CAMP MINECRAFT AI

– Doku 1 –

Kapitel 2 – Die Programme

Unser Camp wird sich verschiedener Programme und Anwendungen bedienen, um die Arbeit mit dem Code zu vereinfachen. Dabei werden wir unseren Code in der Entwicklungsumgebung mit dem Namen „IntelliJ IDEA“ schreiben, wo wir bereits eine Oberfläche für euch vorbereitet haben, in welcher ihr nur noch eure Dateien und Code an der richtigen Stelle einfügen und bearbeiten müsst – aber dazu kommen wir, sobald es nötig wird, nochmal genauer.

Des Weiteren brauchen wir noch eine Möglichkeit, 3-dimensionale Texturen für die Kreaturen zu erstellen, und dafür benutzen wir die weit verbreitete Lösung mit „Blockbench“. In diese werden wir uns auch gleich begeben, denn wir wollen uns einmal mit der Funktionalität dieser Anwendung vertraut machen, damit ihr wisst, wie ihr damit umgeht, um eure Entitäten zu erstellen.

Kapitel 2.1 – Grundlagen des 3D-Raums

Schauen wir uns mal gemeinsam die Steuerung und sonstige Grundlagen für Blockbench an.

Im 3-dimensionalen Raum (3D ist die Kurzschreibweise dafür), haben wir 3 Achsen, auf denen wir uns bewegen können – die X-Achse, Y-Achse und die Z-Achse (Manchmal auch anders benannt, aber standardmäßig werden sie so benannt). Übersetzt bedeutet das für uns also:

- X-Achse: Bewegung nach rechts und links
- Y-Achse: Bewegung nach oben und unten
- Z-Achse: Bewegung nach vorne und hinten

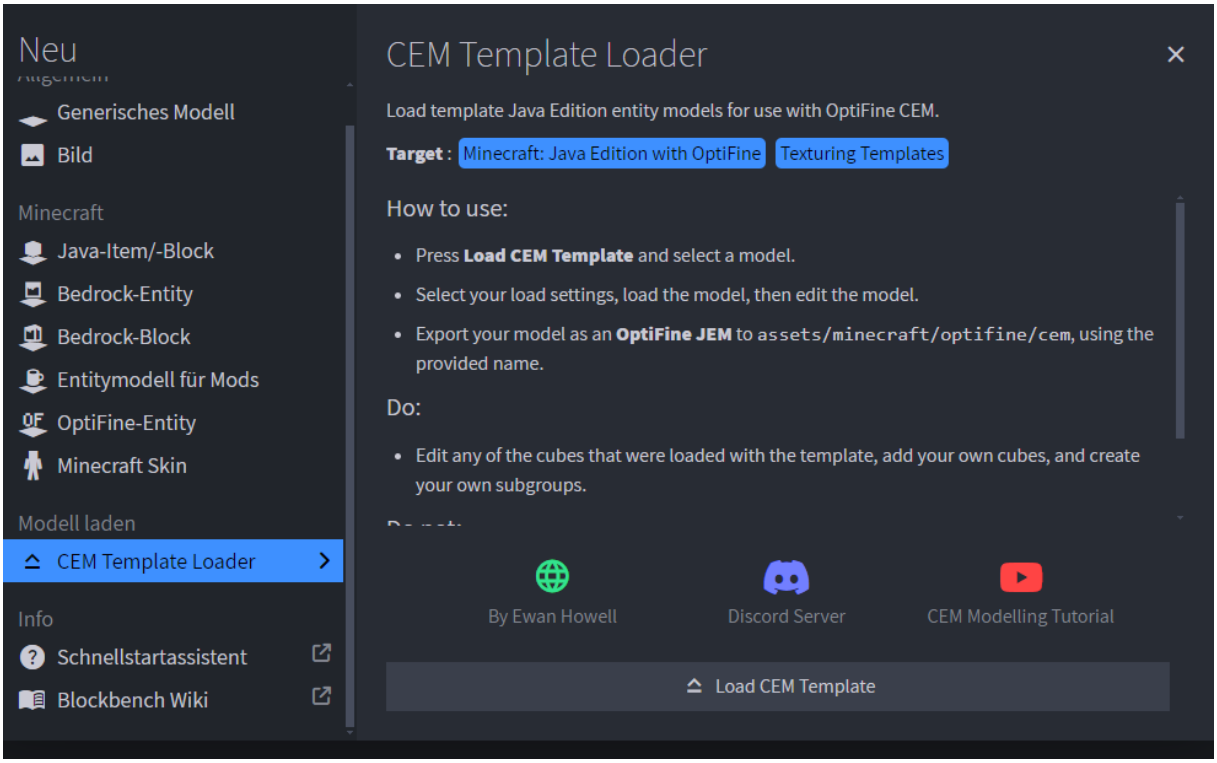
In Blockbench werden die drei Ebenen durch verschieden farbige Pfeile angezeigt. Der rote Pfeil repräsentiert die X-Achse, der grüne die Y-Achse und der blaue die Z-Achse.

Um eure Sicht zu bewegen, könnt ihr die mittlere Maustaste gedrückt halten (also das Mausrad), um euch um den Fixpunkt zu rotieren, wenn ihr dabei noch die *shift* Taste gedrückt haltet, könnt ihr den Blick auf der Y-Achse verschieben. Mit dem Mausrad könnt ihr einfach hinein oder herauszoomen. Pfeile könnt ihr per linker Maustaste gedrückt halten ziehen und verändern.

Macht euch nun zunächst einmal Gedanken, was ihr überhaupt erstellen wollt. Soll eure eigene Kreatur ein friedliches liebes Tier sein, vielleicht ein Begleiter für euch, oder eine neue Kreatur, die die Weiten der Welt erkundet und glücklich ist? – Als Vorlage im Spiel haben wir hier das Stachelschwein (engl.: porcupine), oder soll die Kreatur, die ihr erschaffen wollt, lieber ein furchtsamer, aggressiver Räuber sein, eine Kreatur, die alle um sie herum in Angst und Schrecken versetzt beim Anblick?

Je nachdem, wofür ihr euch entscheidet, könnt ihr euch gleich einmal Gedanken machen, ob ihr irgendein bereits vorhandenes Wesen als Vorlage nehmen wollt, oder lieber etwas ganz Neues, einfaches erschaffen wollt.

Damit wir mal gesehen haben, was es alles zur Auswahl gibt zum Verändern – hier einmal die Liste an Möglichkeiten die bereits als Modell vorhanden sind, und nur darauf Warten angepasst zu werden und eine neue Textur zu bekommen. Diese finden wir, wenn wir „Blockbench“ einmal öffnen, und uns den CEM Template Loader anschauen. Diesen finden wir hier:

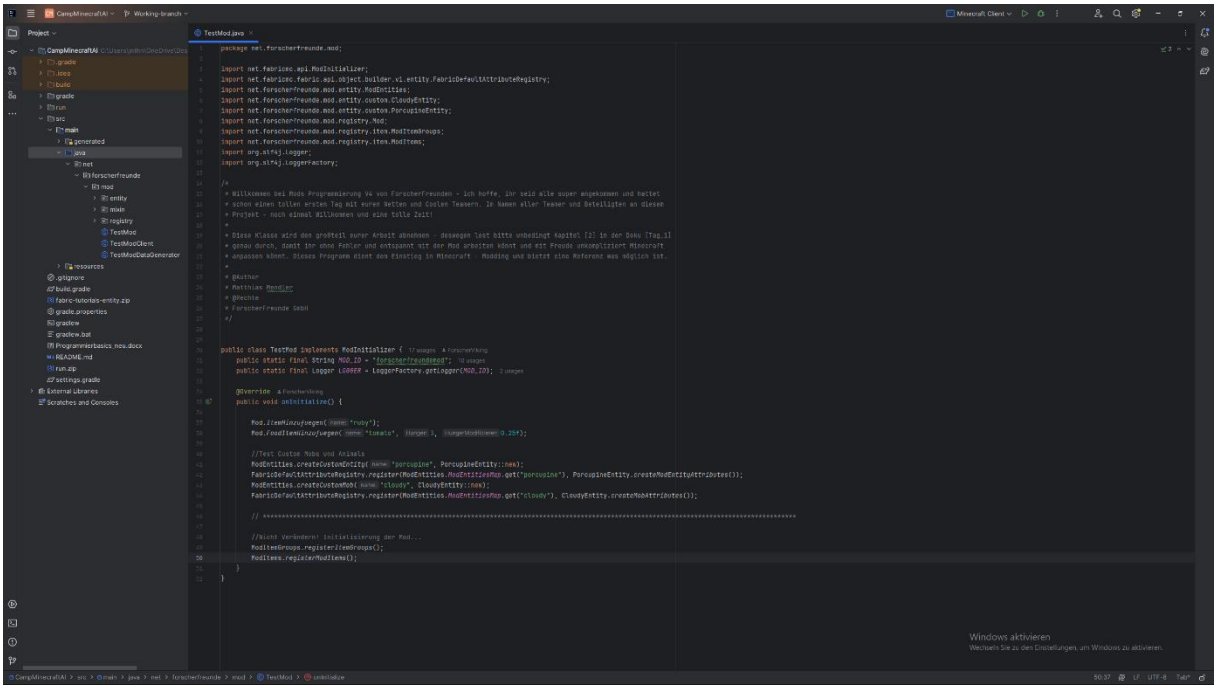


Wenn wir nun auf den Knopf „Load CEM Template“ klicken, wird uns die Liste an allen verfügbaren Modellen angezeigt, welche wir verändern können. Hier könnt ihr euch einmal umschauen und inspirieren lassen – aber behaltet im Hinterkopf, was eure Kreatur später können und machen soll – und dementsprechend auch wie komplex der Code und die Integrierung der Kreatur in Minecraft wird. So werden alle fliegenden Kreaturen – wie blazes, parrot, bat usw. sehr komplex sein, das Verhalten korrekt nachzustellen.

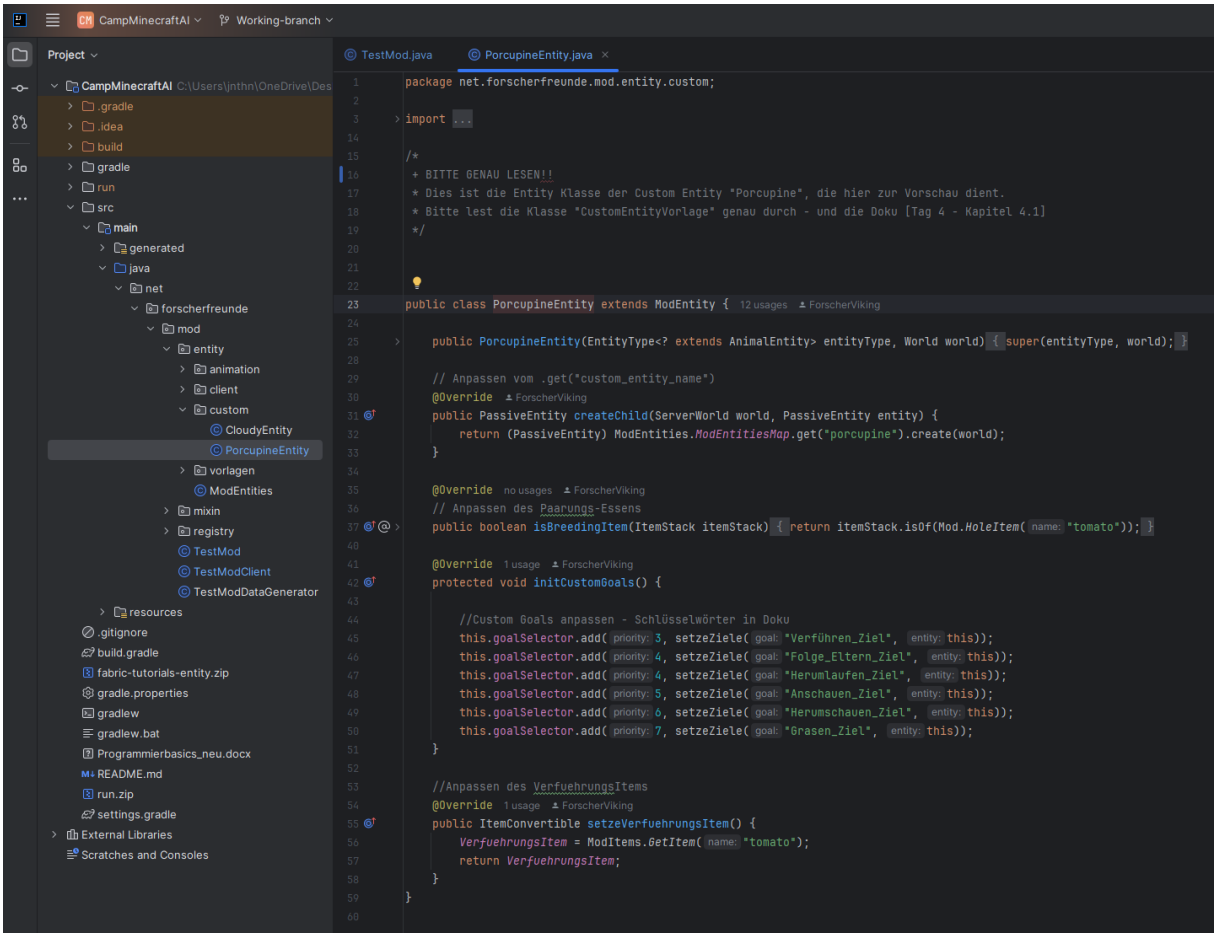
Wenn ihr euch nun für eine Kreatur entschieden habt, gehen wir aber erst mal noch zu den Grundlagen zurück, und schauen uns erst einmal an, wie denn die KI in Minecraft aussieht.

Kapitel 2.2 – Die KI und das Verhalten in Minecraft (vereinfacht)

Dafür öffnen wir gemeinsam einmal IntelliJ mit dem Projekt CampMinecraftAI. Das sollte nach dem Laden dann wie folgt aussehen:



Um einmal einen Blick auf die Implementierung von einem Algorithmus – also der KI von einer Entität in Minecraft zu bekommen, schauen wir uns einmal die vereinfachte Klasse des Stachelschweins an, die wir für euch vorbereitet haben. Dafür öffnen wir auf der linken Seite des Fensters die Ordnerstruktur, und suchen nach der Klasse „PorcupineEntity“ in dem Ordner „custom“:



Dort sehen wir nun verschiedene sogenannte Methoden, also Code Abschnitte, welche eine bestimmte Funktion ausführen. Die für uns wirklich spannende Methode, ist die mit dem Namen „initCustomGoals()“:

```
@Override 1 usage  ForscherViking
protected void initCustomGoals() {

    //Custom Goals anpassen - Schlüsselwörter in Doku
    this.goalSelector.add( priority: 3, setzeZiele( goal: "Verführen_Ziel",  entity: this));
    this.goalSelector.add( priority: 4, setzeZiele( goal: "Folge_Eltern_Ziel",  entity: this));
    this.goalSelector.add( priority: 4, setzeZiele( goal: "Herumlaufen_Ziel",  entity: this));
    this.goalSelector.add( priority: 5, setzeZiele( goal: "Anschauen_Ziel",  entity: this));
    this.goalSelector.add( priority: 6, setzeZiele( goal: "Herumschauen_Ziel",  entity: this));
    this.goalSelector.add( priority: 7, setzeZiele( goal: "Grasen_Ziel",  entity: this));
}
```

Diese Methode ist quasi das gesamte Gehirn der Entity, also legt das Verhalten in verschiedenen Situationen und Reaktionen fest. Dabei haben wir zwei spannende Bereiche, die wir uns genauer anschauen, um die Funktionalität zu verstehen. Wir setzen in dieser Methode ein sogenanntes Ziel in die Liste an Zielen, die diese Entität verfolgt. Einfach gesagt, sind das also die Spielregeln, an die sich die Entität versucht zu halten. Dabei ist der wichtigste Punkt welches Ziel wann abgearbeitet werden soll, also welches Verhalten wann gezeigt wird. Stellt euch vor, ihr habt die Möglichkeit nach der Schule erst etwas zu essen, mit euren Freunden draußen euch zu treffen und zu spielen, Hausaufgaben zu machen, oder online zu Zocken. Aber was davon macht ihr jetzt? In welcher Reihenfolge?

Genau das regelt bei uns im Code von der KI das Attribut „Priorität“ (hier in Grau markiert, engl.: priority). Dieses Attribut geben wir einfach mit einer ganzen Zahl an, also hier im Beispiel mit 3, 4, 5, usw. … Hierbei ist wichtig zu beachten, dass je niedriger die Zahl ist, desto höher ist die Priorität in der Liste. Also hat das Ziel mit der Priorität 0 immer die höchste Gewichtung bei der Entscheidung, was die KI machen möchte oder soll.

Kleiner Exkurs – bei uns im Beispiel sind die Prioritäten 0, 1 und 2 belegt mit den grundlegenden Prioritäten, die jede Entität benötigt, um zu überleben in der Oberwelt in Minecraft. Diese wären hier einmal das Ziel zu schwimmen mit Priorität 0, um zu verhindern, dass die Entität in Wasser springt und dann einfach untergeht und stirbt. Auf der Priorität 1 haben wir das Ziel, vor Gefahr wegzulaufen, also falls sich die Entität in Flammen befindet, zur nächsten Wasserstelle zu laufen, sollte es in Puderschnee feststecken zu fliehen, oder sollte es angegriffen werden, möglichst zu versuchen vor dem Angreifer wegzulaufen. Die Priorität 2 wird dann von der Fortpflanzung belegt, also dem Ziel, sollte die Entität mit dem richtigen Nahrungsitem gefüttert werden, mit einem passenden Partner ein Kind zu zeugen.

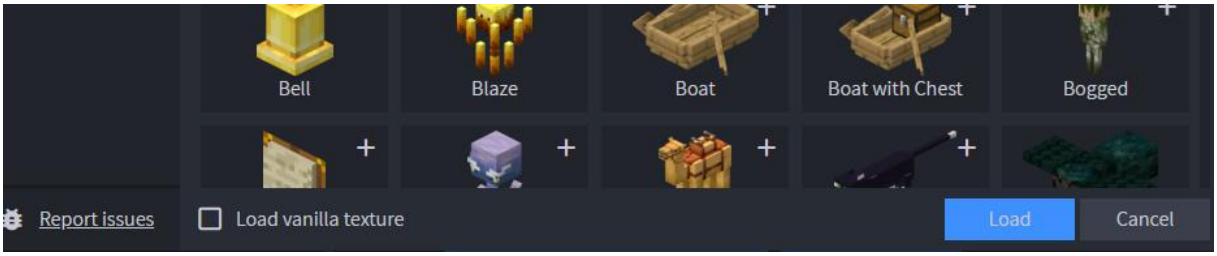
Eine Liste aller Ziele und Verhaltens, die wir für euch vereinfacht zur Verfügung stellen, findet ihr in der Dokumentation mit dem Namen „Ziele“. Darin sind alle Ziele, ihre Codezeilen zum Kopieren oder Abschreiben, und die Funktionsweisen erklärt und aufgeführt.

Solltet ihr weitere Ziele oder Verhalten euren Entitäten aneignen wollen, könnt ihr – falls ihr euch damit auskennt und damit klarkommt – den Quellcode von den weiteren in Minecraft vorhandenen Zielen anschauen und diese einbauen. Dabei müsst ihr nur den Code völlig anders schreiben, und geht nicht davon aus, dass eure Teamer das dann für euch übernehmen!

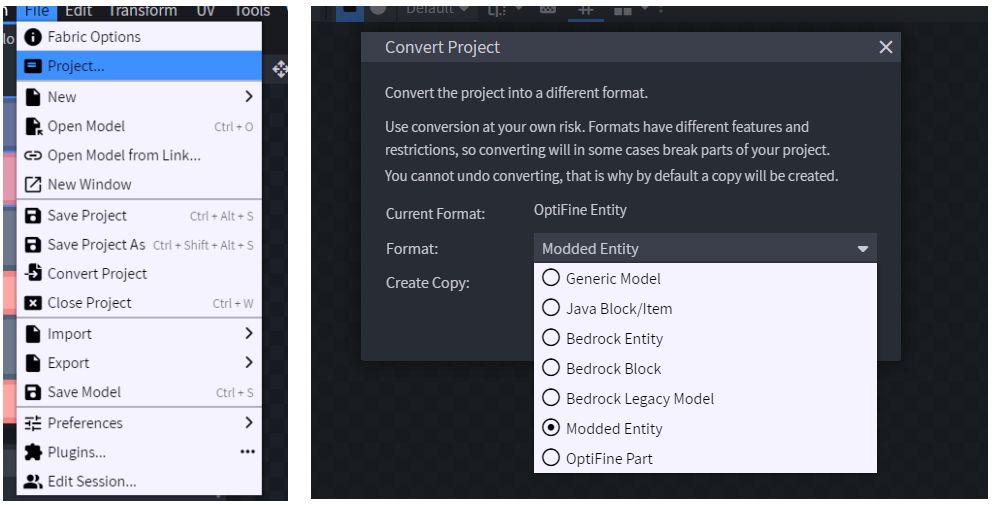
Kapitel 2.3 – die eigene Entität

Gehen wir nun zurück zum Erstellen der ersten eigenen Entität. Ab hier wollen wir sorgfältig arbeiten, damit wir nichts überstürzen und womöglich noch Fehler einbauen, oder etwas aus dem Blick verlieren. Dafür schnappen wir uns einmal ein Blatt Papier, einen Stift und unsere Kreativität, und überlegen uns mit Hilfe der Vorlagen aus der CEM Template Liste einmal, welche Kreatur wir erschaffen wollen, wie sie heißen soll, was ihr Verhalten ausmachen soll, und welches Farbschema wir der Kreatur geben wollen. – Wie bereits erwähnt, überlegt euch gut, wie komplex ihr eure Kreatur gestalten wollt, und wieviel ihr euch selbst zutraut, denn eure Teamer werden das nicht für euch übernehmen, wenn ihr euch zu viel vorgenommen habt am Ende!

Wenn ihr euch gut Gedanken gemacht habt, begeben wir uns zurück zu Blockbench, und fangen mit der Gestaltung eurer eigenen Entität an. Dafür wählen wir den CEM Template Loader aus, wählt die Vorlage aus die ihr bearbeiten wollt, und öffnet diese – WICHTIG: Ohne die Vanilla Texturen mitzuladen (da gibt es ein kleines Kästchen, aus welchem wir den Haken entfernen).



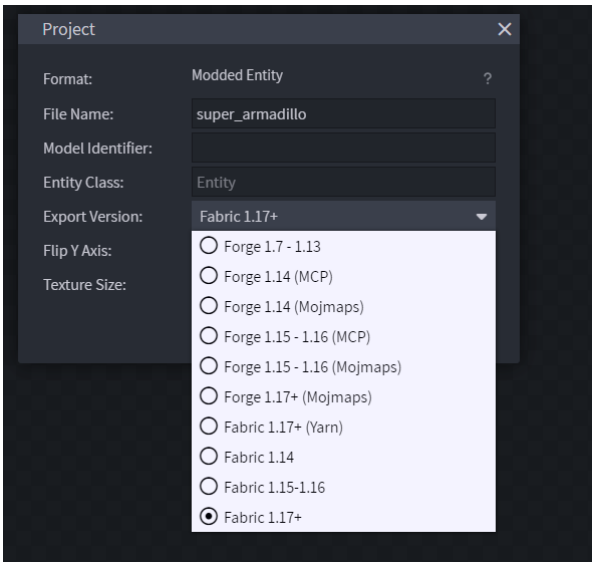
Sobald wir die Textur geladen haben – ich zeige es euch am Beispiel vom Amarillo (engl.: armadillo) – schauen wir uns einmal an, was ihr jetzt mit dem Modell machen könnt. Dabei wollen wir zunächst einmal die Einstellungen der Kreatur anpassen. Also öffnen wir unter „Datei“ (oder engl.: File) die Option „Konvertiere Projekt“ (oder engl.: Convert Project) aus.



Hier wählen wir einmal in dem Menü „Format“ das Format „Modded Entity“ aus, wie im Bild zu sehen. Den Haken bei „Create Copy“ können wir entfernen – wir wollen das Modell selbst ja anpassen. Als nächstes gehen wir wieder auf „Datei“ und wählen diesmal den Reiter „Project…“ aus, und ändern dort erst einmal den Namen zu dem angedachten Namen eurer Entität ab:

WICHTIG: Der Name eurer Entität sollte dem folgenden Format entsprechen, damit keine Fehler auftreten: Großgeschrieben, ohne Leerzeichen. Also am Beispiel vom Amarillo:

Superarmadillo



Dann wählen wir in dem Menü von der „Export Version“ die Version „Fabric 1.17+“ aus – WICHTIG: nicht die Version „Fabric 1.17+ (Mojmaps)“, das ist eine völlig andere Version, mit der wir nicht arbeiten können!!

Sobald wir diese Schritte gemacht haben, können wir nun zum eigentlichen Spaß übergehen – dem Erstellen der eigenen Textur bzw. Anpassung des Modells.

Dieser Schritt unterteilt sich noch einmal in zwei Möglichkeiten:

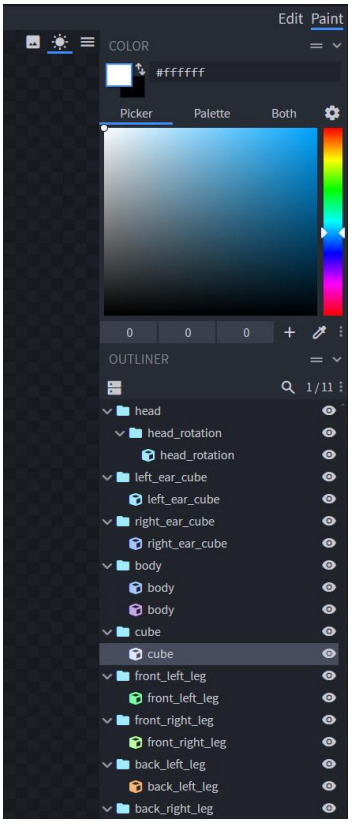
Wenn ihr es euch einfach machen wollt, behaltet ihr einfach die Form und das Modell von der Vorlage bei, und erstellt einfach nur eine neue Textur – also das Aussehen des Modells im Spiel, das Farbschema.

Alternativ könnt ihr auch das vorhandene Modell etwas verändern, dafür hilft es, wenn ihr euch mit Blokbench etwas auskennt, allerdings könnt ihr auch einfach etwas herumprobieren.

Fangen wir einmal mit der Textur an – denn das müsst ihr so oder so alle machen.

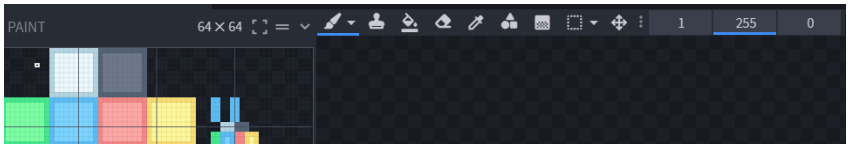
Kapitel 2.4 – die Textur

Um die Textur von eurer eigenen Entität nun anzupassen, gehen wir auf der ganz rechten Seite des Bildschirms in den „Paint“ Modus:

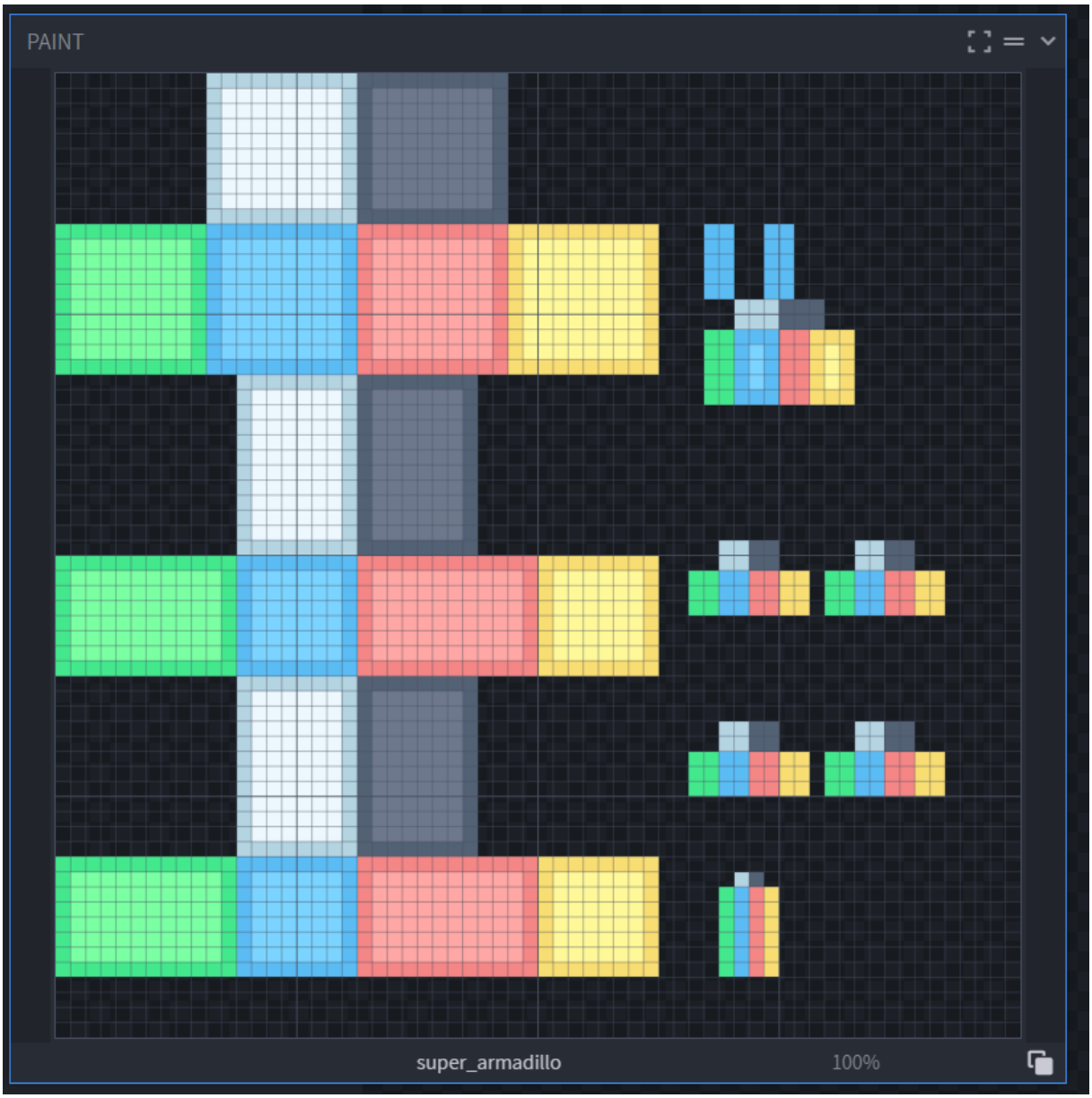


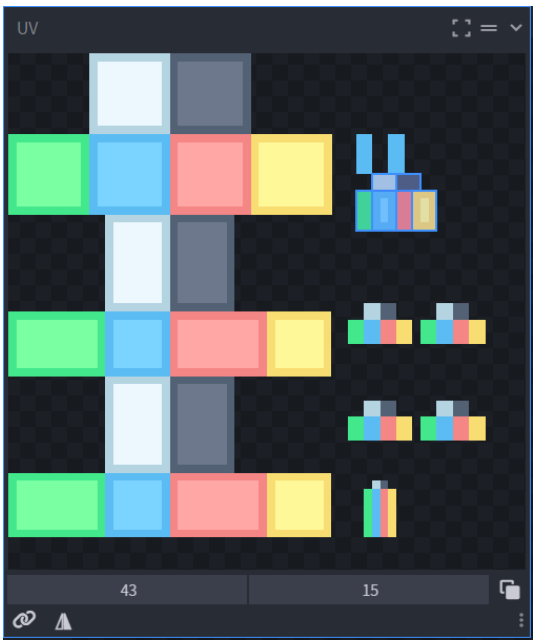
Nun solltet ihr bereits sehen, dass sich über das Modell ein feines Raster gelegt hat, die einzelnen Pixel der Textur. Diese könnt ihr nun einzeln mit einer neuen Farbe belegen, dafür wählt ihr die Farbe einfach rechts an der Seite aus, und klickt anschließend auf den Pixel auf dem Modell, welchen ihr in der Farbe einfärben wollt.

Oben habt ihr auch eine Leist mit weiteren Werkzeugen, zum Bearbeiten:



Wenn ihr auf das kleine Symbol, rechts neben dem „64 X 64“ klickt, könnt ihr auch dort direkt die Textur bearbeiten:





In dieser Ansicht lassen sich die einzelnen Oberflächen von der Entität einfacher bemalen, wenn man sonst nicht gut an die Oberflächen herankommen würde. Falls ihr wissen wollt, welcher Bereich der 2D-Textur nun zu welcher Oberfläche der 3D-Textur gehört, könnt ihr einfach rechts auf der Seite zurück in den „Edit“ Modus wechseln, und die einzelnen Oberflächen des Modells auswählen und in dem Textur Bildschirm schauen, welche Bereiche der 2D-Textur blau umrandet werden.

Hier habe ich zum Beispiel den Kopf des Amarillos ausgewählt, und im „UV“ Bildschirm sehen wir den kleinen Bereich rechts oben blau umrandet. Sollte ich diesen Bereich also im Texturen Bildschirm umfärben, so färbe ich den Kopf des Amarillo neu ein.

Aufgabe 2

Nehmt euch heute die restliche Planung eurer Entität vor, erstellt eine Textur, mit der ihr zufrieden seid, und überlegt euch schon einmal grob, welche Ziele wohl für eure Kreatur wichtig sein könnten. Dabei könnt ihr gerne schonmal einen Blick in die Doku zu den Zielen werfen.

In den nächsten Tagen beschäftigen wir uns dann mit der Einbindung der Entität in Minecraft, also den Code und die KI der Entität, als auch dass wir noch Animationen für die Entitäten erschaffen, wie eine Idle und eine Lauf Animation, jedoch sind euch hier natürlich keine Grenzen gesetzt und ihr könnt noch weitere Animationen erstellen, wenn ihr dafür Zeit habt.