
MCreator: Erstelle deine eigene Mod HaecksenBall!

KidsLab.de gGmbH - MatzE, KingBBQ

2022-12-22

Übersicht

Eigene Minecraft-Mod erstellen: Haecksen-Ball!	4
Einführung	4
Was brauche ich dafür?	4
Vorbereitung	5
Online-Version mit Links zu weiteren zugehörigen Dokumenten	5
Source-Code und weitere Dateien	5
Einführung in MCreator & Unser Projekt	5
Unser Spiel: Haecksen-Ball	5
Mods, MCreator, Java - eine kleine Einführung	6
“Wusstest Du schon...”	6
Was ist MCreator?	6
Mods?	6
“Lust mal ne Mod zu installieren?”	7
Vorbereitung: Neues Projekt in MCreator anlegen	7
“Dauert ewig?”	9
Übersicht über die IDE	9
“ID.... was?”	9
Minecraft Wiki	11
Level 1: Der Ball für unser Spiel	11
“Was passiert da im Hintergrund?”	15
“Geschafft: Der Ball fliegt!”	17
Verhalten anpassen	17
“Ball fertig - yeah!”	20
Level 2: Tor! Oder: da muss der Ball rein	20
“Probier es aus!”	23
Anzeige des Spielstandes - wieviele Tore wurden geschossen?	23
“Probiere es aus!”	24
Wie werden die Tore gezählt? Auslöser!	24
“Mal Nachschauen:”	28
Was ist mit der gegnerischen Mannschaft?	28
“Geschafft!”	31

Level 3: Tore / Punkte zählen	31
“Variablen - eine kurze Einführung”	32
Punkte merken	32
Tore erkennen	33
Prozedur 1: Punkte zurücksetzen	35
Prozedur 2: Tore zählen	35
“Blau bekommt einen Punkt?”	37
“Alternative: Punkte abziehen”	37
“Geschafft! Tore zählen - ich habe fertig!”	38
Jetzt nochmal für Orange	38
“Funktioniert?”	40
Level 4: Erweiterte Punkte-Anzeige / Punkte zurücksetzen	40
“Aufgepasst! Platz lassen”	41
“Testen schadet nie ;)”	42
Level 5: Die Haecksenball-Schläger	42
Erstelle einen neuen Gegenstand	42
“Umlaute und Programmieren”	43
Textur für den Schläger pixeln	44
Eigenschaften des Schlägers	45
Rückstoß	45
“Ausprobieren!”	47
“Alternative Schläger”	48
Level 6: Das Spielfeld: Die Arena!	48
Beispiel - so sieht unsere Arena aus	49
“Tipp: Schneller bauen”	49
Level 7: Exportieren & Installieren der Mod	49
JAR-Datei exportieren	49
Mod Installieren	51
“Mods & Server”	51



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

Geräte:	Computer pro Teilnehmer*in & Mentor*in, Beamer
Internet:	MCreator braucht Internet
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>)
- 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>

Source-Code und weitere Dateien

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

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 Zaubern und Hexen gespielt wird. Um das ganze noch perfekt in den Hacker-Kontext einzubetten, haben wir es [Haecksen](#)-Ball genannt: eine Haeckse ist ein weiblicher Hacker :)

Mods, MCcreator, 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 MCcreator?

"MCcreator 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

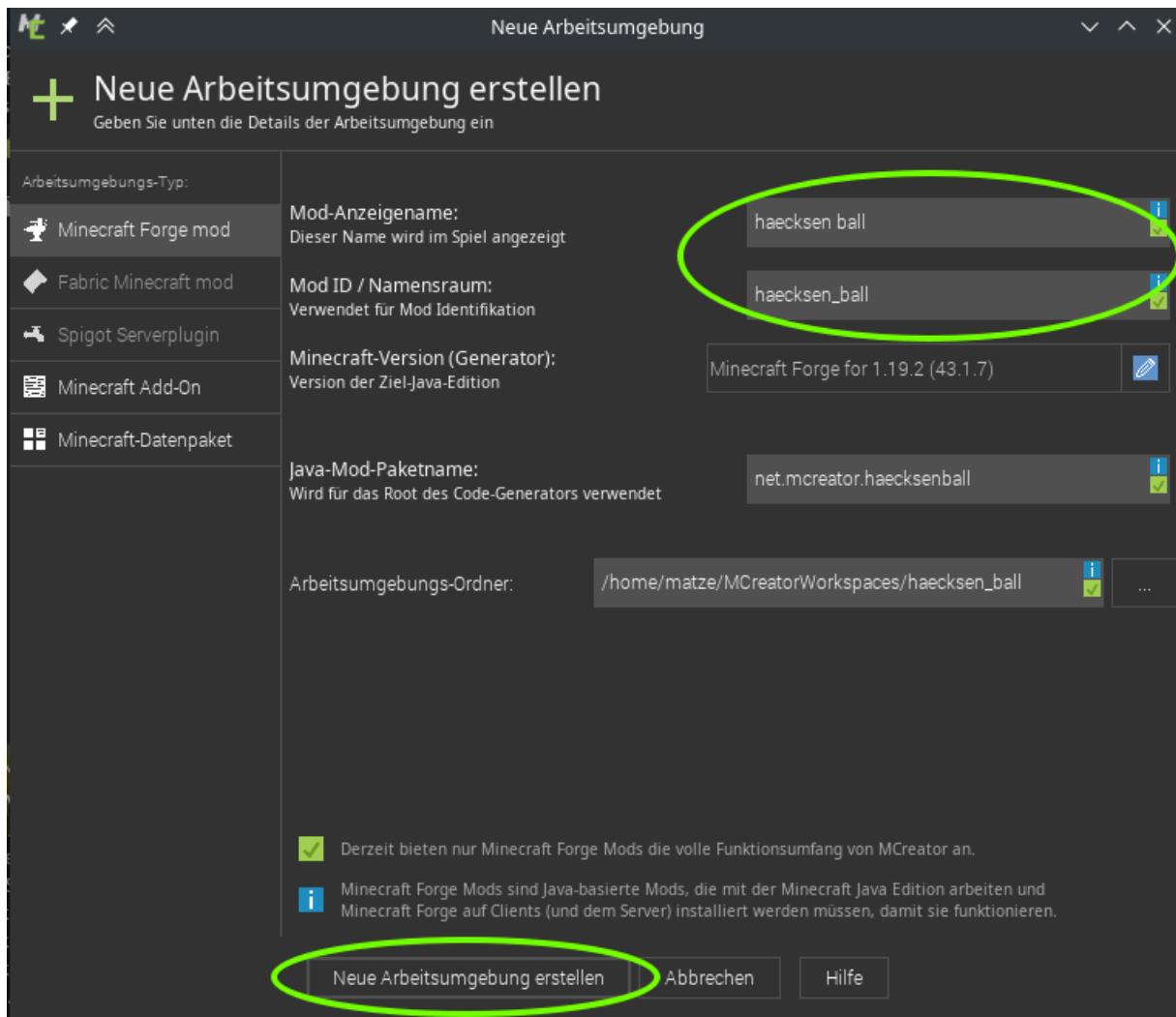


Abbildung 3: Neues Projekt anlegen - 2

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 :)

**“Dauert ewig?”**

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 Entwicklungsumgebung. 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:

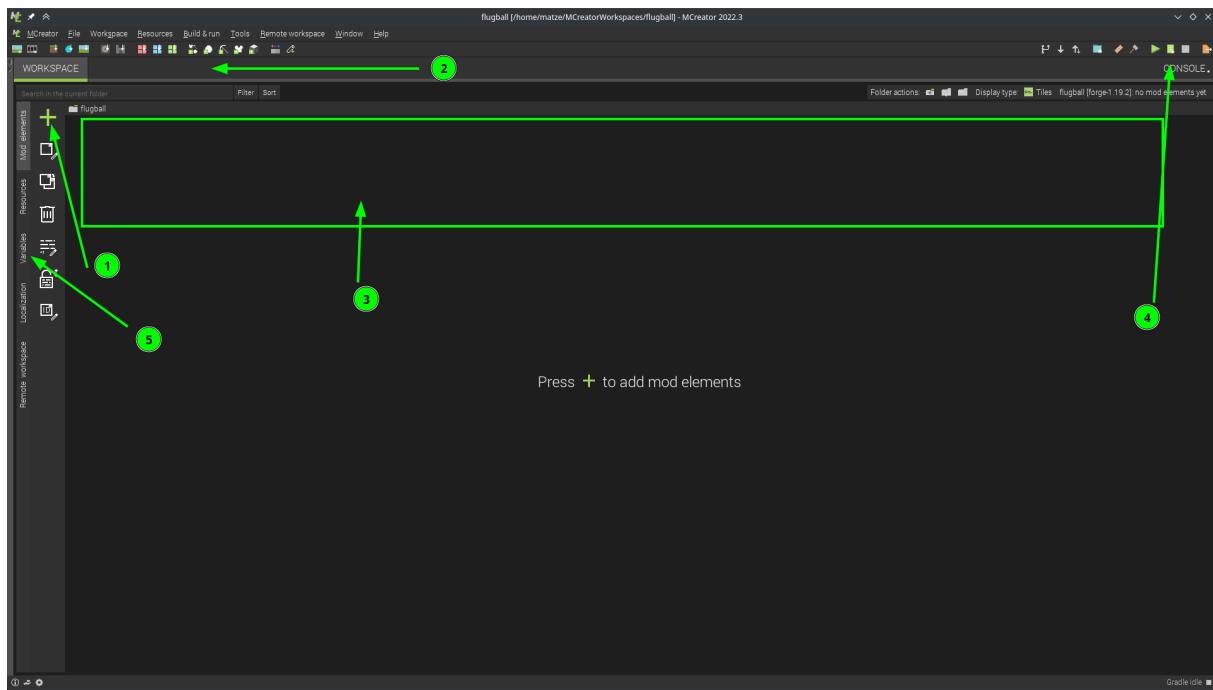


Abbildung 4: IDE Übersicht: das sind die wichtigsten Elemente

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 ein **Lebewesen** als Ball entschieden. Warum gleich noch mehr...

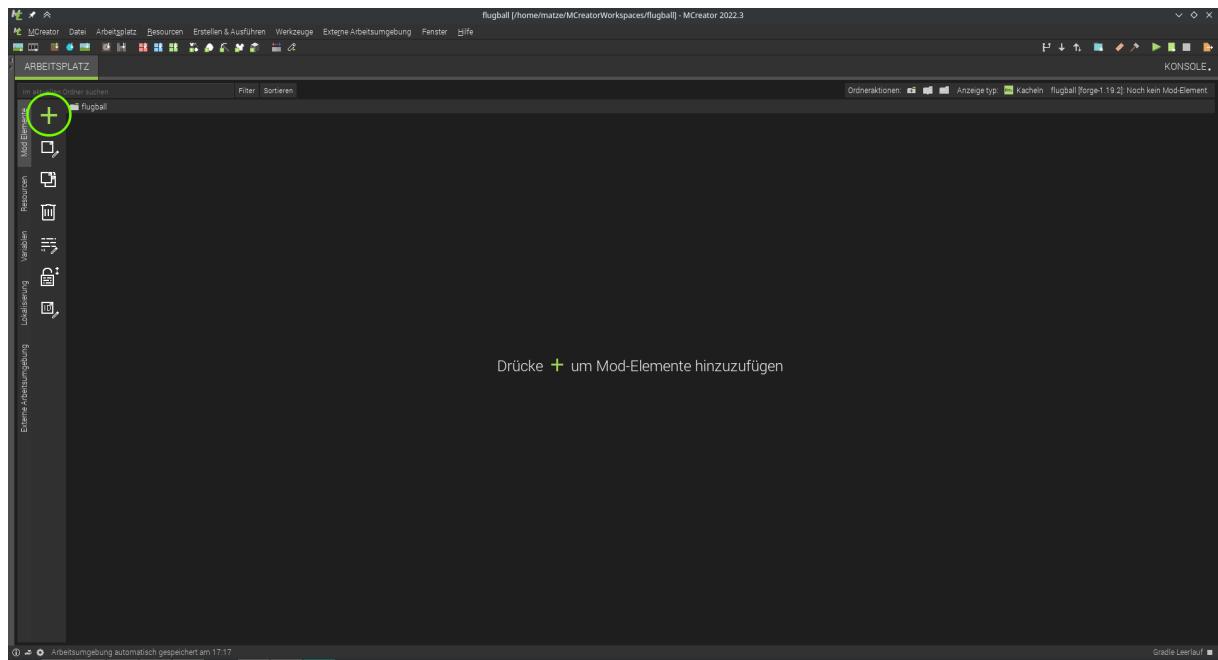


Abbildung 5: Starte die IDE und erstelle neues Objekt

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

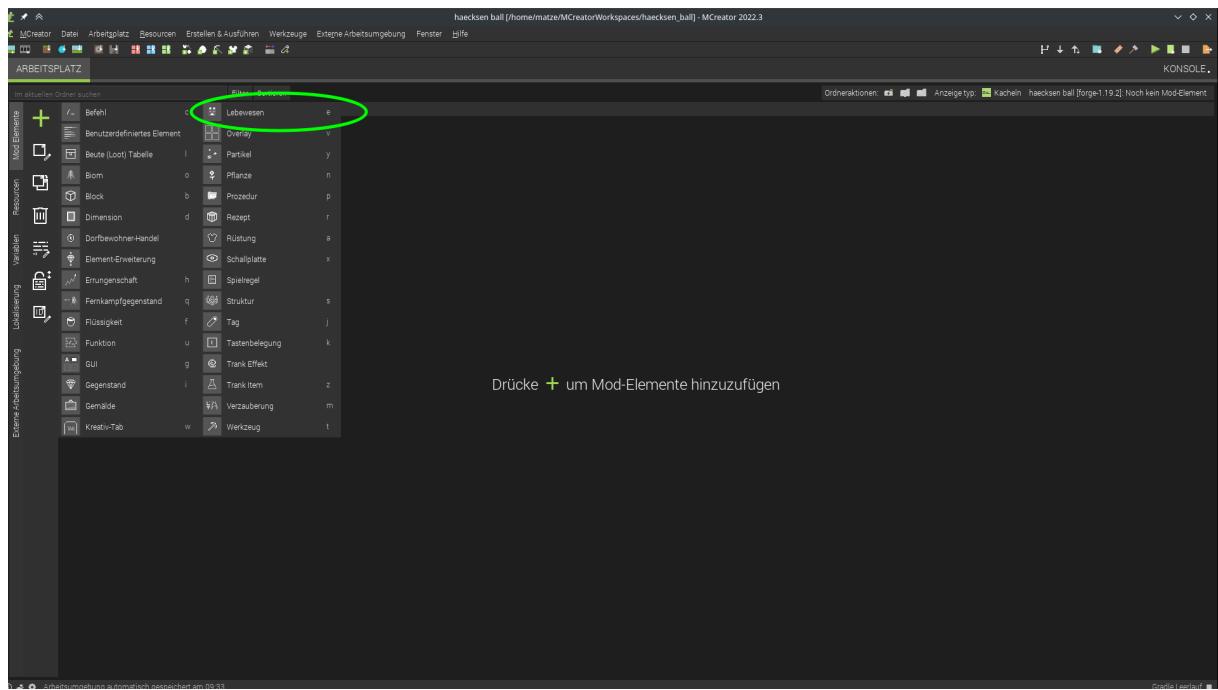


Abbildung 6: Wähle “Lebewesen”

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:

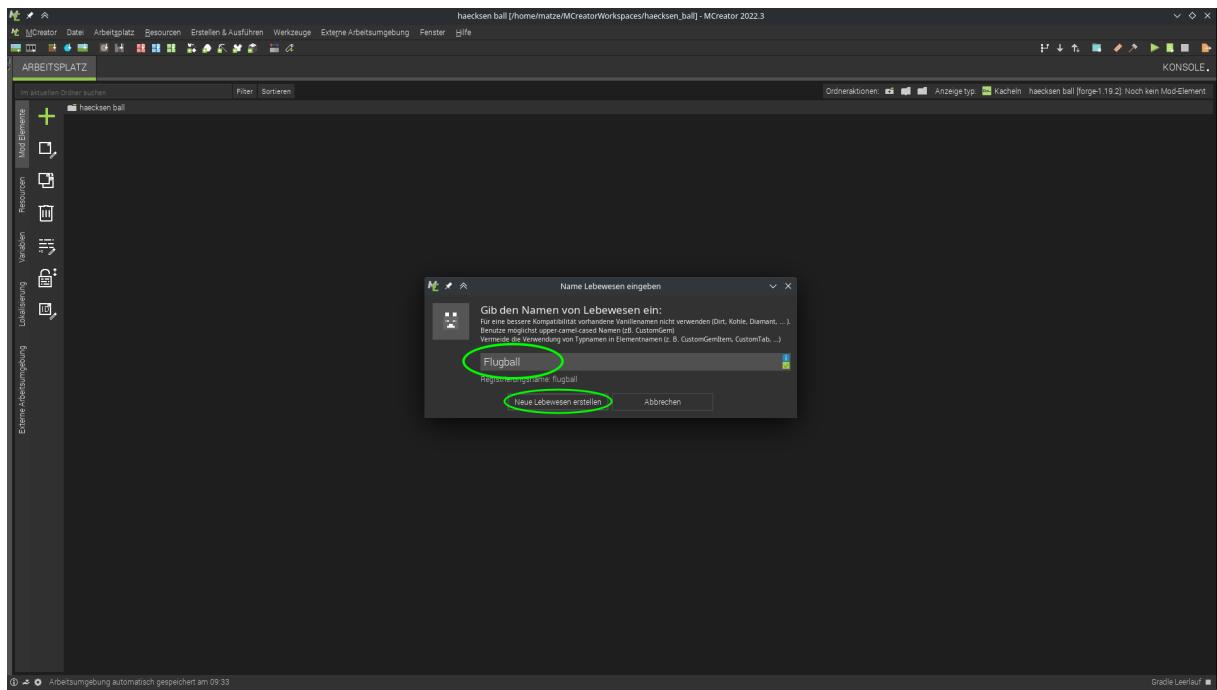


Abbildung 7: Name des Lebewesens: "Flugball"

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:

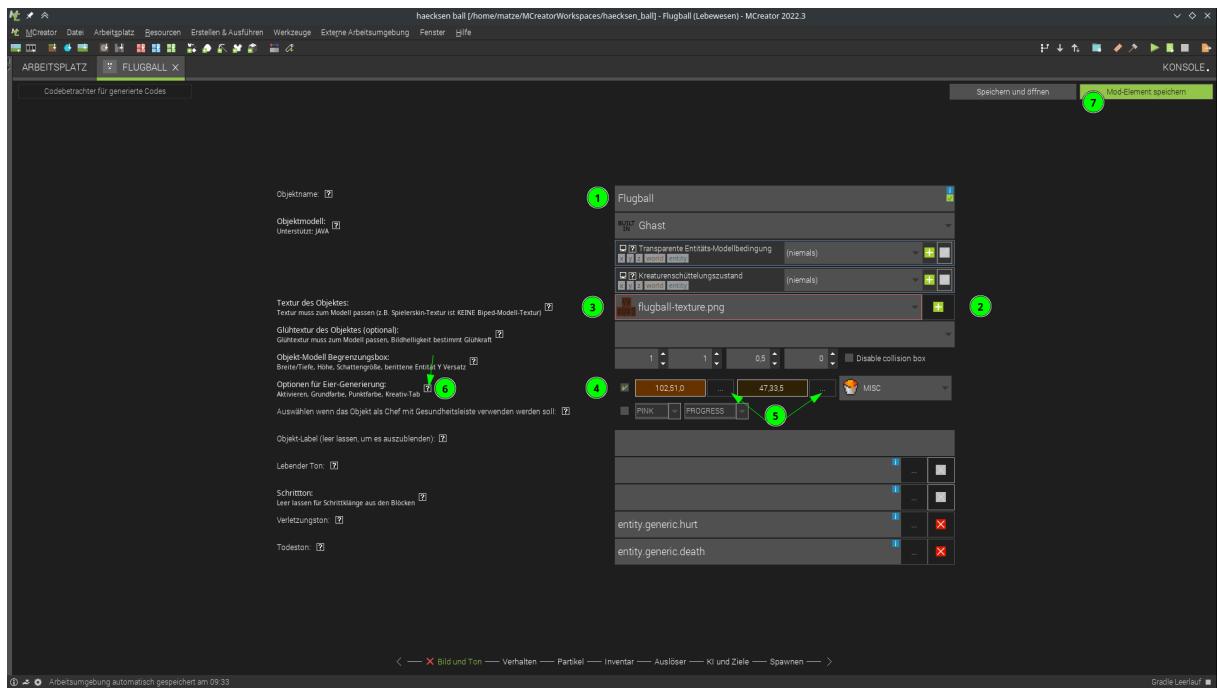


Abbildung 8: Lebewesen - Bild und Ton anpassen

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:

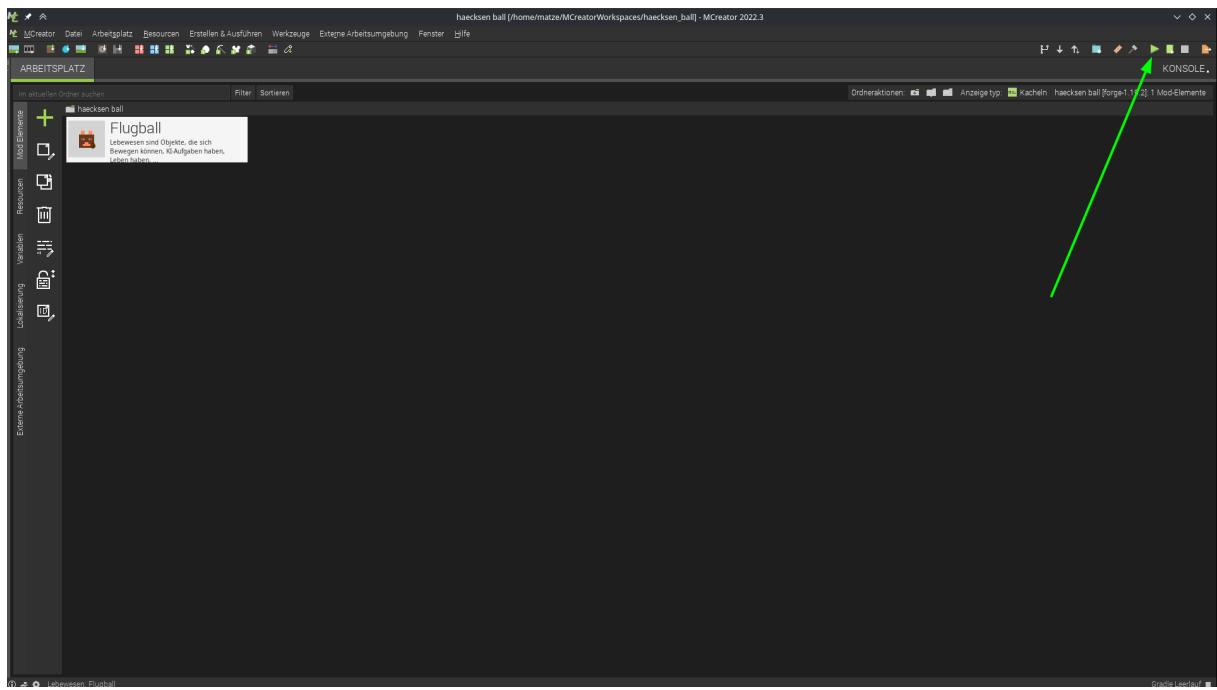


Abbildung 9: Starte Minecraft mit deiner Mod: der Play-Knopf



“Was passiert da im Hintergrund?”

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 10: Inventar: hier findest Du dein erstelltes Lebewesen, bzw. dessen Ei

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

**“Geschafft: Der Ball fliegt!”**

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 anpassen

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:

MCreator: Erstelle deine eigene Mod HaecksenBall!

2022-12-22

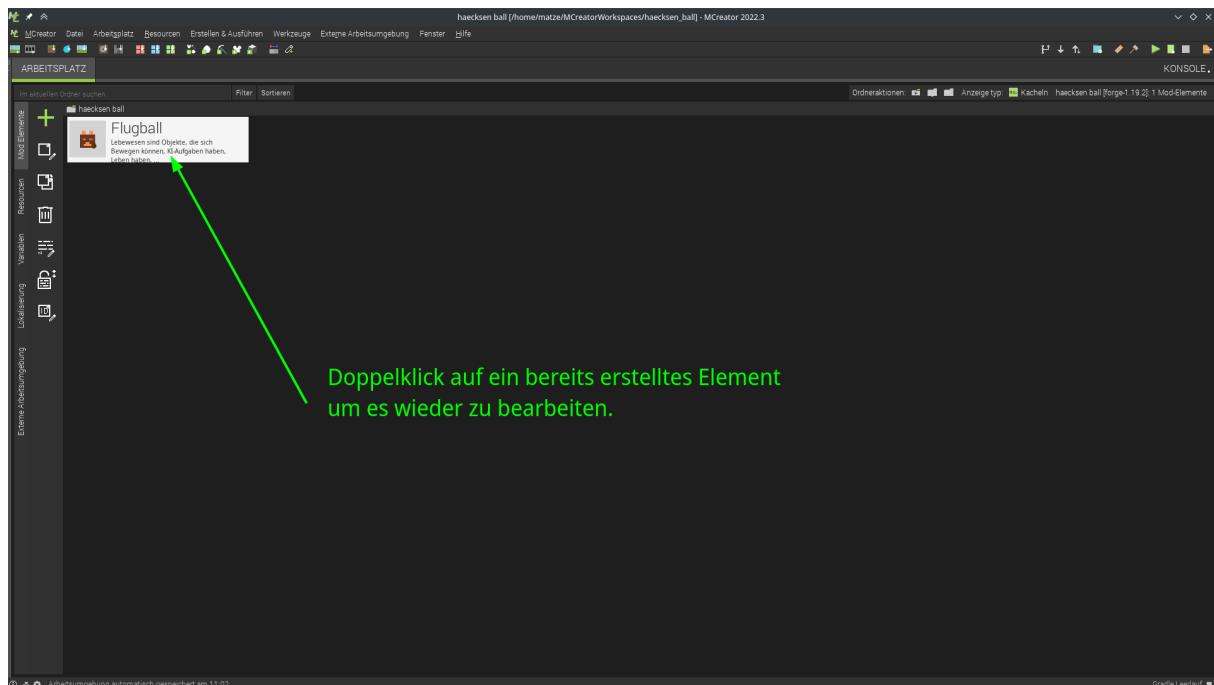


Abbildung 11: Bearbeiten der Eigenschaften: Verhalten ändern

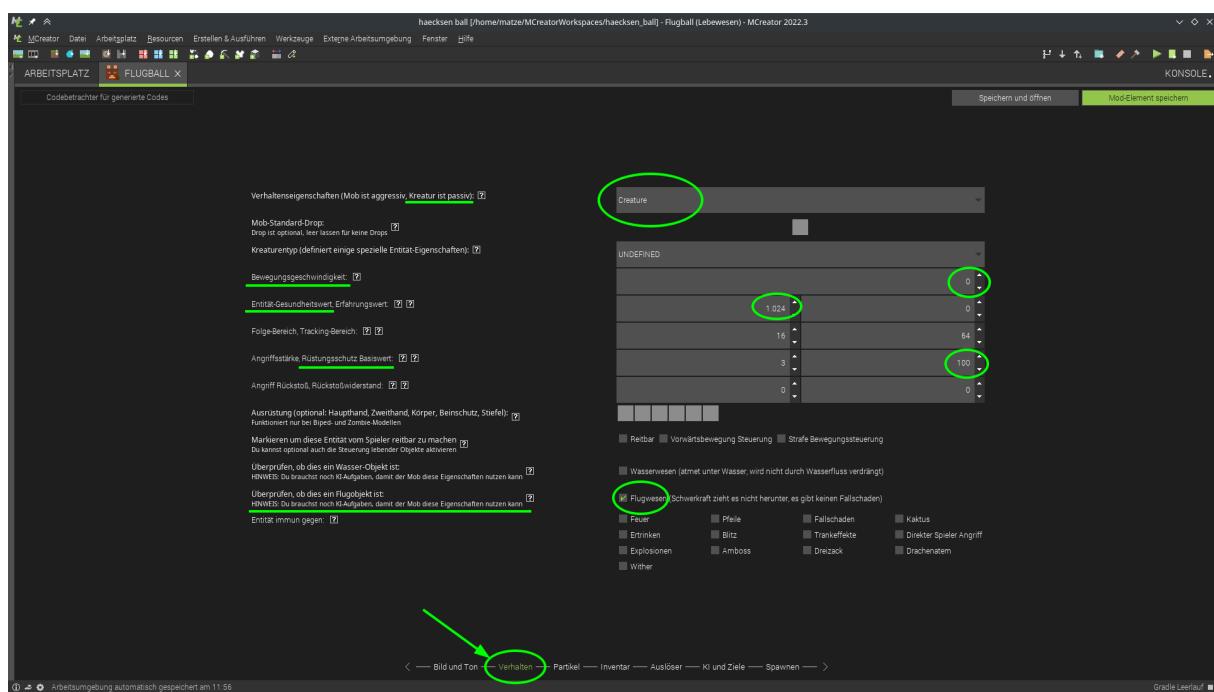


Abbildung 12: Hier kannst Du alles zum Verhalten anpassen

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)



“Ball fertig - yeah!”

Das hast Du schon geschafft:



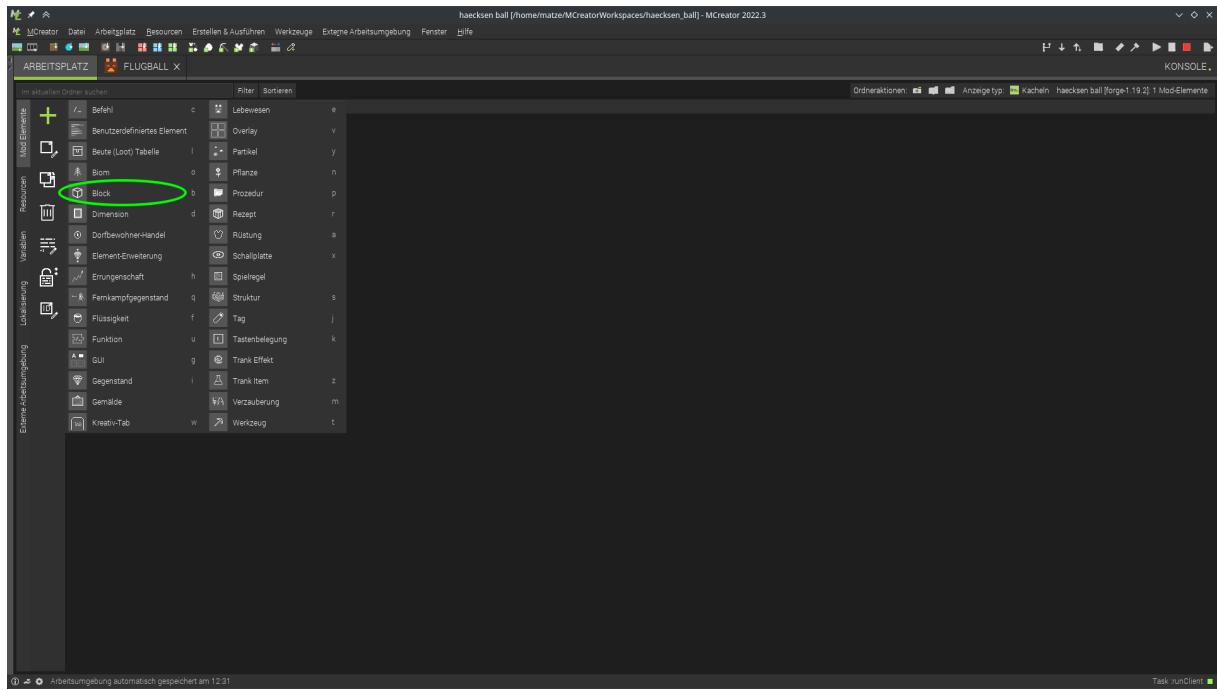
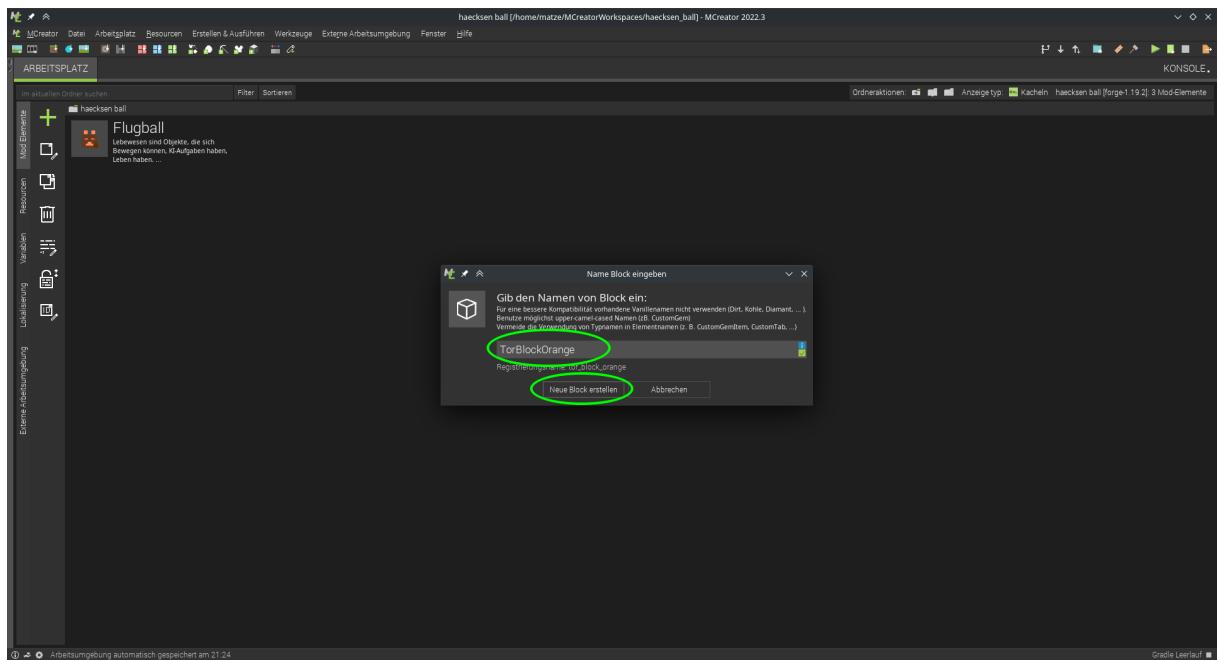
Abbildung 13: Stand Level 1 - Bälle sind geschafft!

- 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 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 ja schon aus unserem ersten

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

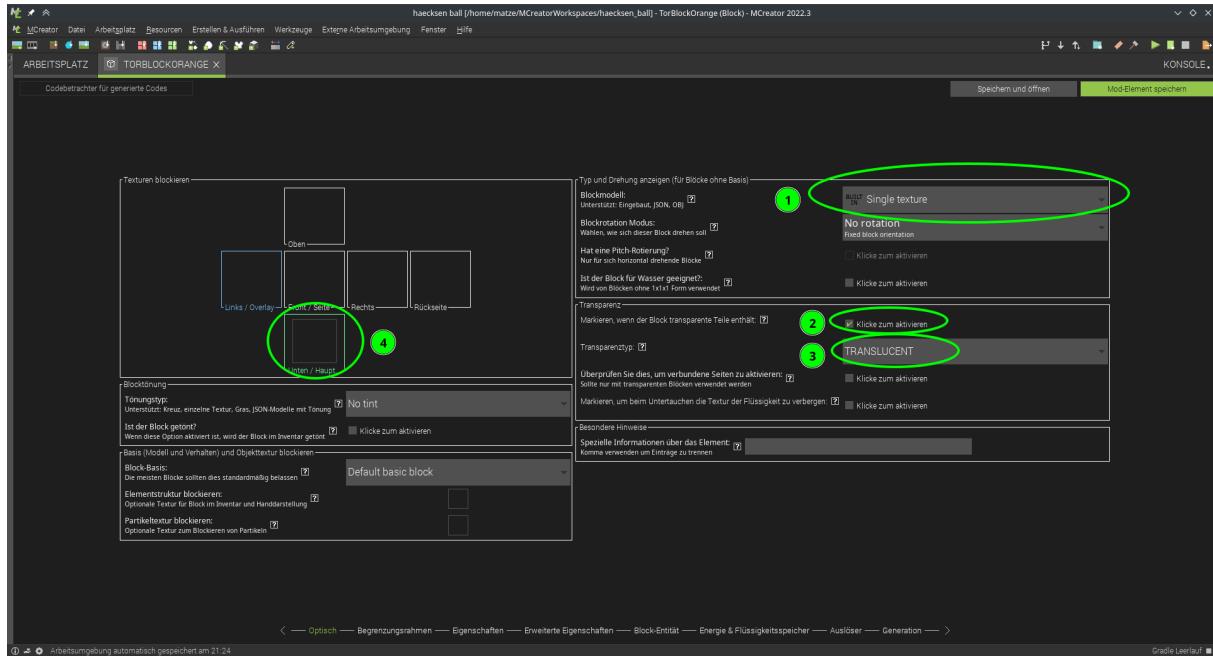


Abbildung 16: Die optischen Eigenschaften des Blocks

1. Wähle hier aus, 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:

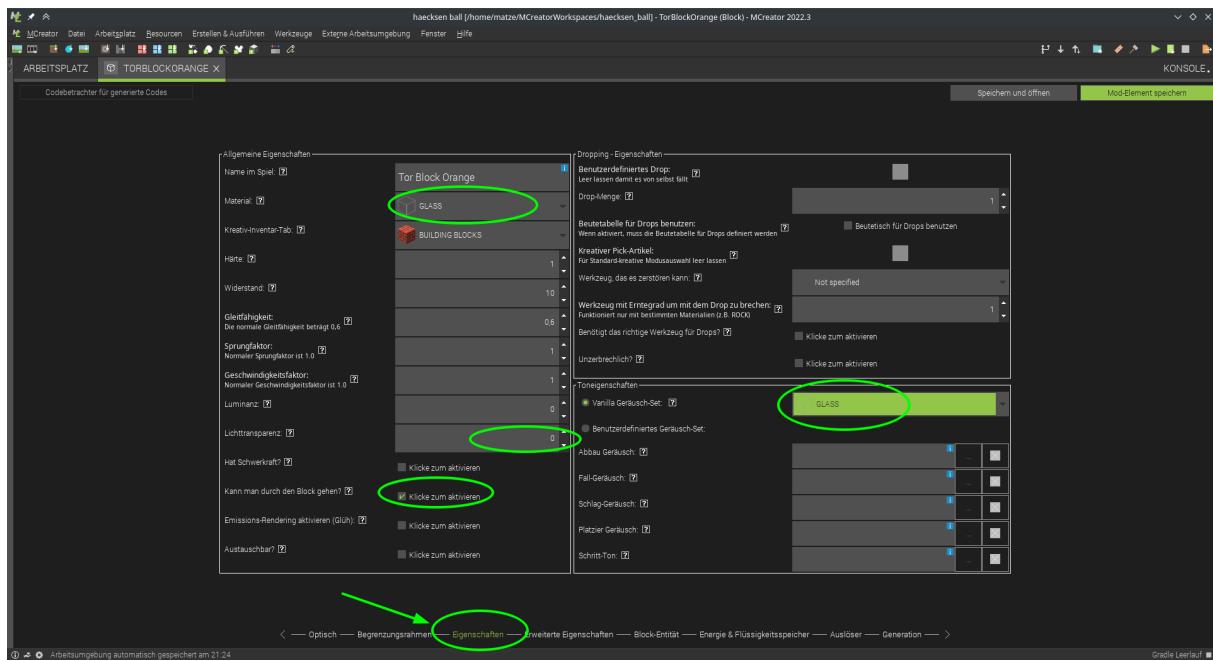


Abbildung 17: Eigenschaft: Glass-ähnlich

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:

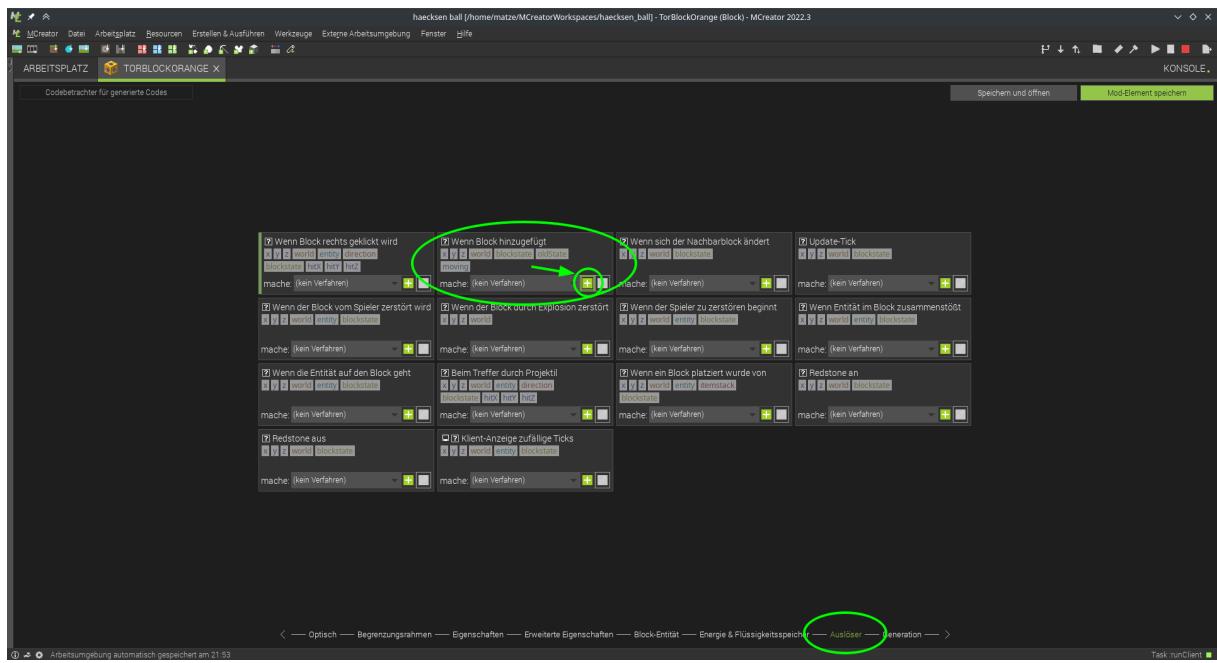


Abbildung 18: Auslöser hinzufügen - erzeugt ein Ereignis

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.

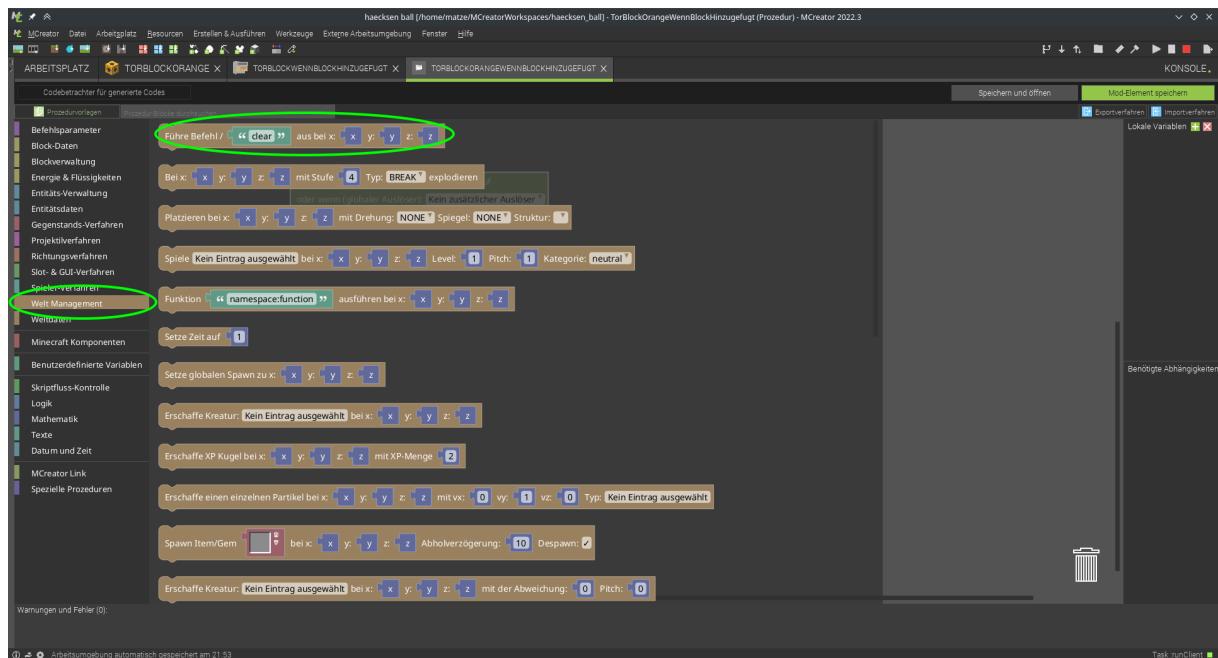


Abbildung 19: Wie können Minecraft-Befehle ausgeführt werden

- 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:

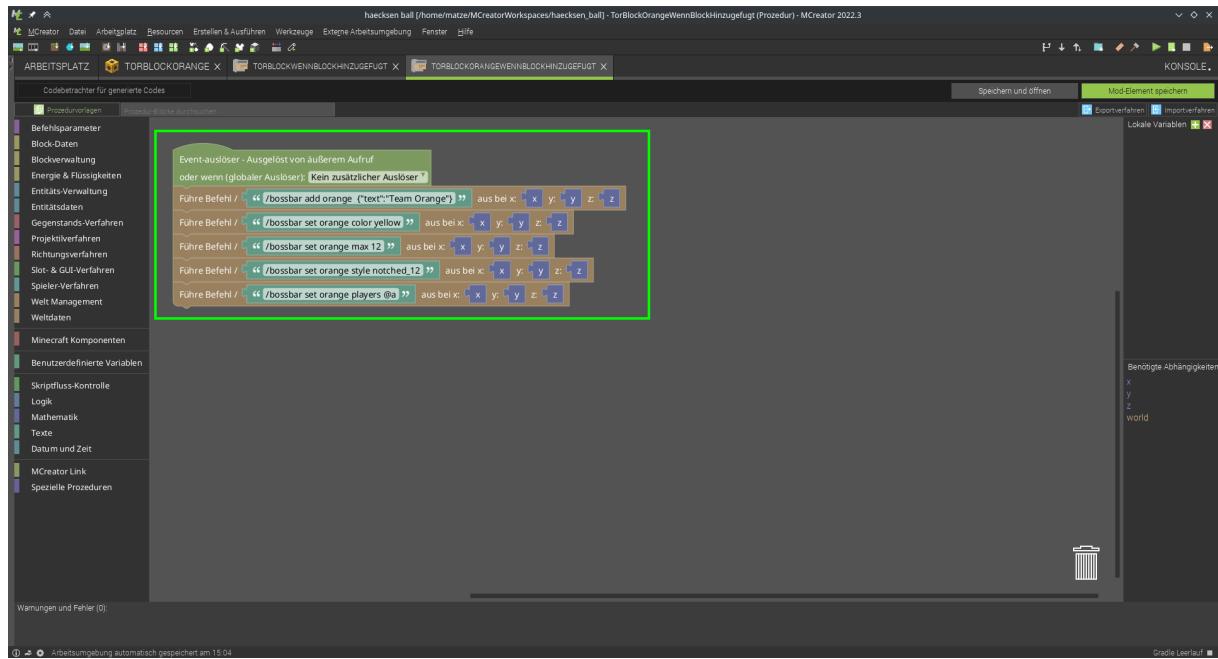


Abbildung 20: Minecraft Befehle in MCreator für bossbar

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 21: Bossbar für Team 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:

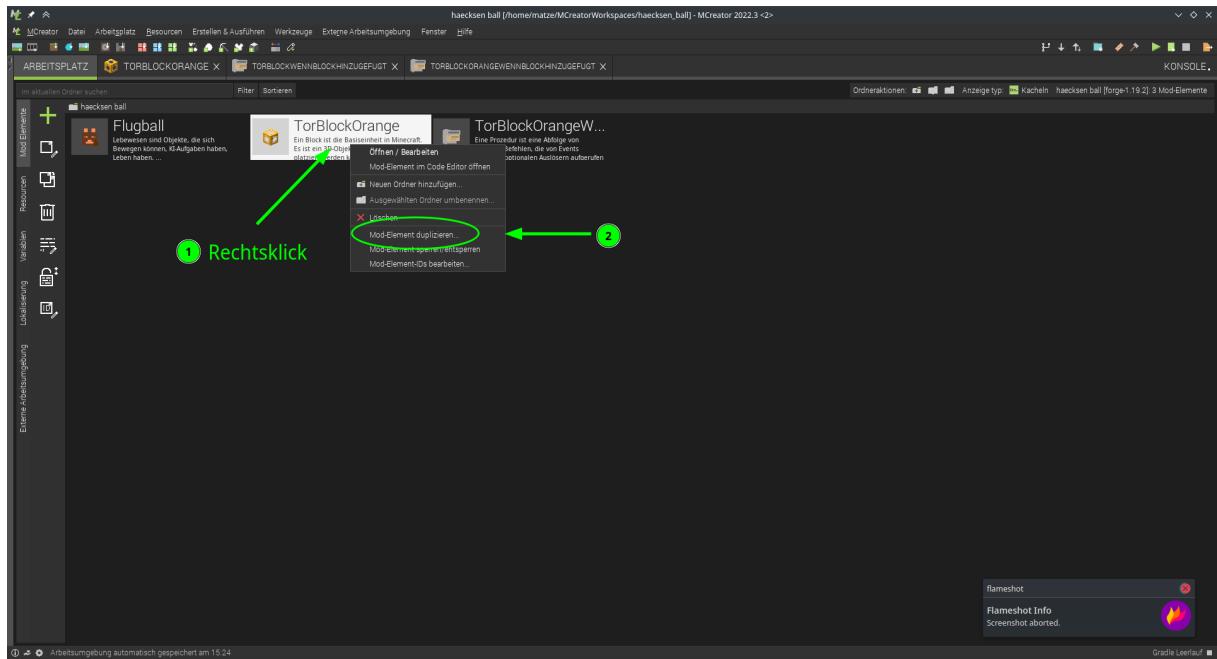


Abbildung 22: Arbeit sparen: so kannst Du bestehende Elemente duplizieren

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:

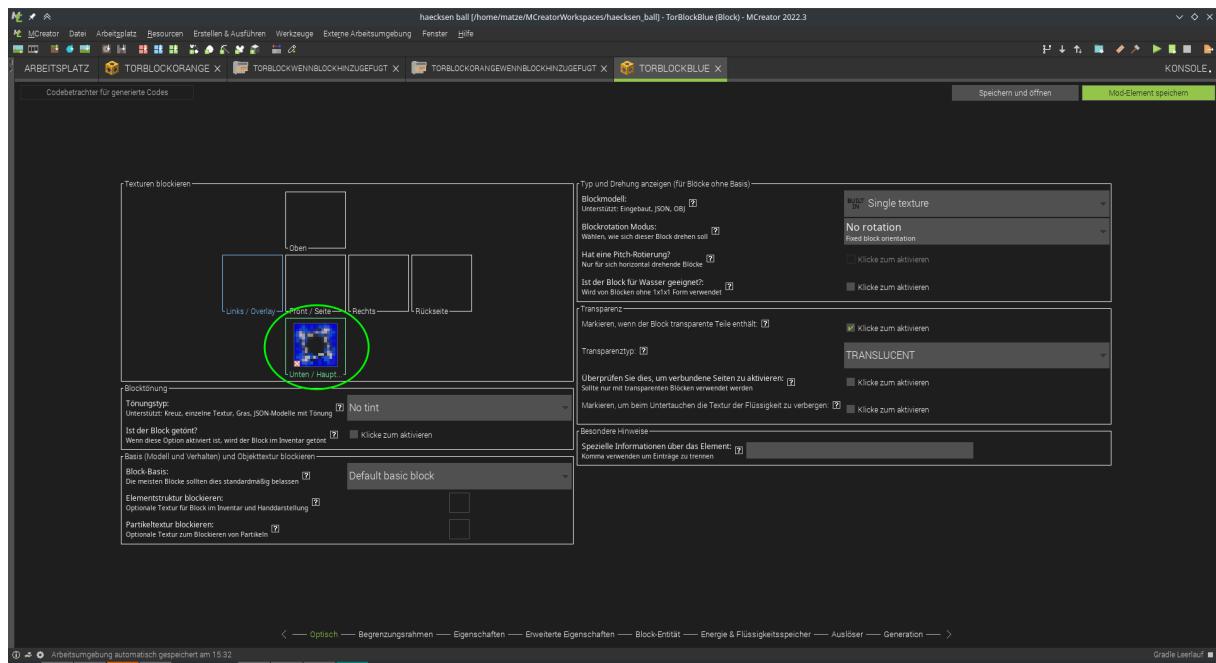
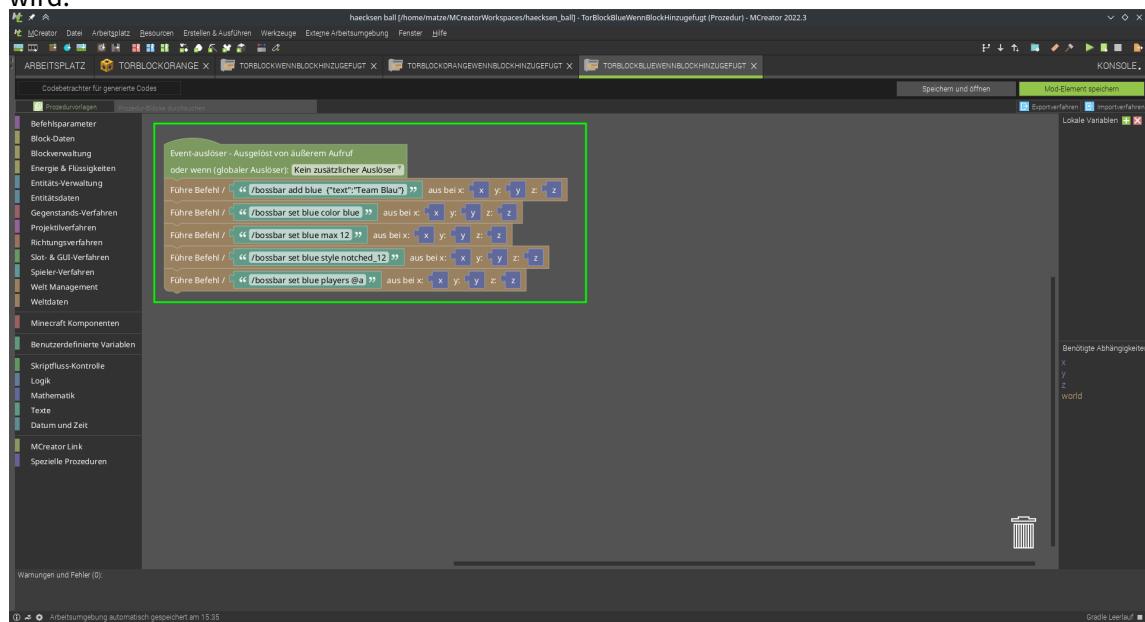


Abbildung 23: Tausche die Textur der Blockes aus

- 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

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



“Variablen - eine kurze Einführung”

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

Hintergrund: Variablen-Typen

Es gibt verschiedene Arten von Variablen, ja nach dem, was man sagen will:

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

Und noch ganz viele mehr...

Achtung: Gute Namen sind gute Helfer

Variablen-Namen können nur aus einem Buchstaben, aber auch ganzen Wörtern bestehen. Wähle den also weise: gib den Variablen Namen, die andere (und auch Du noch 2 Wochen ;-)) noch verstehen. Dann findest Du Dich viel leichter im eigenen Programm zurecht.

Punkte merken

Wir erstellen also eine Variable - in der merken wir uns jeweils die Tore / Punkte pro Team

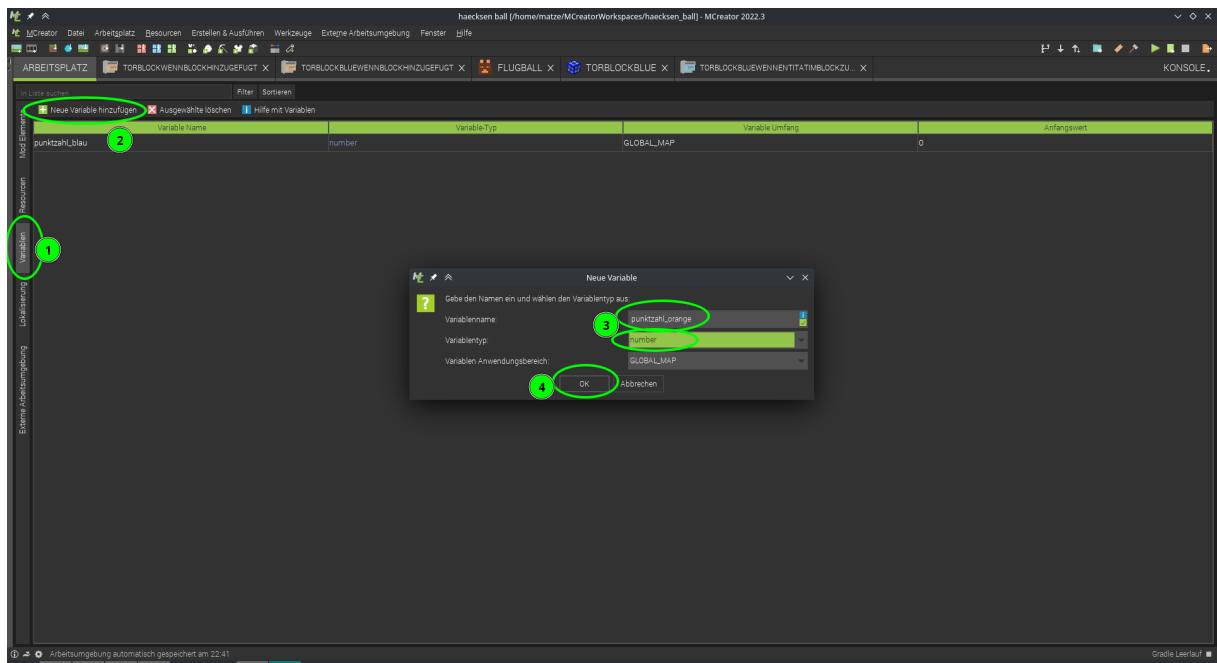


Abbildung 24: So erstellst Du eine neue Variable

Dazu erstellen wir eine variable für jedes team:

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

Tore erkennen

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

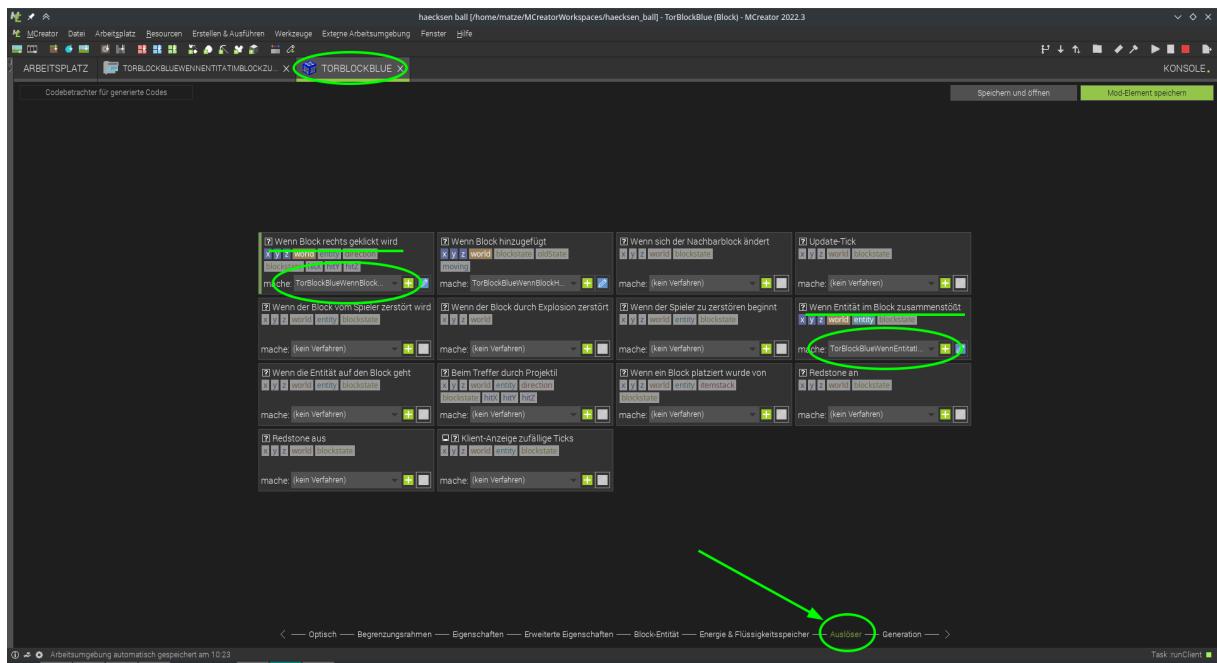


Abbildung 25: Auslöser: So erkennt das Tor, was gerade passiert

Wir fügen 2 Auslöser hinzu:

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

Prozedur 1: Punkte zurücksetzen

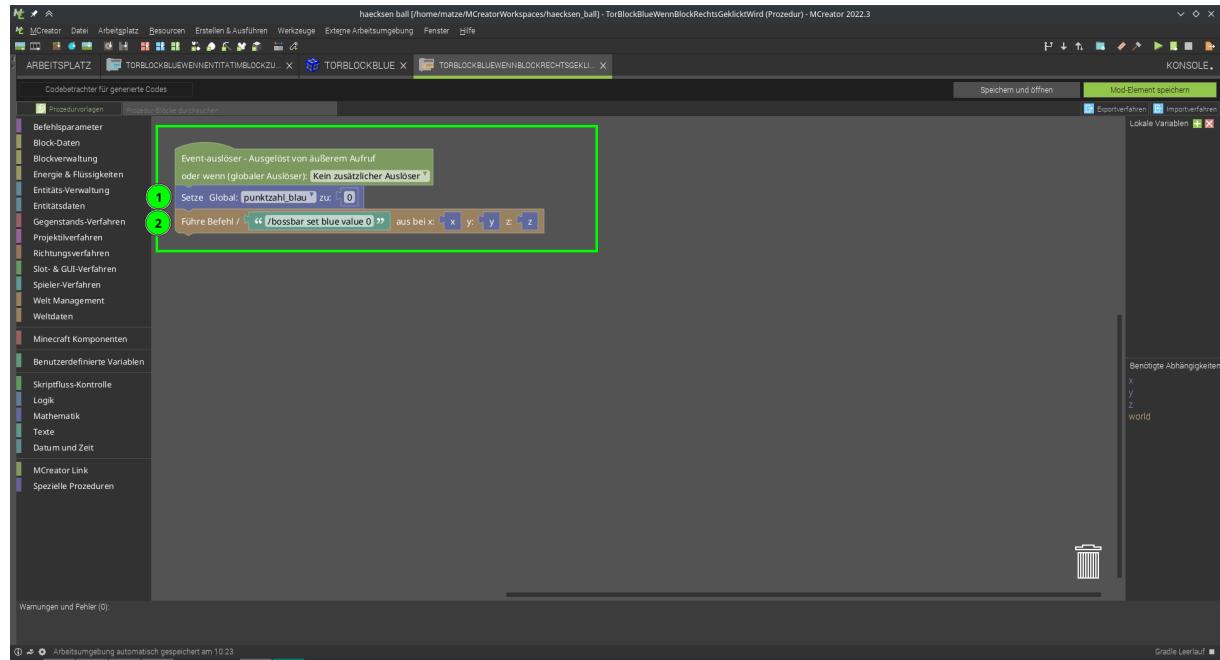


Abbildung 26: Code, um die Tore zurückzusetzen

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. 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.

Prozedur 2: Tore zählen

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

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

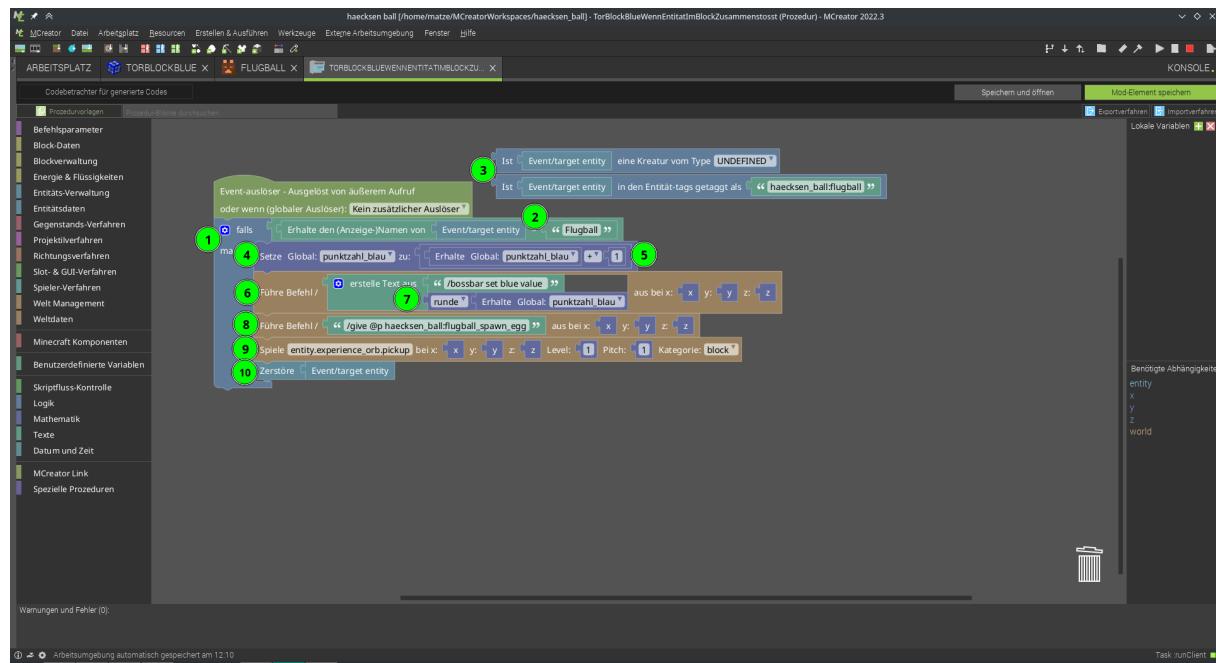


Abbildung 27: Code, um die Tore zu zählen

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 "holen" wir uns den Anzeige-Namen der "Berührenden Entity" mit `Erhalte den Anzeige-Namen von Event/target entity` und prüfen ob der gleich dem String `Flugball` ist. (Muss übereinstimmen, wie der Objektname bei [Bild und Ton von Ball erstellen](#))
3. Diese beiden Zeilen, die vom Hauptcodeblock abgelösten sind, werden nicht ausgeführt. Sie sind nur ein 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_blaue`
 - und fügt mit `+`
 - die Zahl `1` hinzu
 - und Speichert das Ergebnis mit `Setze Global: punktzahl` zu wieder in der Punktezahl für blau ab.
6. Wir führen den Minecraft Befehl für die Bossbar aus - den kennst Du ja schon aus Level 3.
7. Dazu kombinieren wir den text `/bossbar set blue value` (**Achtung! Leerzeichen am Ende wichtig, dass es funktioniert**) mit der `punktzahl_blaue` und runden diese, weil es sich sonst um eine Kommazahl handelt. (zwar immer ,0 aber Minecraft kennt hier keine Kommazahl,

desshalb machen wir die so weg)

8. wir geben dem Spieler, der am nächsten steht (@p) ein neues Spawn-Ei
`/give @p haecksen_ball:flugball_spawn_egg`
9. 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, wenn man XP aufsammelt: `entity.experience_orb.pickup`
10. Am Schluss wollen wir den Ball natürlich "los werden" - er soll verschwinden und mit einem neuen Ball geht es weiter.
 - 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 ;)**



“Blau bekommt einen Punkt?”

Vielleicht wunderst Du Dich, warum wir jetzt dem blauen Team einen Punkt geben, wenn ein Ball ins blaue Tor geht - müsste ja eigentlich anders rum sein oder? So dachten wir, es es einfacher - wenn man mit 2 Teams spielt, geht es ja noch. Aber man kann das ganze ja auch für 3 oder mehr Teams programmieren, dann geht es nicht mehr so einfach. Mit 12 Punkten hat man also verloren...



“Alternative: 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 ;)



“Geschafft! Tore zählen - ich habe fertig!”

Jetzt kann's 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 fehlen nur noch die passenden Prozeduren, die können wir erst mal kopieren vom Team Blau.

- Gehe zum **Arbeitsplatz**
- Rechts auf die Prozedur klicken - wähle **Mod-Element duplizieren**
- Nenne die neuen Prozeduren ...Orange... statt ...Blau...

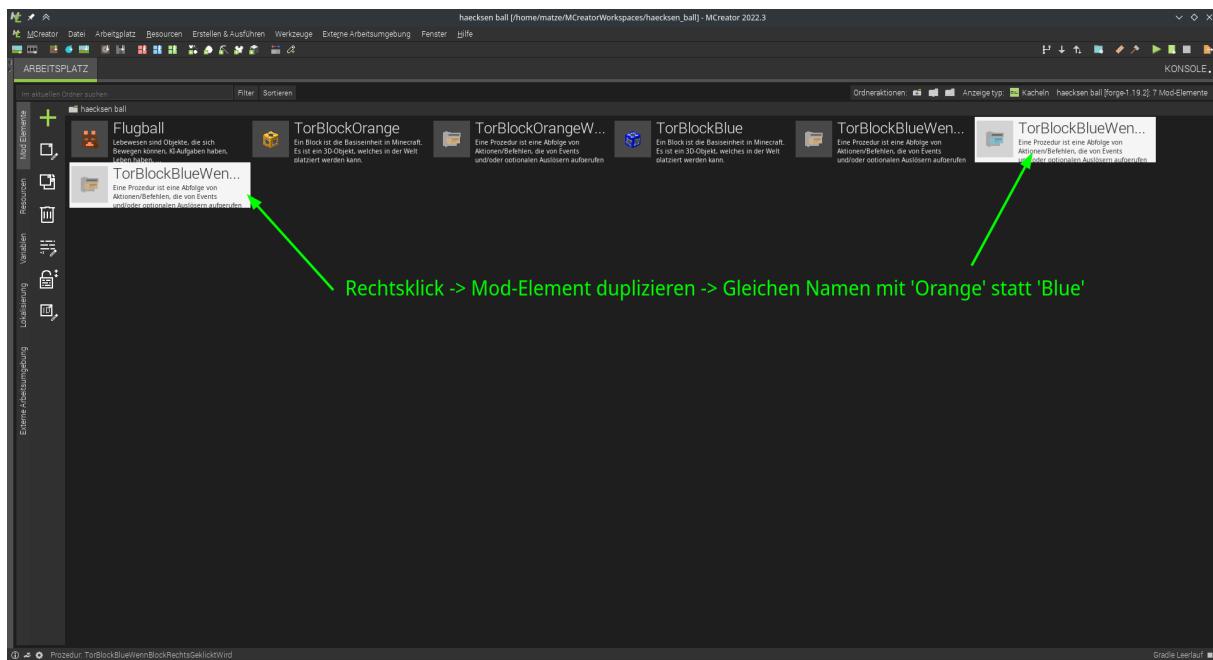
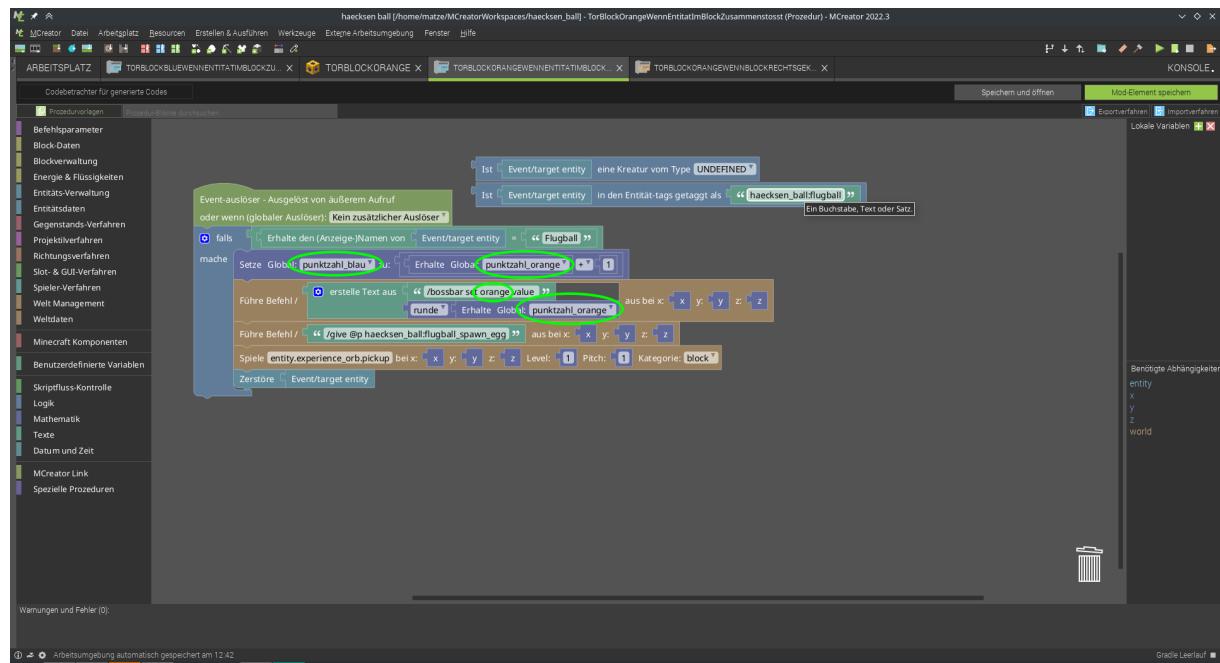
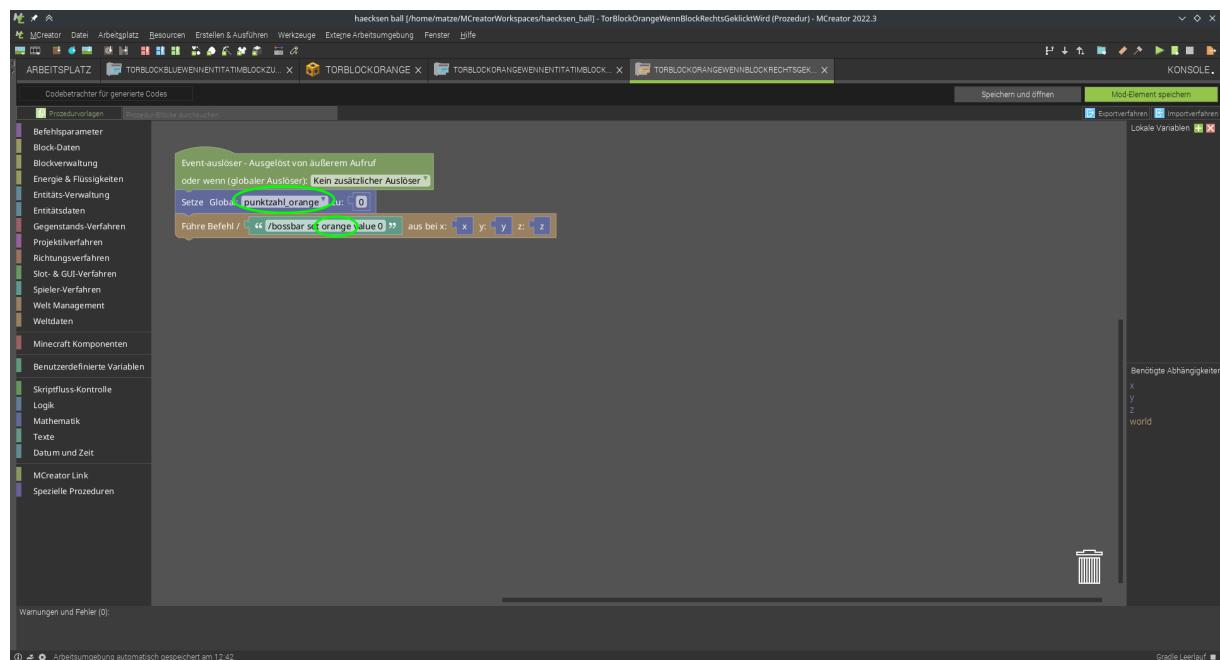


Abbildung 28: Faulheit siegt, Arbeit sparen: dupliziere den bestehenden Code

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

**Abbildung 29:** Neuer Code - Zusammenstoß für Orange...**Abbildung 30:** ... und der Rechts-Click-Code für Orange.

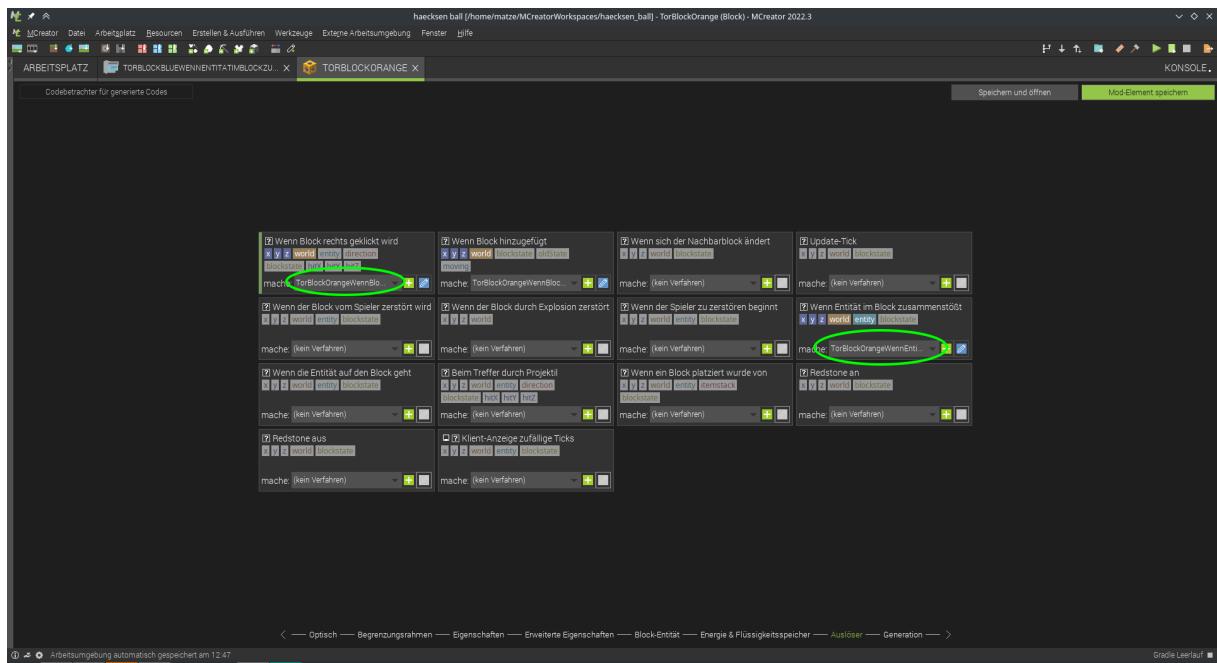


Abbildung 31: Und nicht vergessen!!: Auslöser auch anpassen



“Funktioniert?”

Und nochmal testen ;)

Wenn was nicht klappt, schau oben Schritt für Schritt alles durch, ob Du was vergessen hast.

Level 4: Erweiterte Punkte-Anzeige / Punkte zurücksetzen

Beim testen und spielen, ist uns etwas aufgefallen: Jeder kann ja relativ einfach dem anderen Team Punkte klauen - einfach zum Tor schleifen und rechts klicken. Da haben wir uns folgendes überlegt: wenn es jedes mal eine Nachricht an alle Spieler gibt, wenn Punkte zurückgesetzt werden, wäre der Mogler schnell enttarnt! Schau Dir mal folgenden Programm-Code an:

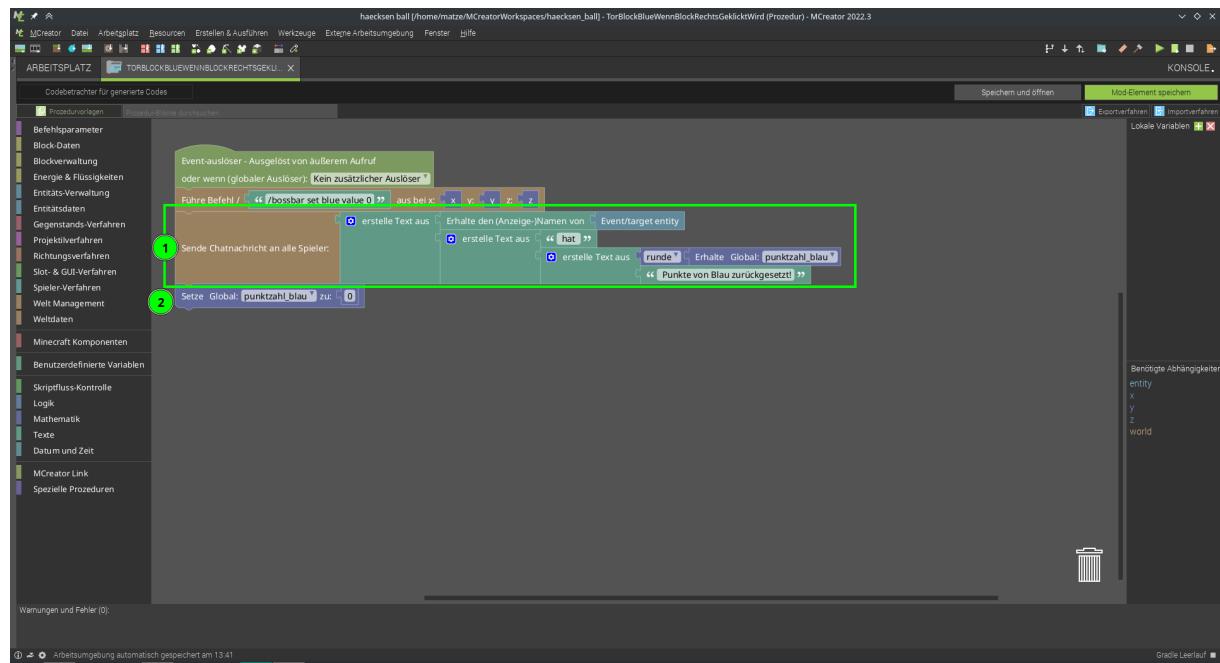


Abbildung 32: Erweiterung des Codes zum Zurücksetzen

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

- Spielername mit *Erhalte den (Anzeige-)Name von > Event/target entity*
- „ hat “ (Leerzeichen nicht vergessen)
- *runde]> Erhalte Global punktzahl_blaau*
- „ 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 ;)"

Starte Dein Programm und teste alles noch mal durch! Geht alles?

Für beide Farben?

Level 5: Die Haecksenball-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
- “+” klicken und “Gegenstand” auswählen

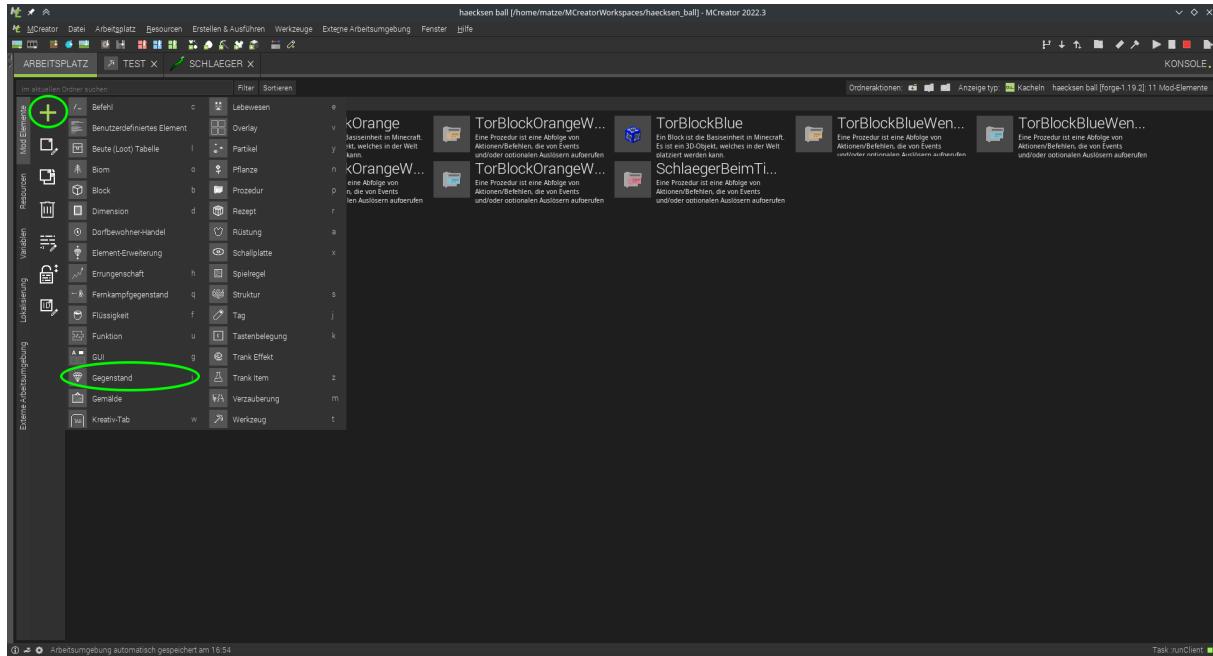


Abbildung 33: Gegenstand erstellen

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

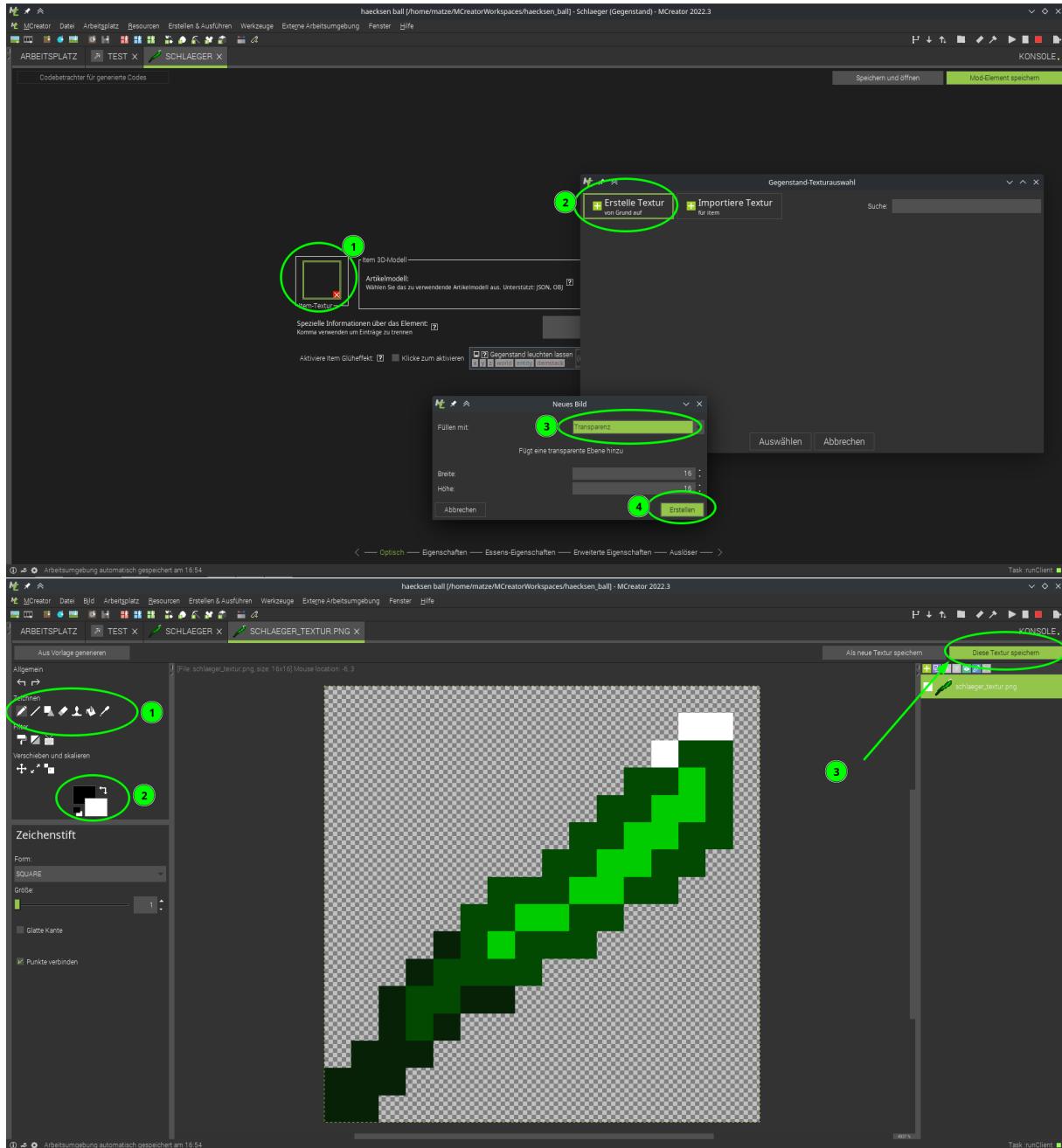


"Umlaute und Programmieren"

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 wirklich 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 **Schlaeger** mit entschieden.

Textur für den Schläger pixeln



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

Dann beim Gegenstand die gerade erstellte Textur nochmal auswählen.

Eigenschaften des Schlägers

Jetzt müssen wir noch ein paar Sachen festlegen, damit auch alles so ist, wie wir das für einen Schläger erwarten. Wechsel dazu zu den *Eigenschaften*.

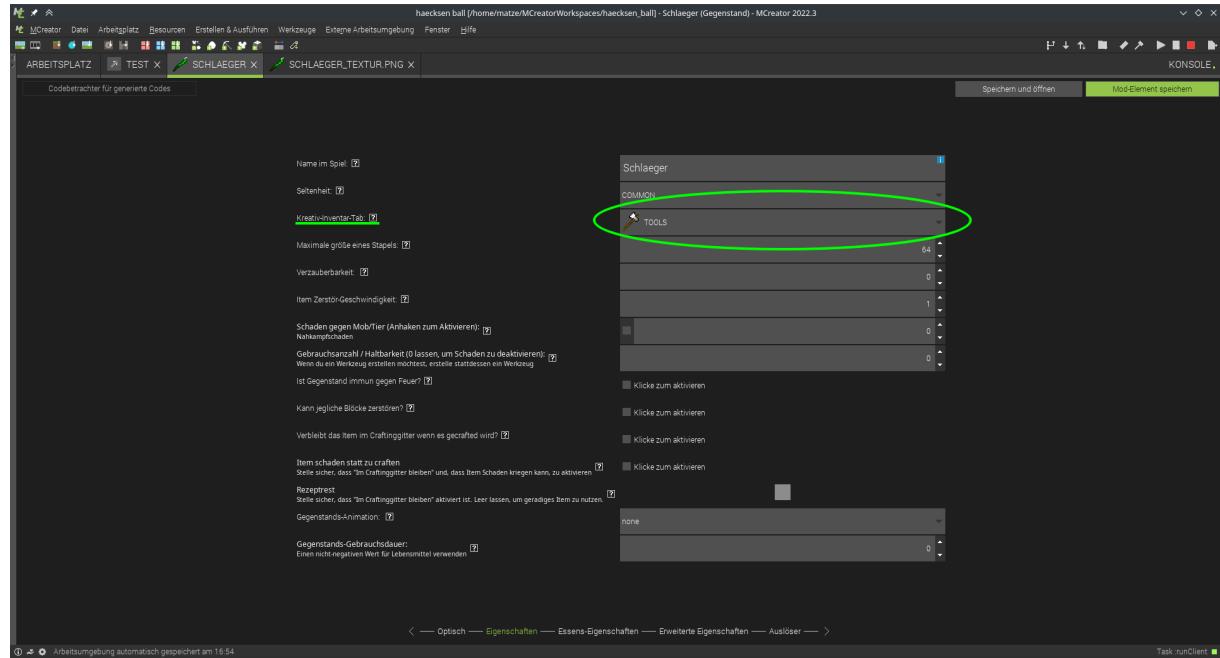


Abbildung 34: Gegenstand Eigenschaften

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

Rückstoß

Der Schläger soll natürlich jetzt besondere Eigenschaften haben, wie eich echter Sport-Schläger. Weil es aber in der normalen Eigenschaften keine Einstellung dafür gibt, wie viel Rückstoß der Schläger haben soll, behelfen wir uns anders.

Genau - die Auslöser kann man ja nutzen, um auf Sachen im Spiel zu reagieren. Du nutzt jetzt einen neuen Auslöser - der immer läuft, wenn man ihn in der Hand hält:

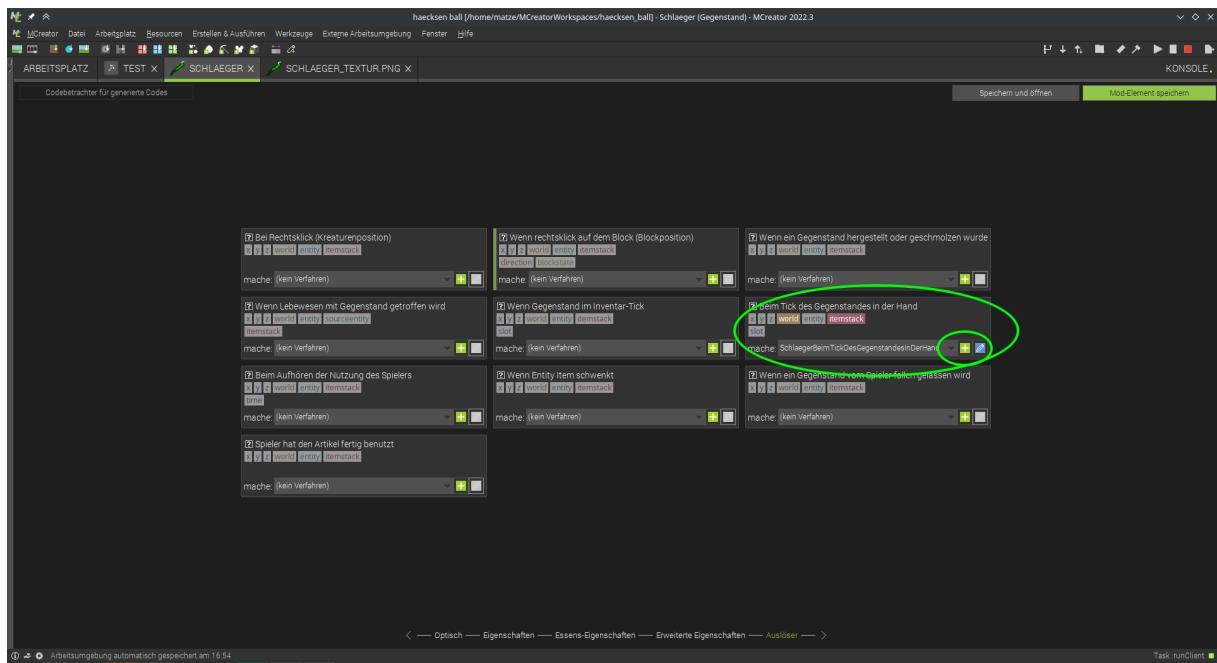
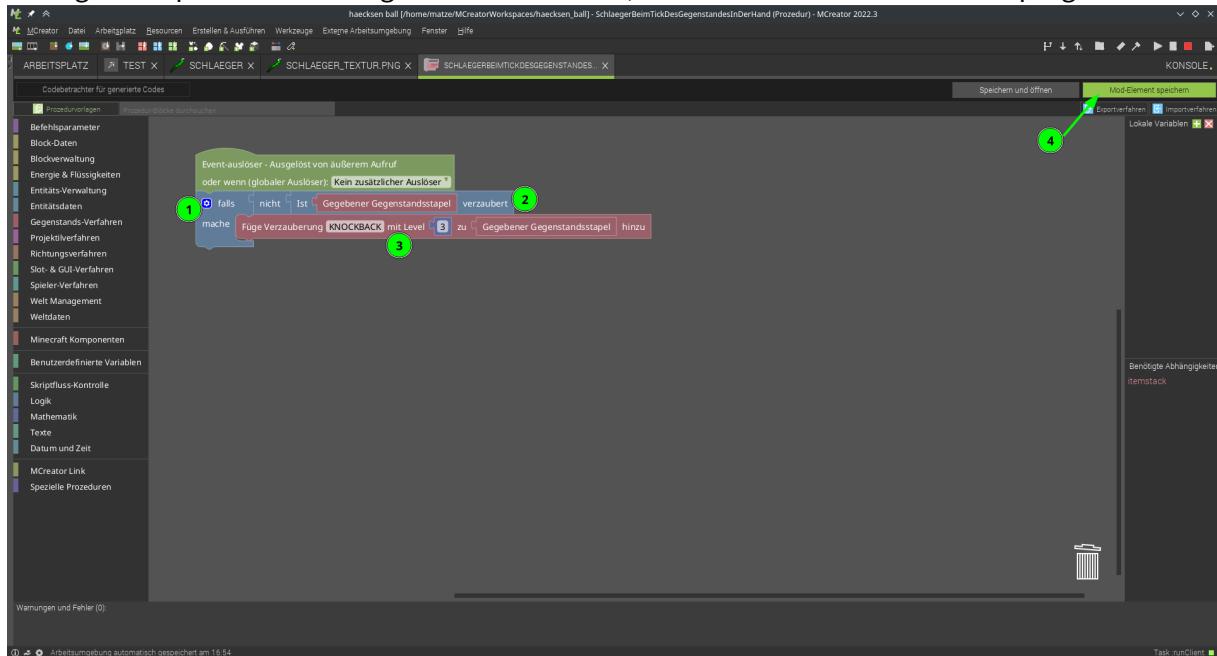


Abbildung 35: Auslöser für den Schläger - Bei jedem Tick

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



1. Testen ob der Gegenstand noch nicht verzaubert ist - sonst würde er ja jeden Tick neu verzaubert
2. Dazu „verneinen“ wir die Antwort auf die Frage *Ist Gegebener Gegenstandsstapel verzaubert*. Das

bedeutet, mit dem Befehl `nicht` drehen wir das Ergebnis von `ist ... verzaubert` um, und erhalten `wahr`, wenn er nicht verzaubert ist.

3. und dann fügen wir die Verzauberung `KNOCKBACK` auf deutsch “Rückstoß” mit dem `Level *3*` hinzu.



“Ausprobieren!”

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



Abbildung 36: 1 ist in der Hand - und verzaubert!



“Alternative Schläger”

Man kann auch mit einem Bogen (vtl 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 37:** Beispiel Haecksenball-Arena**“Tipp: Schneller bauen”**

Keine Lust lange rumzubuddeln?

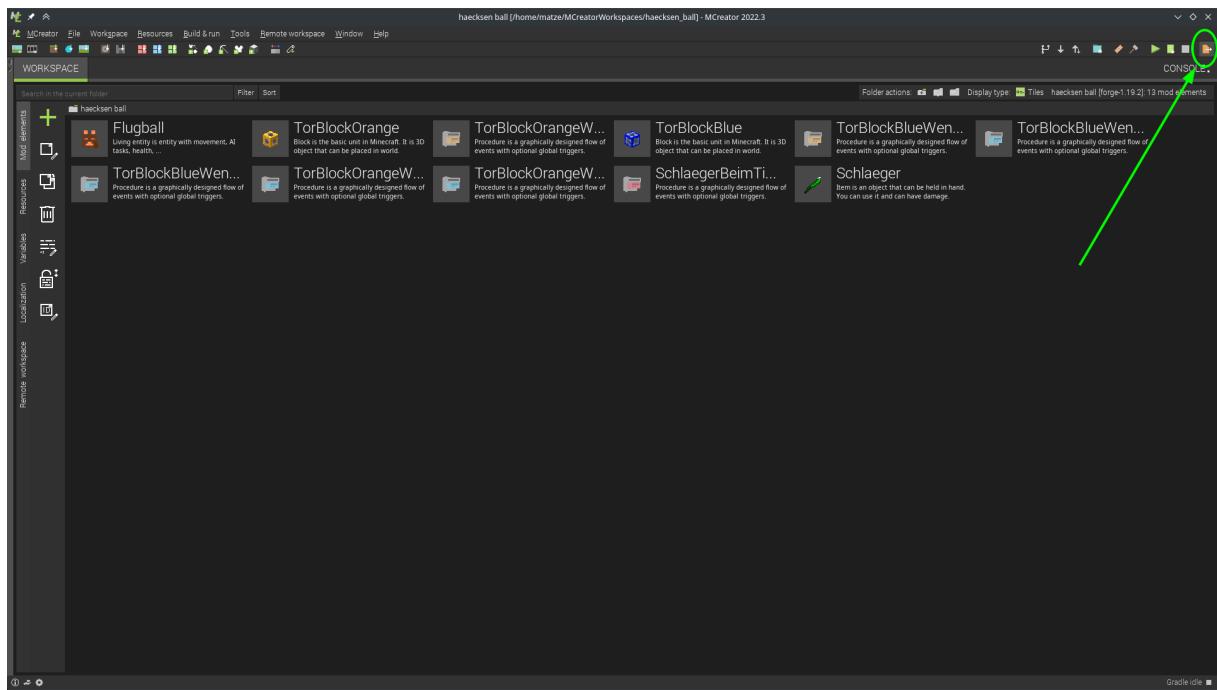
[Minecraft Wiki](#): /fill schau dir mal den Fill-Befehl an!

Level 7: Exportieren & Installieren der Mod

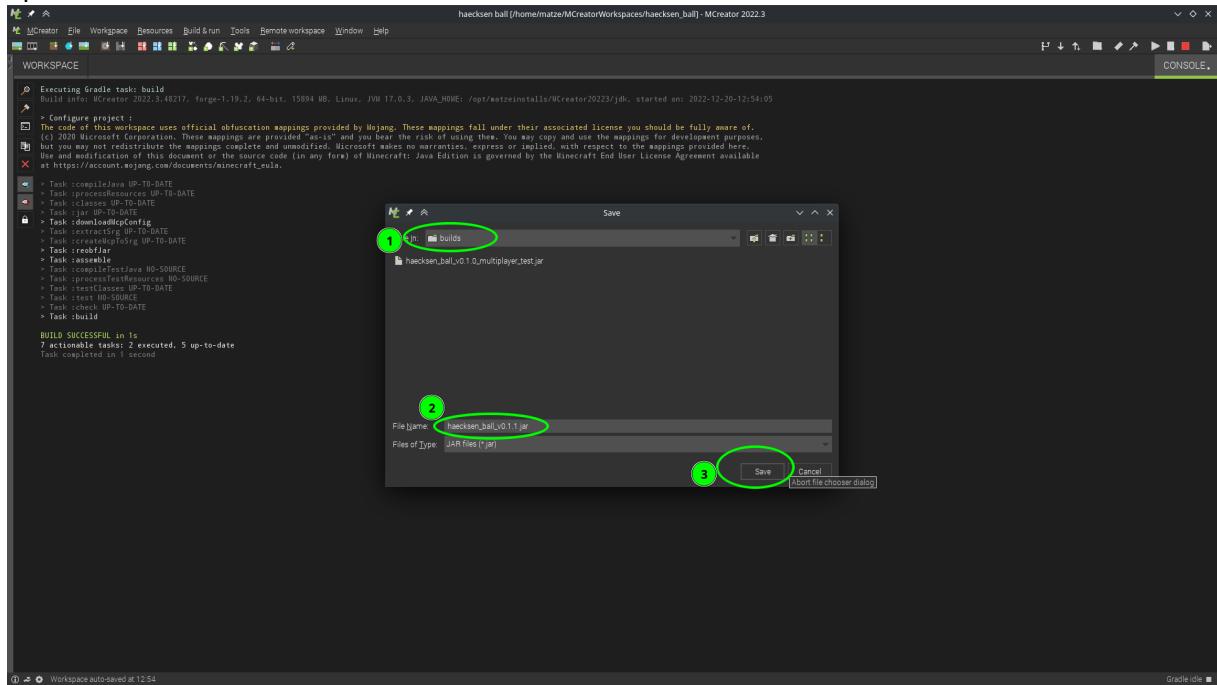
So alles fertig - jetzt wird es Zeit, die Mod zu exportieren und zu veröffentlichen, damit auch deine Freunde die Mod herunterladen und installieren können.

JAR-Datei exportieren

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 Installieren

Am einfachsten klappt die Installation von Mods mit einem erweiterten Launcher, zum Beispiel unserem Lieblings-Launcher “[ATLauncher](#)”:

1. ATLauncher herunterladen und installieren: <https://atlauncher.com/>
2. Neue Minecraft Instanz erstellen:
 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 auf den Reiter **“Instances”**
4. Klicke den Knopf **“Edit Mods”**
5. Klicke auf **“Add Mod”** und wähle die JAR Datei von oben aus



“Mods & Server”

Gemeinsam auf einem Server spielen? Die einfachste Variante ist, in Minecraft von einem Rechner das Spiel freizugeben:

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

1. Starten Sie das Spiel **Minecraft** wie gewohnt.
2. Wählen Sie “Singleplayer” und starten Sie die gewünschte Welt, oder erstellen Sie eine neue.
3. Klicken Sie im Spiel auf die “Escape Taste” und wählen Sie anschließend “Für **LAN** öffnen”.
4. Passen Sie die Einstellungen an und Klicken Sie auf “OK”.

Wenn Du auf einem “öffentlichen” Server im Internet gemeinsam spielen willst, muss dieser auch mit Forge laufen und auch die Mod installiert haben.