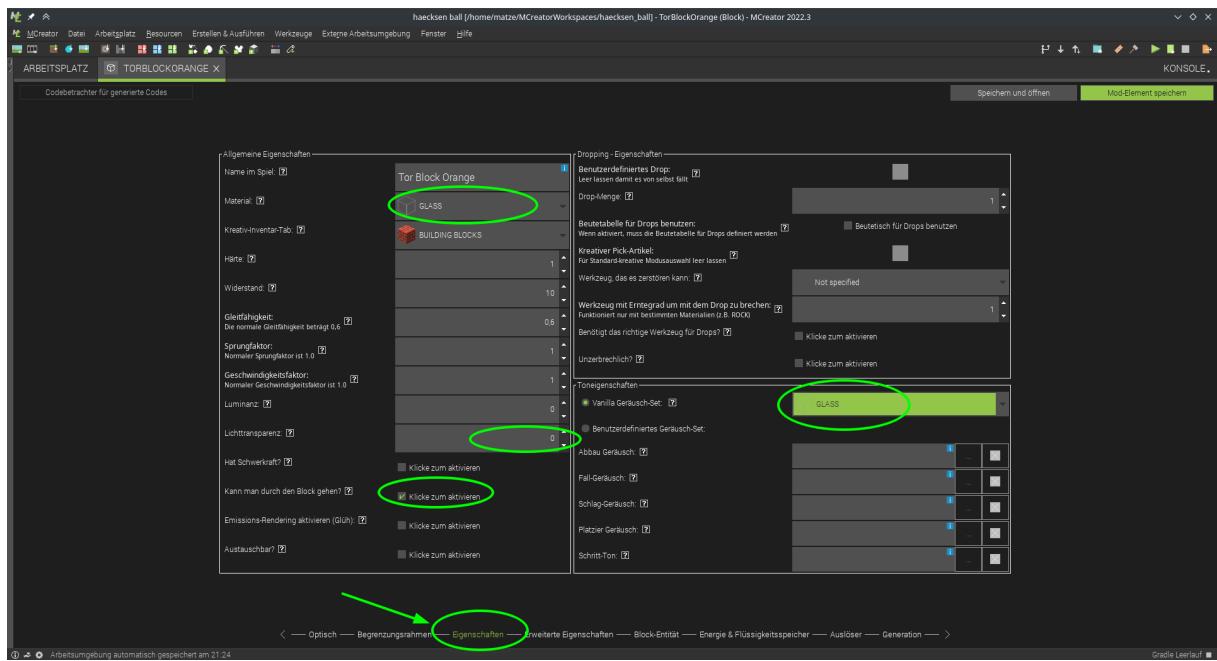
**Abbildung 13:** Die optischen Eigenschaften des Blocks

1. Wähle hier aus, auf allen Seiten die gleiche Textur verwendet werden soll, das spart uns Arbeit ;)
2. die Beispieltextur (schritt 4) ist halb transparent, deshalb wähle hier aus, dass die 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**”

Natürlich kannst Du auch selbst kreativ werden und deine eigene Textur erstellen!

Gib dem Block jetzt glasähnliche Eigenschaften:



**Abbildung 14:** Eigenschaft: Glass-ähnlich

Wir haben verschiedene Sachen getestet, wir haben uns entschieden, den Block so zu konfigurieren, dass man durch ihn laufen kann.



**Probier es aus!** Beim Programmieren ist es immer gut, wenn man zwischendurch immer mal wieder testet und ausprobiert. Jetzt ist ein guter Zeitpunkt: drücke den grünen Play-Knopf (rechts oben) und teste das ganze mal!

### Anzeige des Spielstandes - wieviele Tore wurden geschossen?

Wie könnte man 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.

Dazu benutzt man 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. Stiel auswählen (schön, passend zum Max-wert)

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

## 5. ... und jetzt noch anzeigen:

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

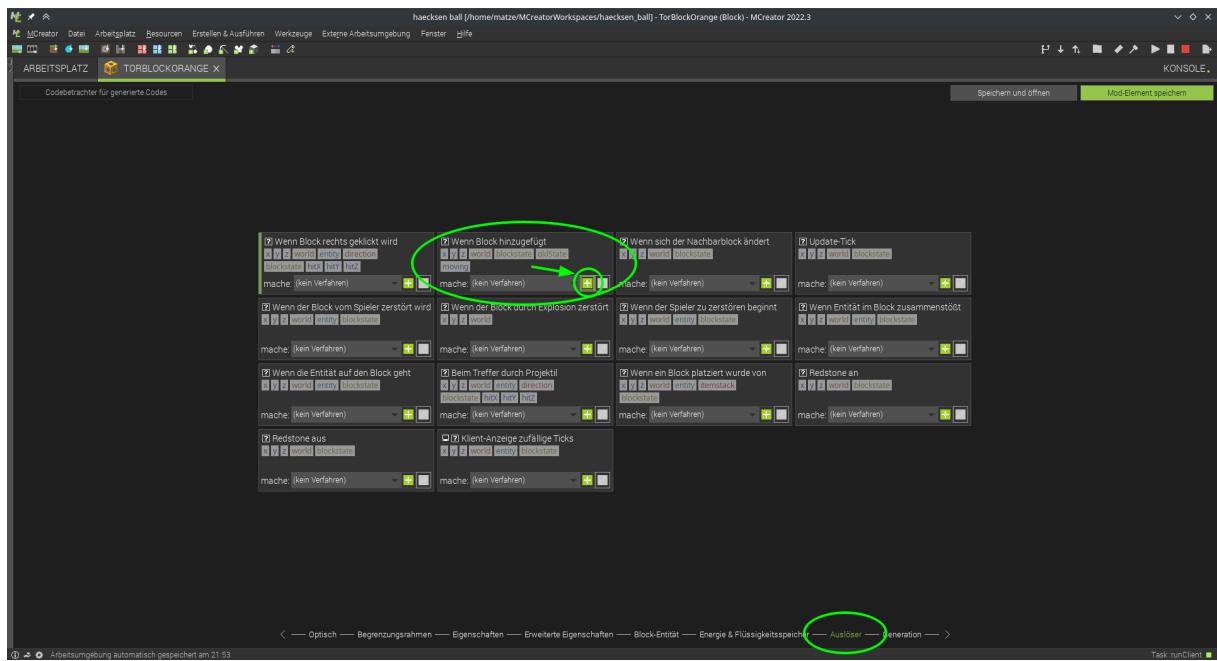


Probiere es aus! Öffne Minecraft (am einfachsten wieder über MCreator, grüner Knopf...) und starte eine Kreativ Welt. Und gib die Befehle wie oben Schritt für Schritt in der Konsole ein. Klappt es? Dann solltest Du jetzt eine Bass-Bar sehen!

## Wie werden die Tore gezählt? Auslöser!

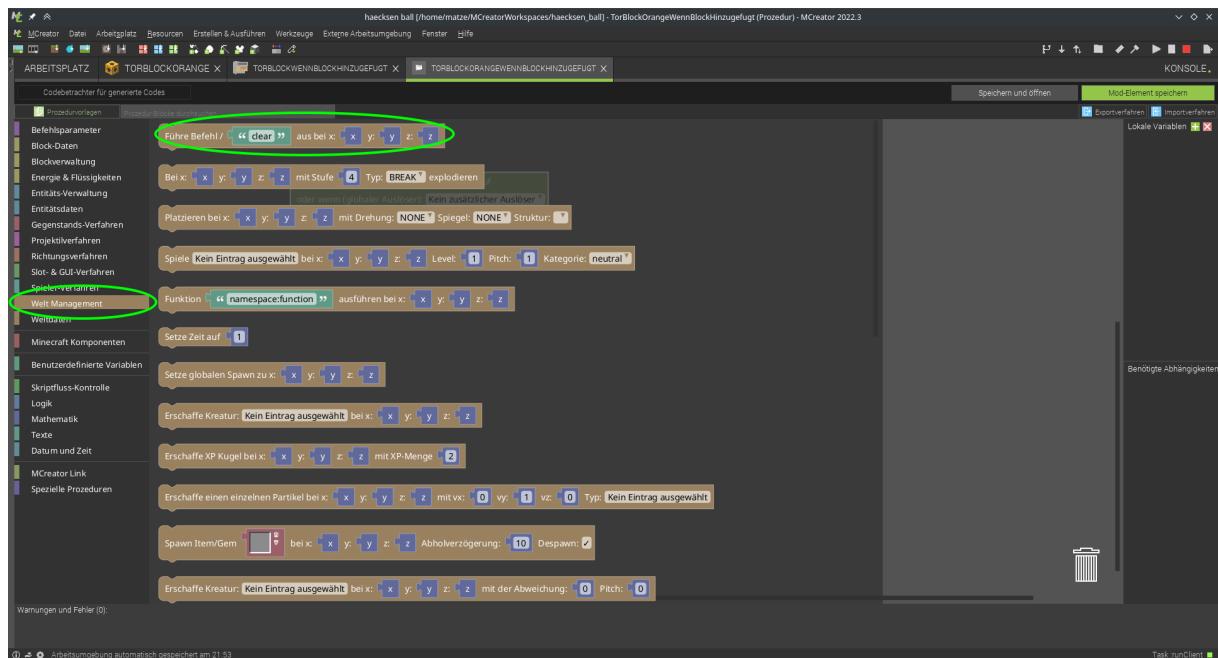
Jetzt muss das Tor ja wissen, dass ein Ball rein geflogen kommt. Wie könnte man das am besten anstellen? **Auslöser** bieten die Möglichkeit auf gewisse Ereignisse in Minecraft zu reagieren und unseren Code auszuführen, z.B. wenn ein Block gesetzt wird. Gehe noch mal in die Eigenschaften des Tor-Blocks und wähle unten “Auslöser” aus. Jetzt geht’s ans eingemachte und Du schreibst deinen ersten Code für die Mod: wähle im Bereich **Wenn Block hinzugefügt** aus, dass du eine neue Prozedur erstellen willst - wie gewohnt mit dem **+** **Plus** Knopf.

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



**Abbildung 15:** Auslöser hinzufügen - erzeugt ein Ereignis

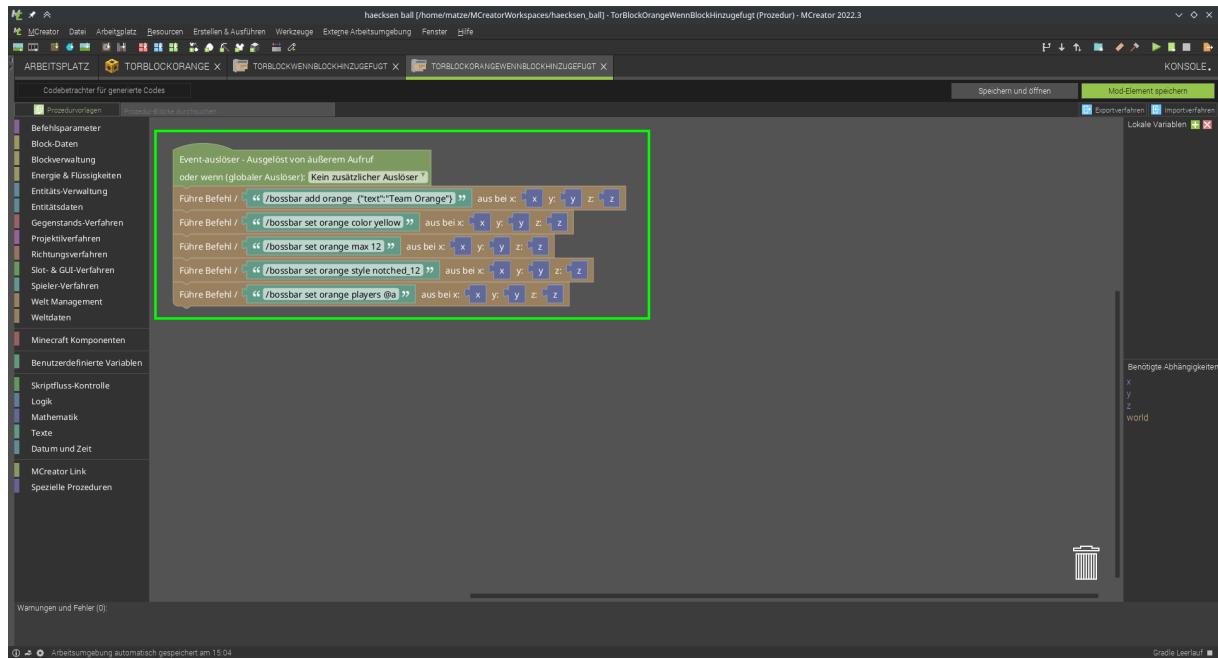
Der Name der Prozedur wird automatisch generiert, denn kannst Du so lassen. Jetzt sollten die 4 Minecraft-Befehle von oben ausgeführt werden, wenn das Tor gebaut wird. Also suchen wir den Baustein, der uns ermöglicht einen Minecraft-Befehl auszuführen.



**Abbildung 16:** Wie können Minecraft-Befehle ausgeführt werden

- Wähle links in den Kategorien *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 das dann aus wenn die Befehle von oben alle ausgeführt werden sollen:



**Abbildung 17:** Minecraft Befehle in MCreator für bossbar

Speichern nicht vergessen ;) Drücke dazu rechts oben auf **Mod Element speichern**

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

**“Mal Nachschauen:”**

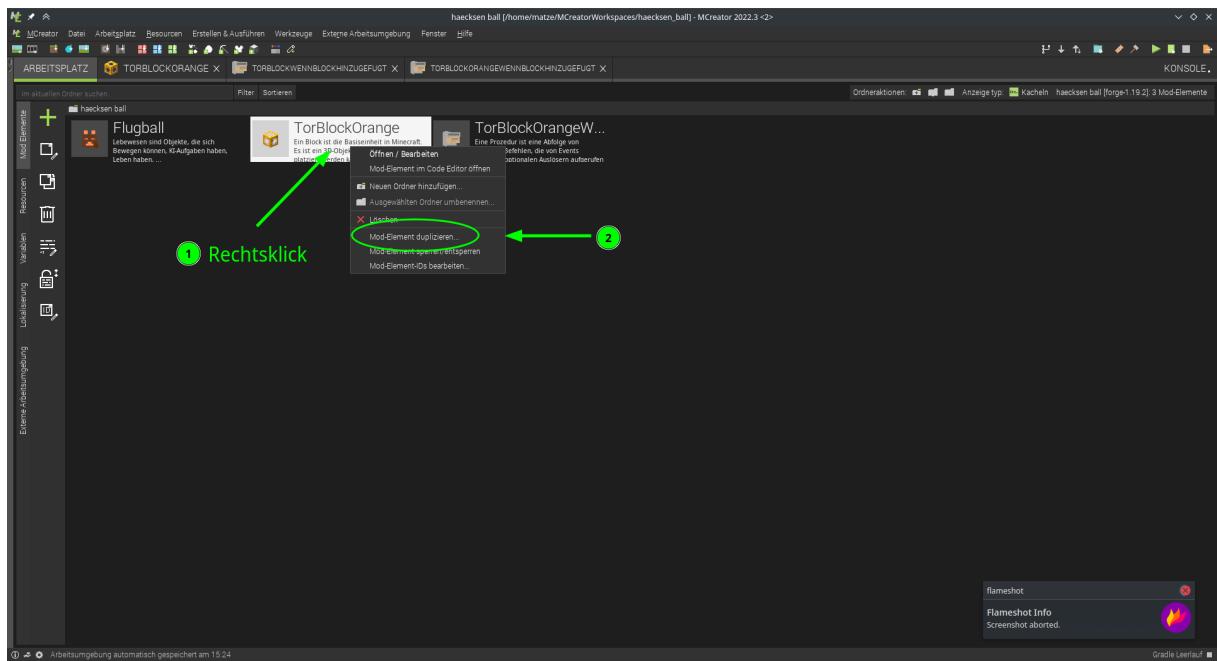
Jetzt sollte es so aussehen nachdem man einen orangenen Torblock gesetzt hat:



**Abbildung 18:** ingame bossbar orange

**Was ist mit der gegnerischen Manschafft?**

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:



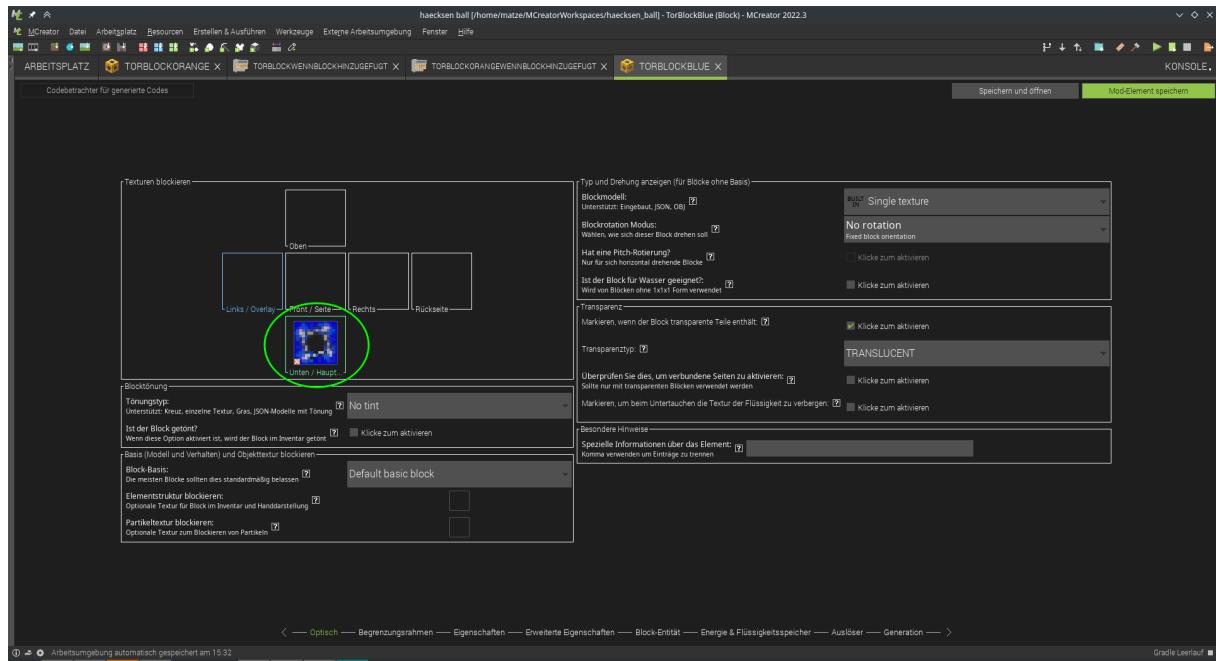
**Abbildung 19:** Arbeit sparen: so kannst Du bestehende Elemente duplizieren

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

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

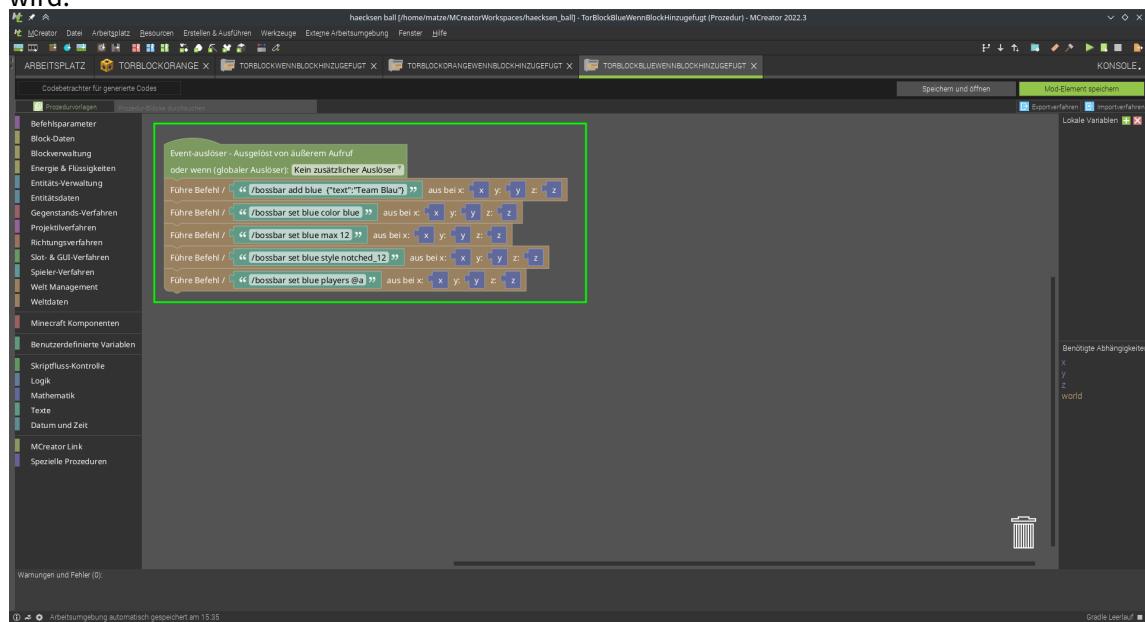
- Rechtsklick
- *Mod-Element duplizieren*
- Neuer Name *TorBlockBlueWennBlockHinzugefügt*

Und in beiden Elementen musst Du die IDs und Farbrelevanten Eigenschaften anpassen:



**Abbildung 20:** Tausche die Textur der Blockes aus

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



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



### “Geschafft!”

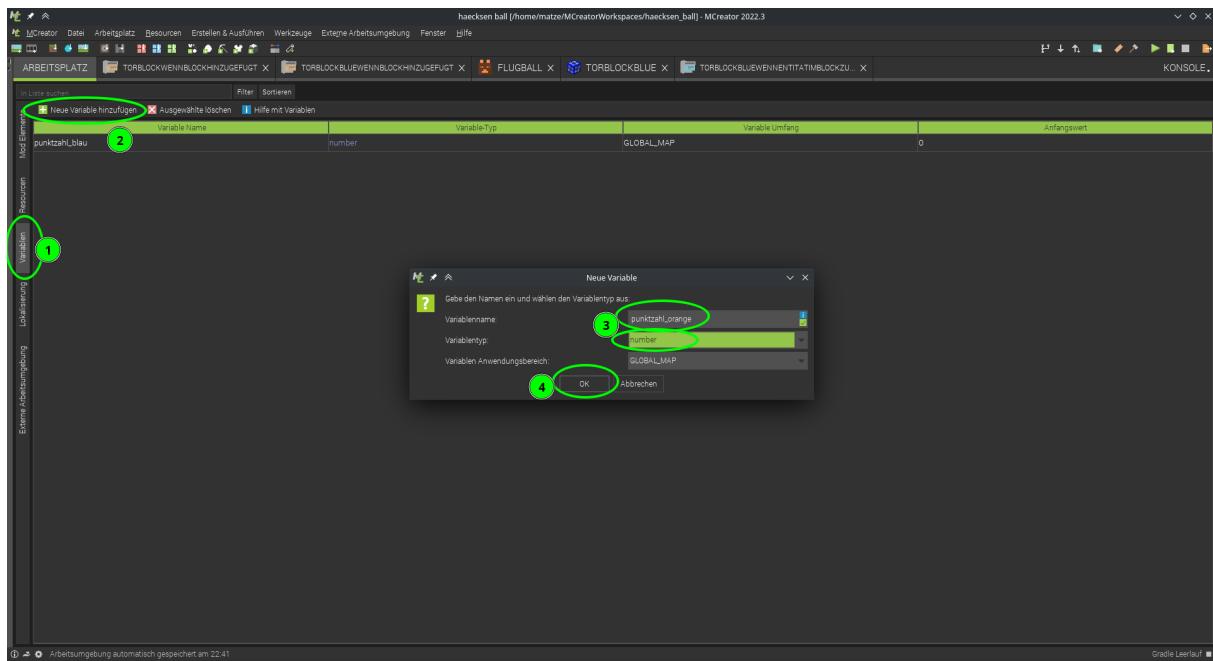
Wow - Du hast schon eine Menge geschafft: Bälle gibt's, Tore - und sogar die Tor-Anzeige!



## Level 3: Tore / Punkte zählen

### Punkte merken

Als erstes müssen wir uns die Punktzahl für jedes Team merken:



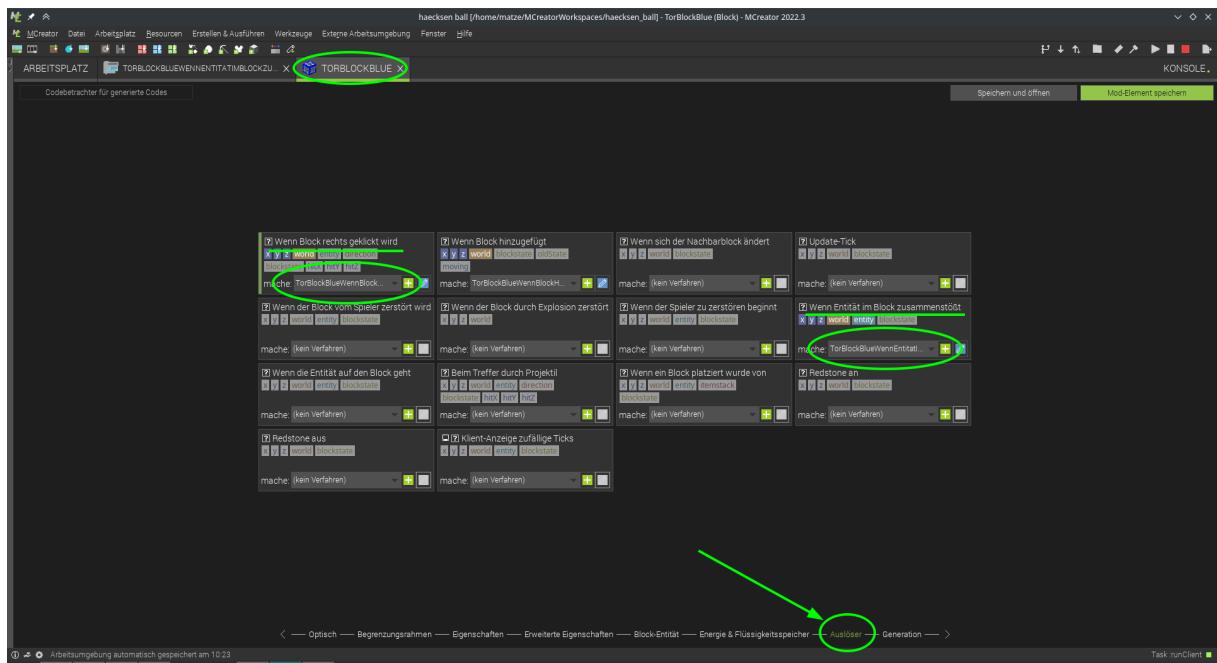
**Abbildung 21:** variable erstellen

dazu erstellen wir eine variable für jedes team:

1. In der seitlichen Leiste *Variablen* auswählen
2. Eine *Neue Variable hinzufügen*
3. *Variablenname* vergeben und typ *VariablenTyp* auf *number* setzen
4. mit *Ok* speichern
5. das gleiche nochmal für die andere Farbe

## Tore erkennen

Dazu brauchen wir einen Auslöser der erkennt wann ein Tor geschossen wurde.

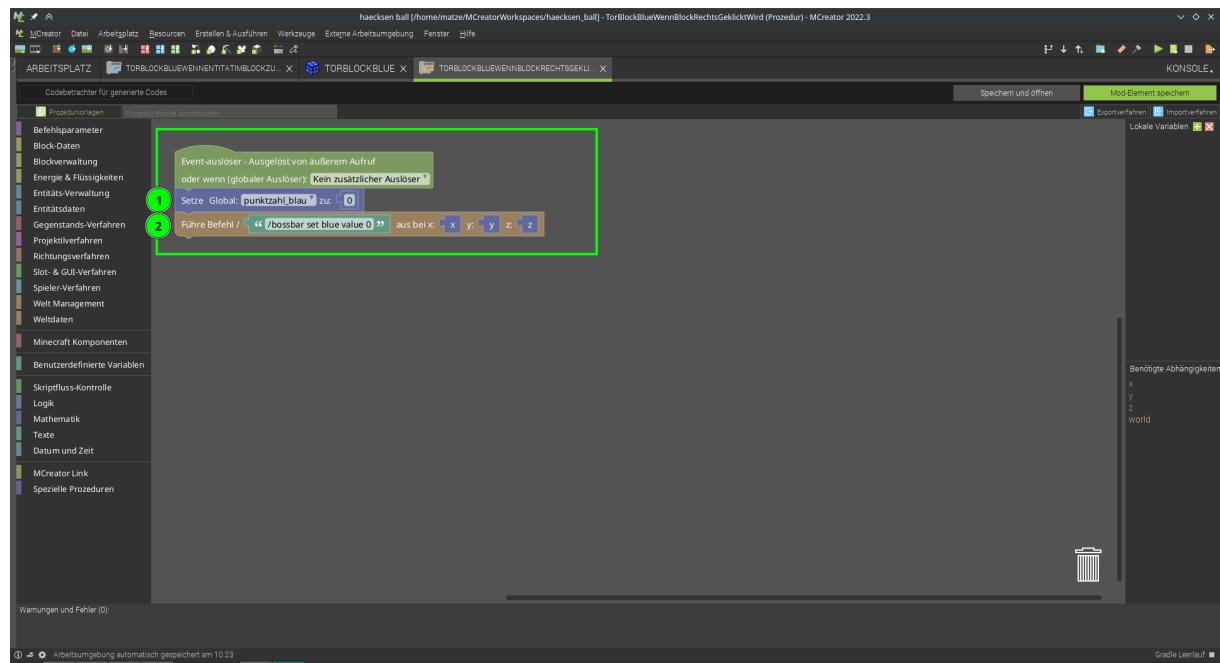


**Abbildung 22:** neue auslöser

Wir bearbeiten den Torblock und gehen zu den Auslösern. Wir fügen 2 Auslöser hinzu:

- einen *Wenn Entität im Block zusammenstößt*  
um Punkte hoch zu zählen und Ball zu entfernen
- einen *Wenn Block rechts geklickt wird*  
um Punkte zurück zu setzen.

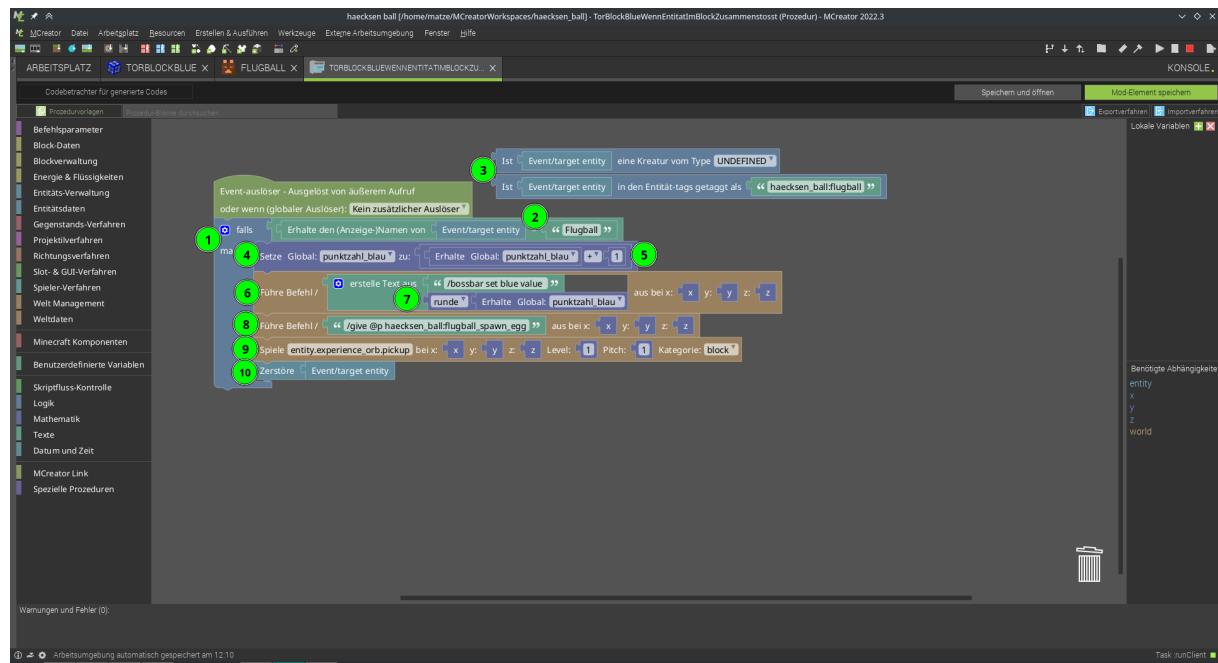
Die Prozedur für den Rechtsklick sieht so aus:



**Abbildung 23:** code rechtsklick

1. Setzt die globale variable zum Punkte Zählen auf 0 zurück
2. führt `/bossbar set blue value 0` in Minecraft aus um die Bossbar wieder auf 0 zu setzen.

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



**Abbildung 24:** code zusammenstoß entität

1. Weil wir nur wollen, dass ein Punkt nur dann verteilt wird, wenn der Ball das Tor berührt checken wir bei ob es sich bei der Entität um einen Ball handelt
2. dazu erhalten wir den Anzeige-Namen und prüfen ob der gleich *Flugball* (muss übereinstimmen, wie der Objektname bei [Bild und Ton von Ball erstellen](#))
3. die vom Hauptcodeblock abgelösten Anweisungen werden nicht ausgeführt und dienen nur als Hinweis was ich probiert hab: bei der (3) wollt ich auf diese 2 Wege testen ob es sich um einen Ball handelt, was leider nicht geklappt hat.
4. Erhöht die Punktzahl von Blau:
5.
  - dazu holt es sich die aktuelle *punktzahl\_blaau*
  - und fügt mit +
  - die Zahl 1 hinzu
  - und Speichert das Ergebnis mit *Setze Global: punktzahl* zu wieder in der Punktzahl für blau ab.
6. Wir führen den Minecraft Befehl für die Bossbar aus
7. dazu kombinieren wir den text `/bossbar set blue value` (!Achtung! Leerzeichen am Ende wichtig, dass es funktioniert) mit der *punktzahl\_blaau* und runden diese, weil es sich sonst um eine Kommazahl handelt. (zwar immer ,0 aber Minecraft kennt hier keine Kommazahl, deshalb machen wir die so weg)

8. wir geben dem nächsten Spieler (@p) ein neues Spawn-Ei

`/give @p haecksen_ball:flugball_spawn_egg`

9. mit diesem Befehl können wir einen Ton abspielen lassen, wenn ein Tor geschossen wurde. Hier z.B. der gleiche Ton, wenn man XP aufsammelt: `entity.experience_orb.pickup`

10. am Schluss wollen wir den Ball natürlich los werden:

- dazu am besten den Befehl `Zerstöre` verwenden
- Achtung: die Minecraft Befehle `kill @e[...]`, `tp @e[...] ~ -500 ~` und MCreator `Lösche` gehen nicht aus verschiedenen Gründen.

11. Speichern nicht vergessen ;)

Ich hab absichtlich dem Blauen Team ein Punkt gegeben, vtl ein Negativ-Punkt, damit man sehr einfach um ein weiteres Team ergänzen kann.



### Punkte abziehen

Wenn ihr wollt könnt ihr am Anfang auch 12 Punkte vergeben und mit jedem Tor in den blauen Torblock 1 Punkt von Blau abziehen. Spielt gerne mit den Zahlen rum ;)

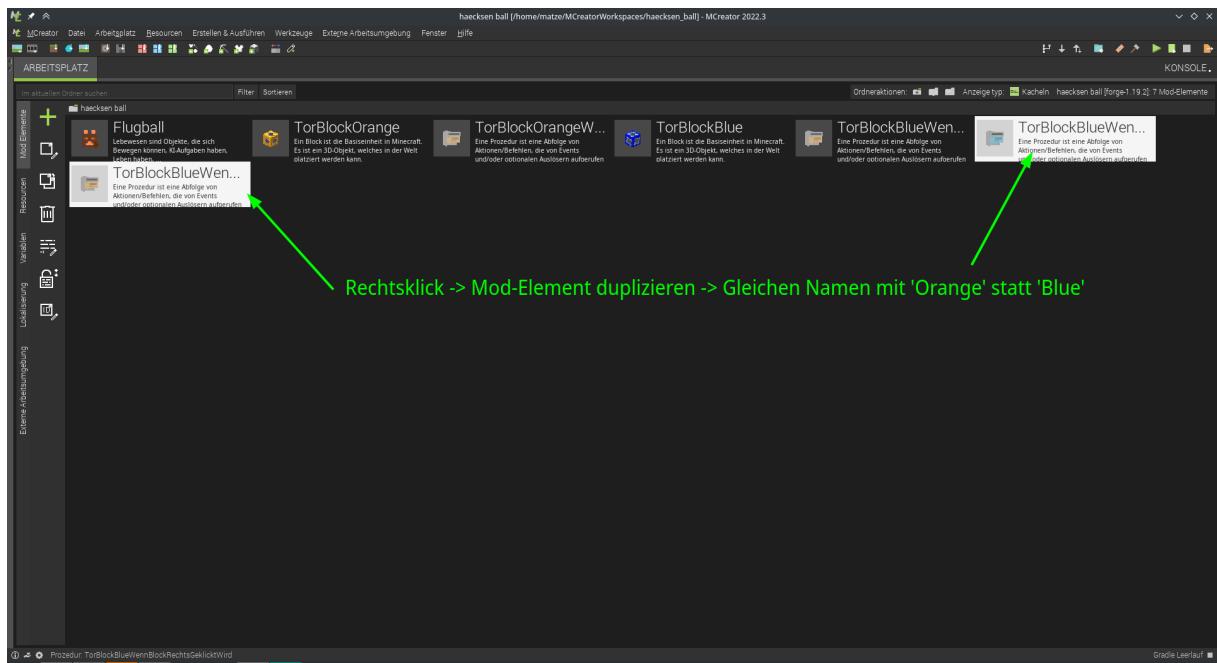


### Erfolg

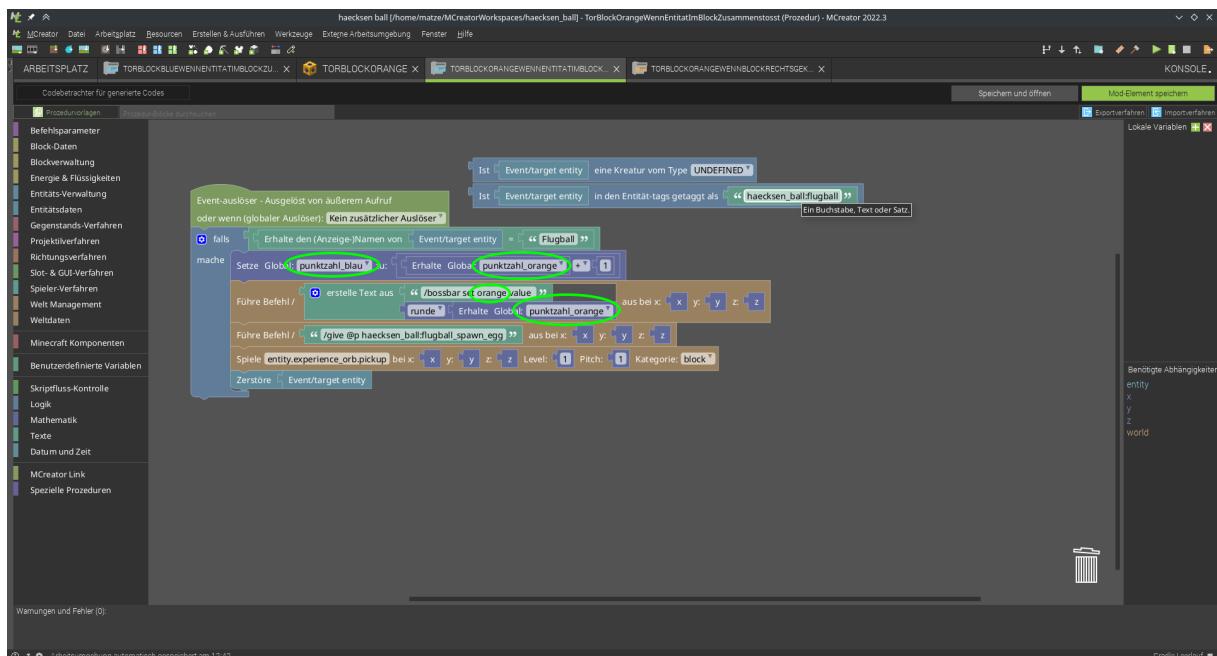
Jetzt kanns im Spiel getestet werden, und wenn ein Ball in einen blauen Torblock fliegt, dann sollte ein Ton kommen, die Punktzahl in der Bossbar erhöht werden, der Ball verschwinden und man bekommt ein neues Spawn-Ei.

### Jetzt nochmal für Orange

Die Variable `punktezahl_orange` für Orange haben wir ja schon erstellt. jetzt noch die Prozeduren kopieren:

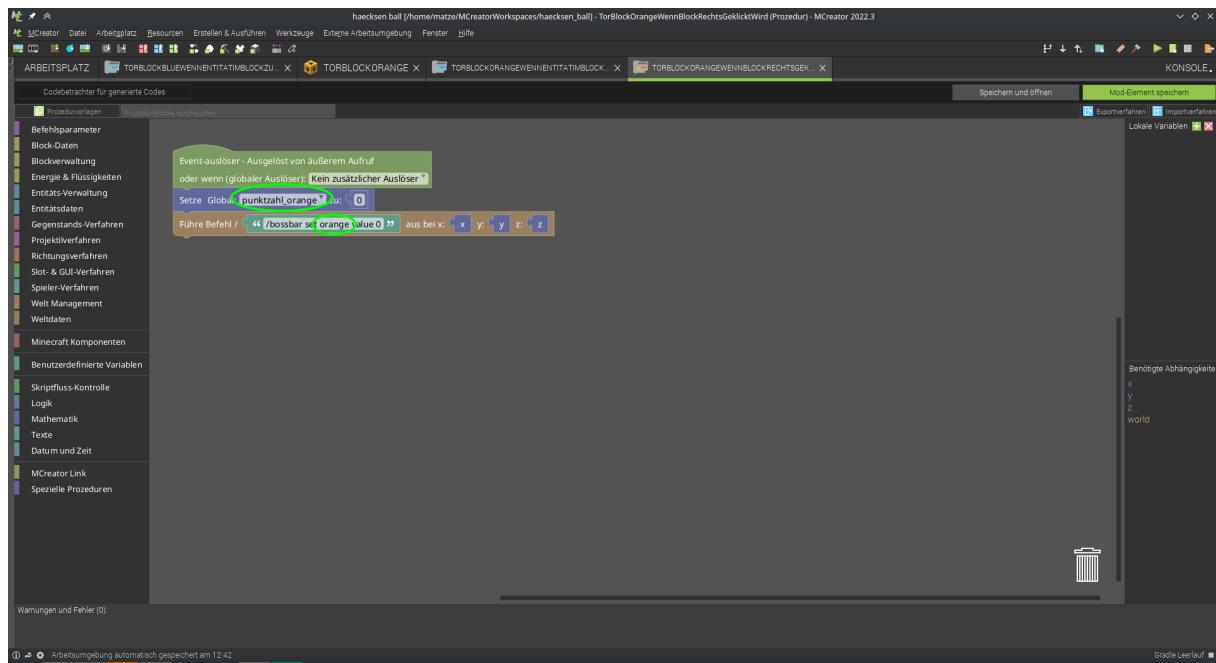
**Abbildung 25:** element duplizieren

Jetzt natürlich noch alle dinge von Blau zu Orange ändern in den beiden Codedateien:

**Abbildung 26:** code zusammenstoß orange

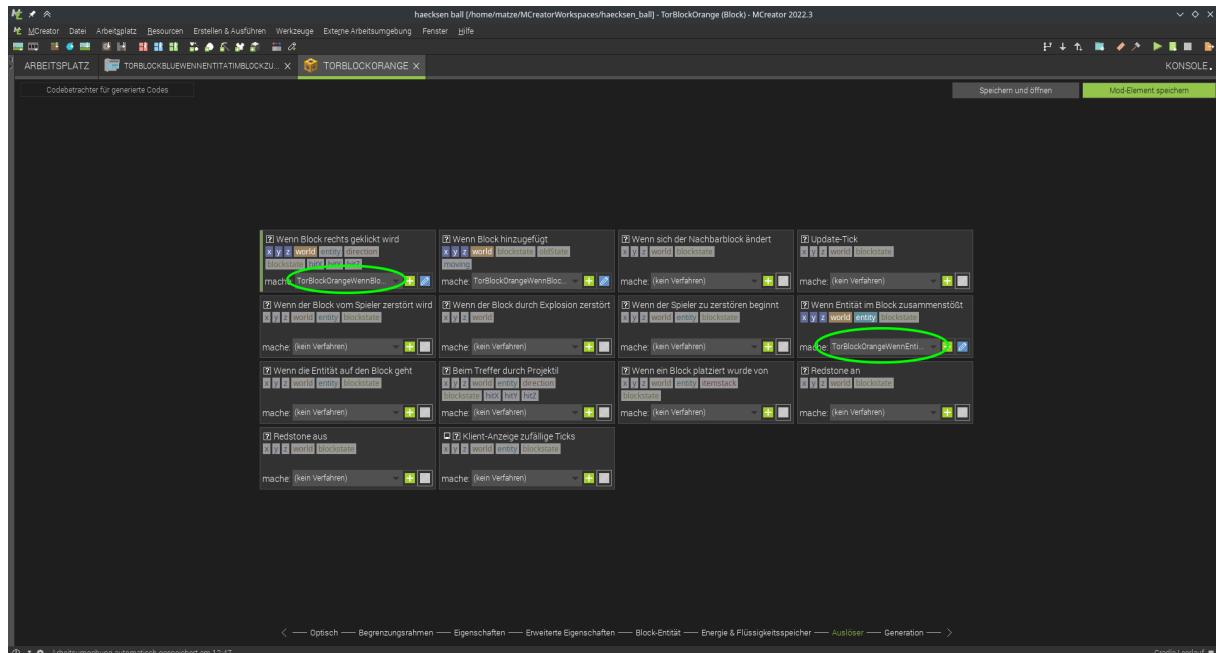
MCreator: Erstelle deine eigene Mod HaecksenBall!

2022-12-22



**Abbildung 27:** code rechtsklick orange

Und nicht vergessen!!: Auslöser auch anpassen



**Abbildung 28:** Auslöser auswählen

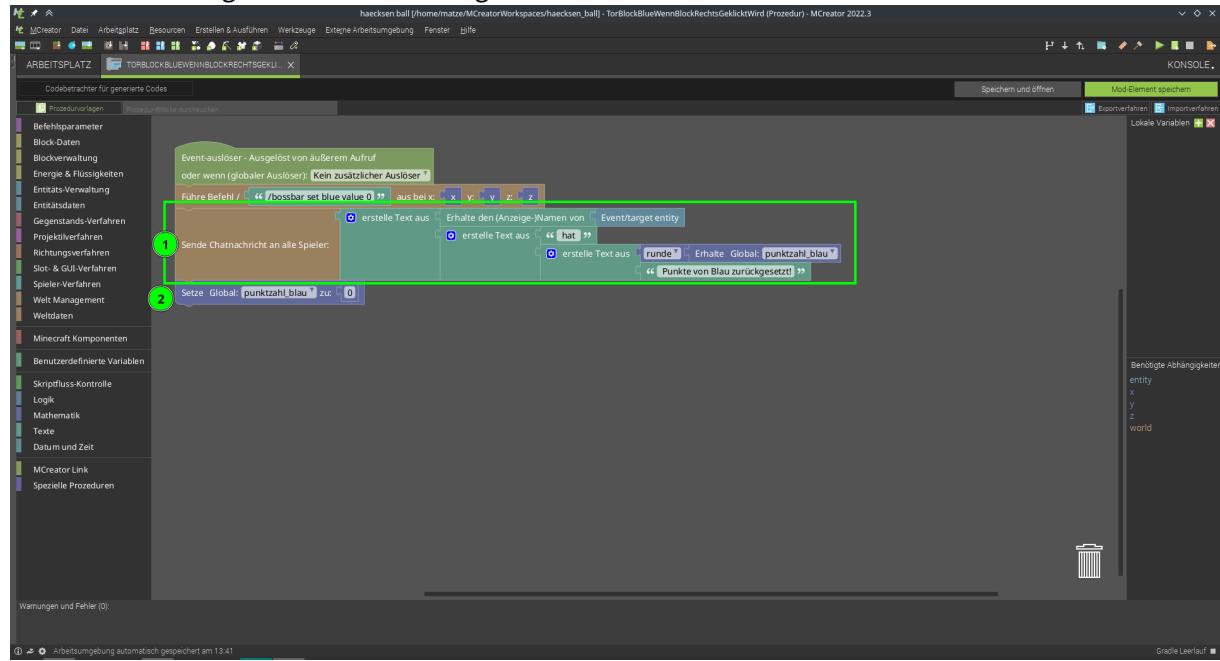


**“Funktioniert?”**

Und nochmal testen ;)

## Level 5: Erweiterte Punkteanzeige / Punkte zurücksetzen

Weil ja noch jede\*r dem gegnerischen Team einfach Punkte klauen könnte ohne dass mans merkt können wir der logik noch was hinzufügen:



1. Wir setzen aus vielen Bauteilen die Nachricht für den Chat zusammen

- Spielername mit *Erhalte den (Anzeige-)Name von > Event/target entity*
- ” hat ” (Leerzeichen nicht vergessen)
- *runde ]> Erhalte Global punktzahl\_blue*
- ” Punkte von Blau zurückgesetzt!”

2. Aufpassen, dass die Punkte erst danach auf 0 gesetzt werden, ansonsten wird im Chat immer 0 angezeigt.

Das gleiche natürlich auch noch in dem Code für Orange, damit es für beide Farben passt.



### **“Aufgepasst! Platz lassen”**

Wenn wir mehrere Textbausteine aneinander hängen, müssen wir drauf achten wo die Leerzeichen stehen, sonst wird der Text schwer lesbar ;) z.B. vor und nach *hat* ist jeweils ein Leerzeichen.



### **“Testen schadet nie ;)”**

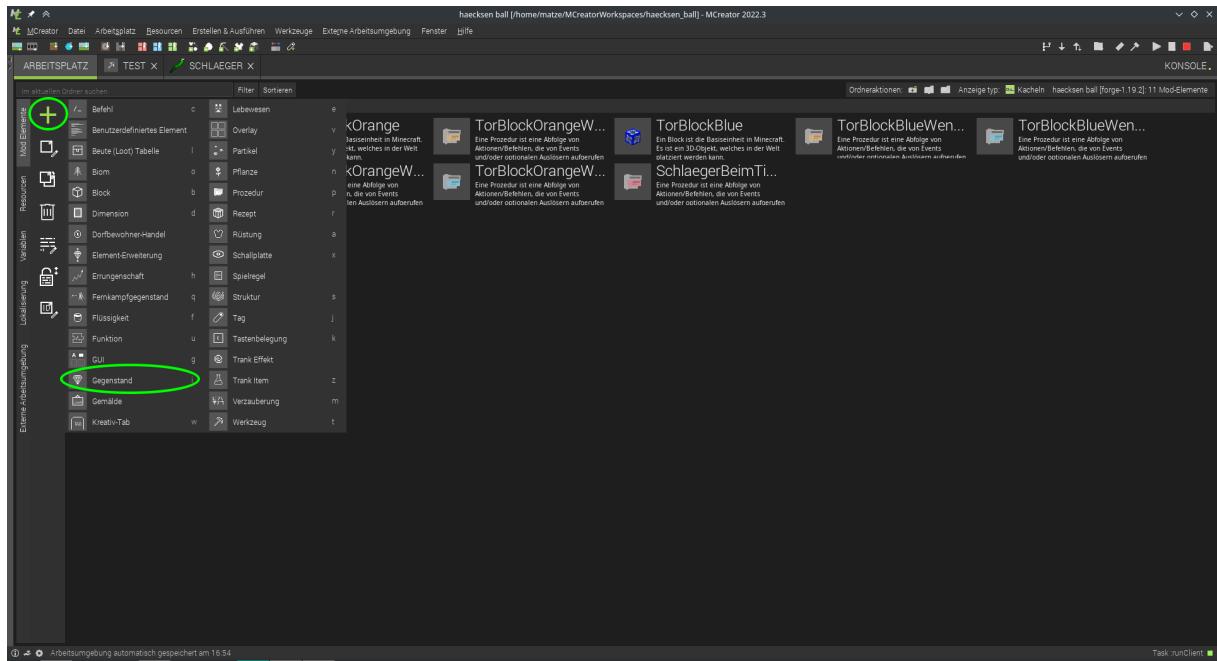
Klappt alles wie erwartet?

Für beide  
Farben?

## **Level 5: Die Haecksenball-Schläger**

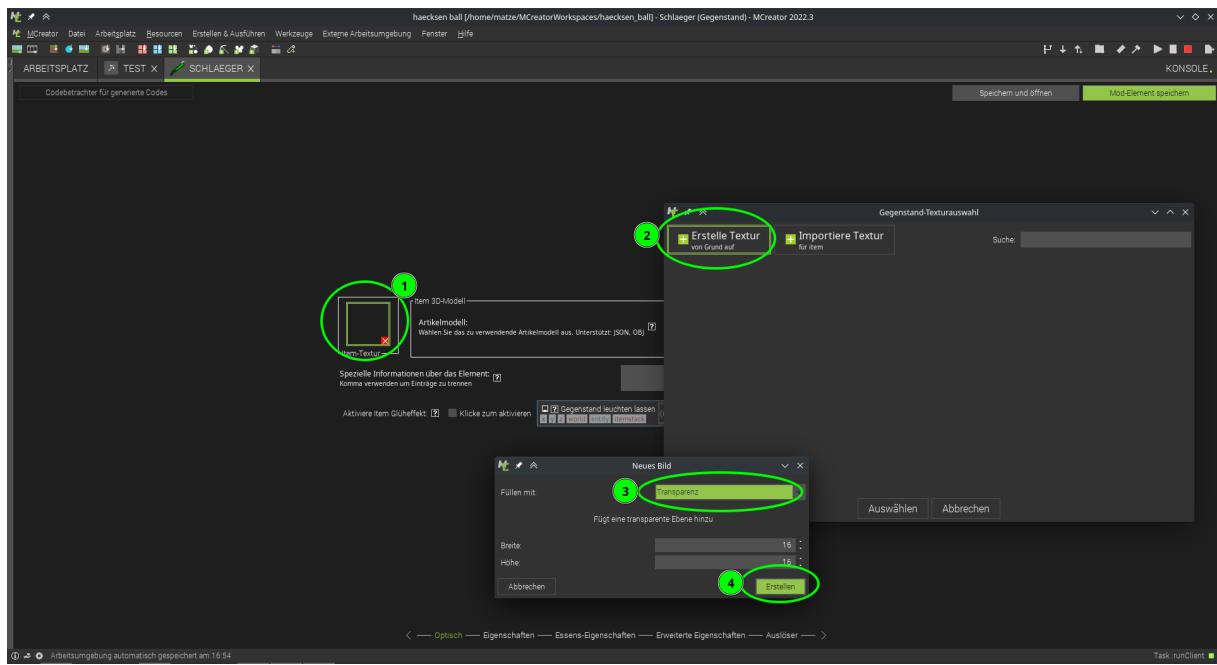
Wie beim Eishockey, Golf und Tennis macht das Spiel mehr spaß mit einem richtigen Schläger für den Ball ;)

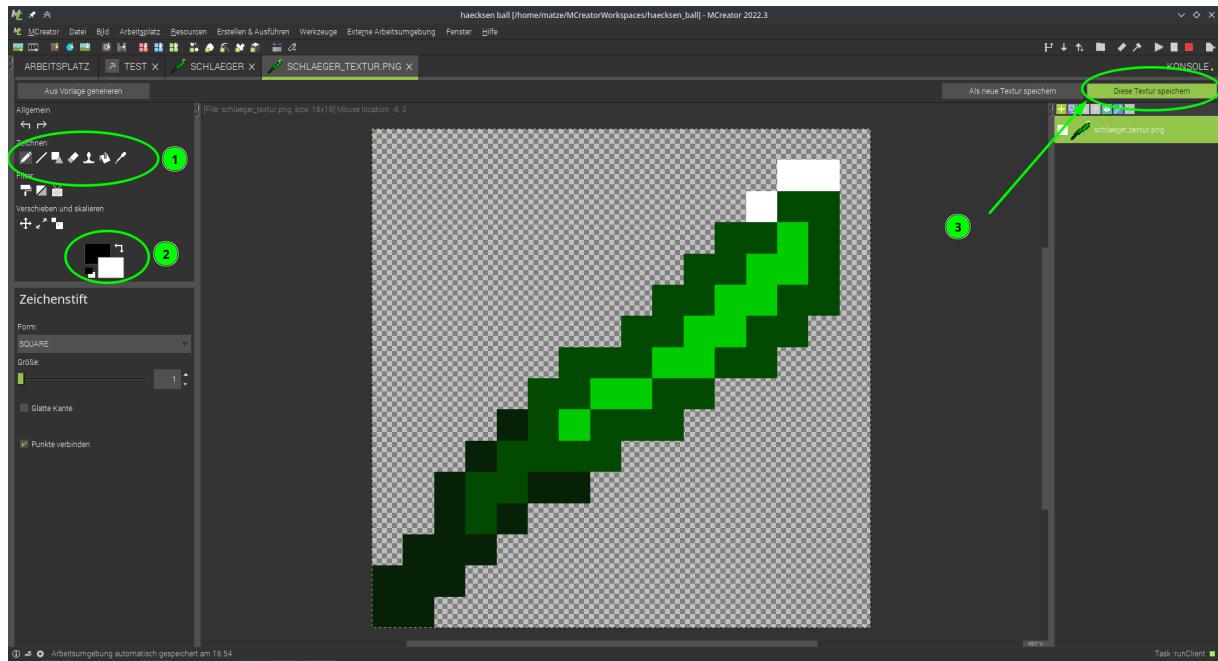
Wir fügen ein Gegenstand hinzu:



**Abbildung 29:** Gegenstand erstellen

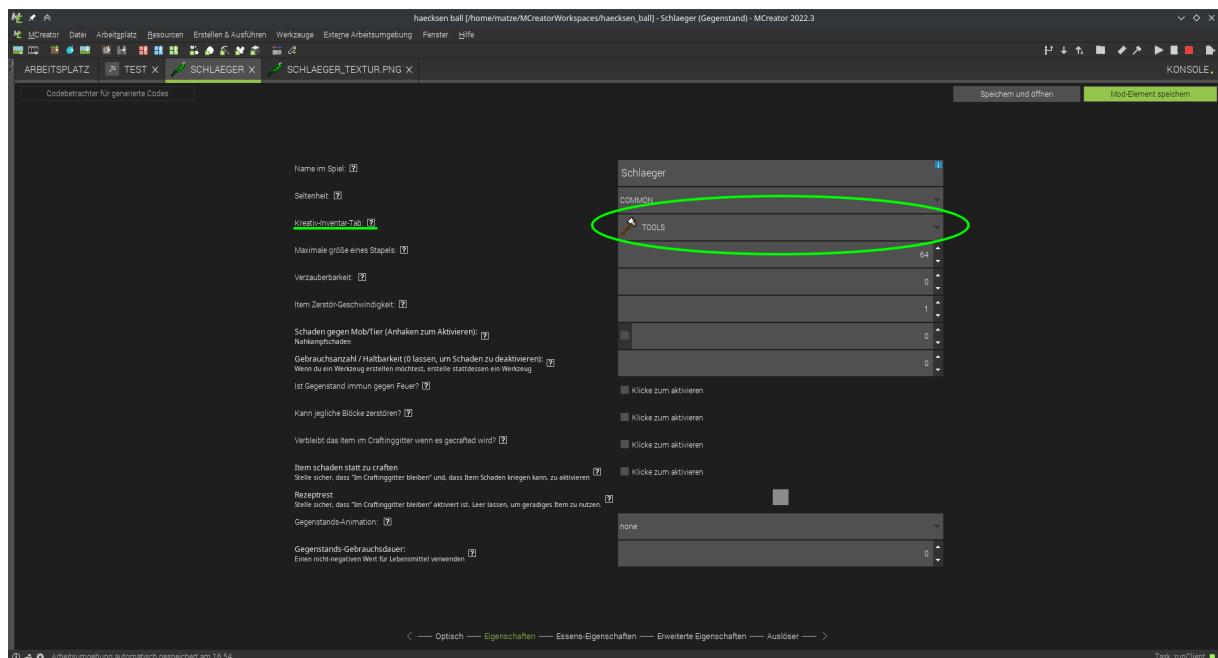
Name vergeben z.B. Schlaeger





1. Zeichentool auswählen
2. Farbe auswählen
3. Speichern, Als **Gegenstandtextur!**

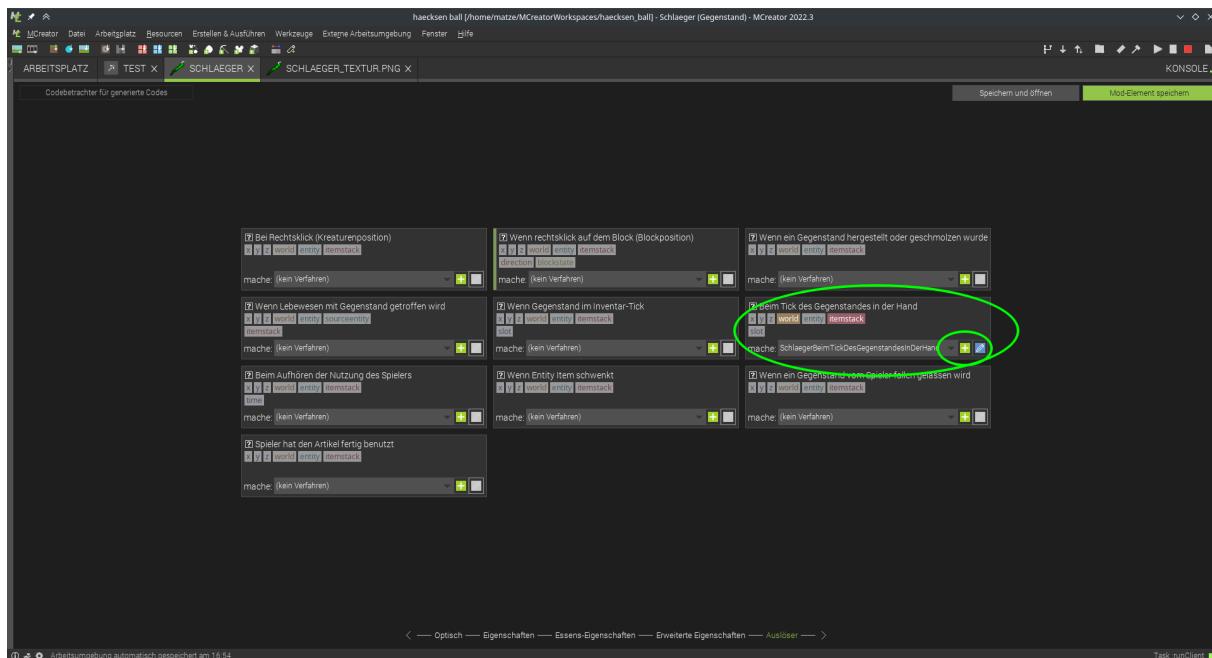
Dann beim Gegenstand die gerade erstellte Textur nochmal auswählen.



**Abbildung 30:** Gegenstand Eigenschaften

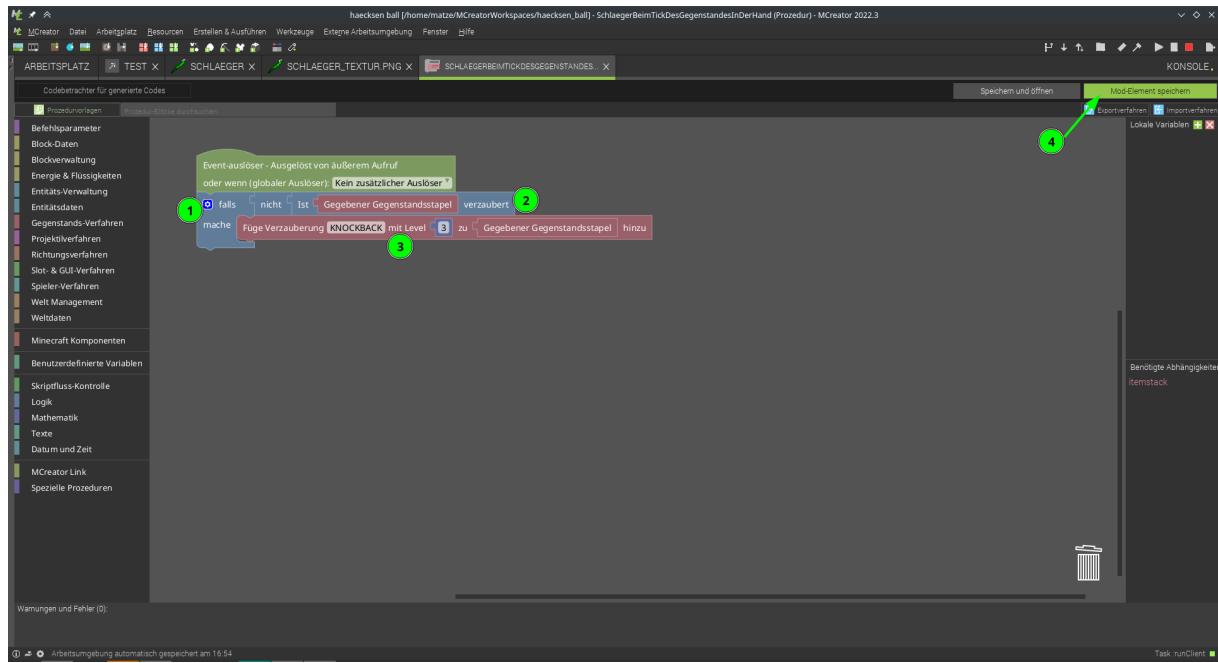
Als Inventar Tab hab ich *Tools* ausgewählt, dass es bei den anderen Werkzeugen auftaucht im Inventar.

Weils hier keine Einstellung dafür gibt wie viel Rückstoß der Schläger haben soll machen wir dafür wieder eine Prozedur die immer ausgeführt wird wenn der Schläger in der Hand gehalten wird:



### **Abbildung 31:** Auslöser

Der code, dass der Schläger automatisch verzaubert wird:



1. Testen ob der Gegenstand noch nicht verzaubert ist.
2. Dazu verneinen wir die Antwort auf die Frage *Ist Gegebener Gegenstandsstack verzaubert*
3. und dann fügen wir die Verzauberung *KNOCKBACK* auf deutsch Rückstoß mit dem level 3 hinzu.



Kleiner Test im Spiel:

Schläger in der Hand wird sofort verzaubert :D

### Kleiner Tipp

man kann auch mit einem Bogen (vtl mit Schlag verzaubert) sehr gut Spielen wenn der Ball weiter weg ist ;)

## Level 6: Das Spielfeld: Die Arena!

Superflachwelt erstellen: TODO: Anleitung dafür nötig?

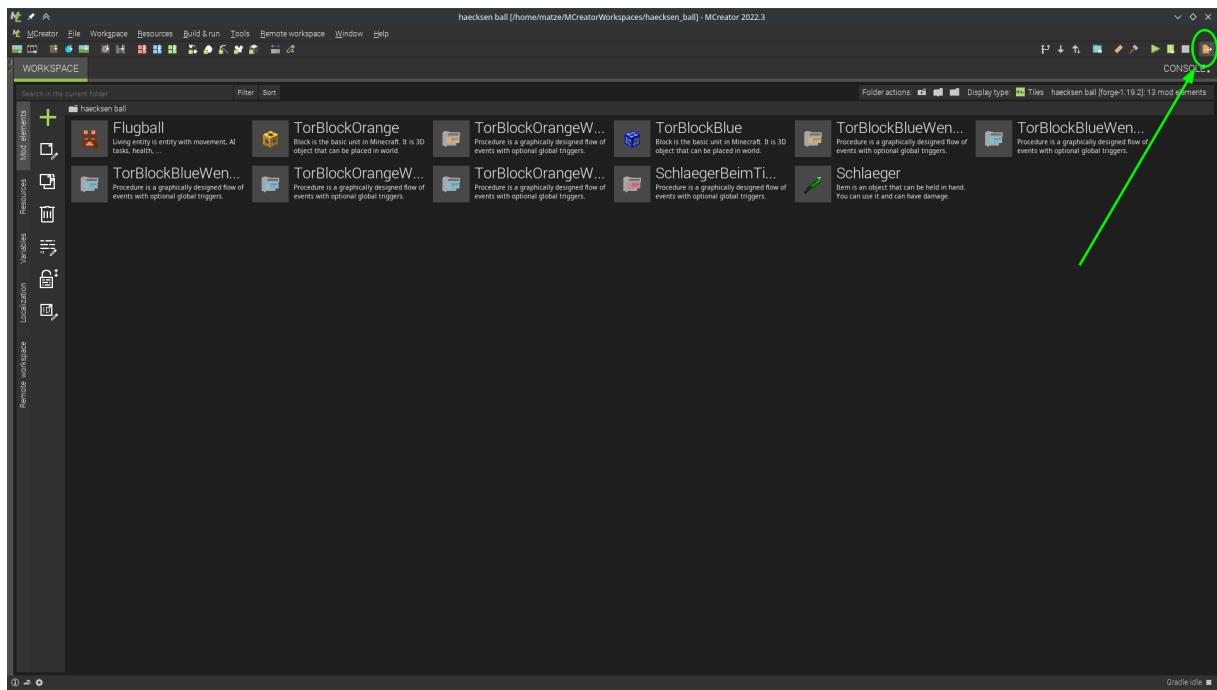
So hab ich mir eine Arena gebaut:



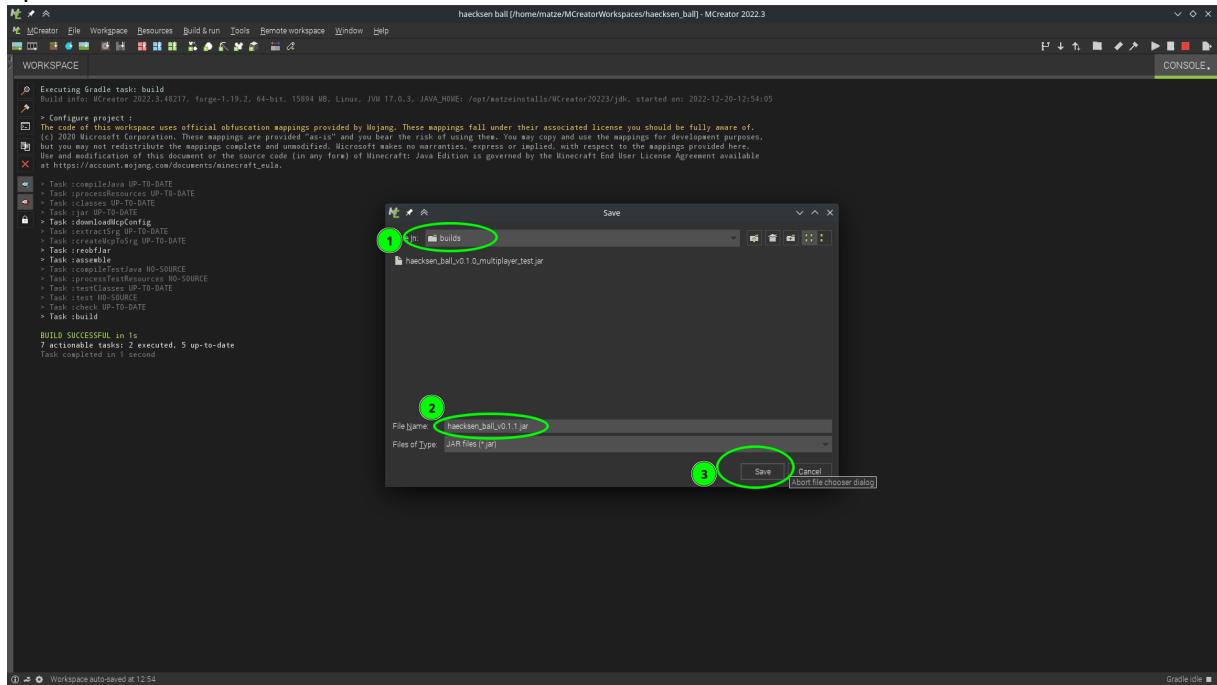
Minecraft Wiki: [/fill](#) sehr hilfreich

## Level 7: Exportieren & Veröffentlichen der Mod

Um die Mod in Minecraft ohne MCreator zu spielen brauchen wir eine [.jar](#) Datei.  
Diese können wir uns generieren lassen:



nachdem click auf das exportieren symbol wird dier .jar generiert und MCreator fragt nach einem Speicherort:



diese .jar Datei kann man wie jede andere mod installiert werden.  
mod installiren