

# CAMP MINECRAFT MODS

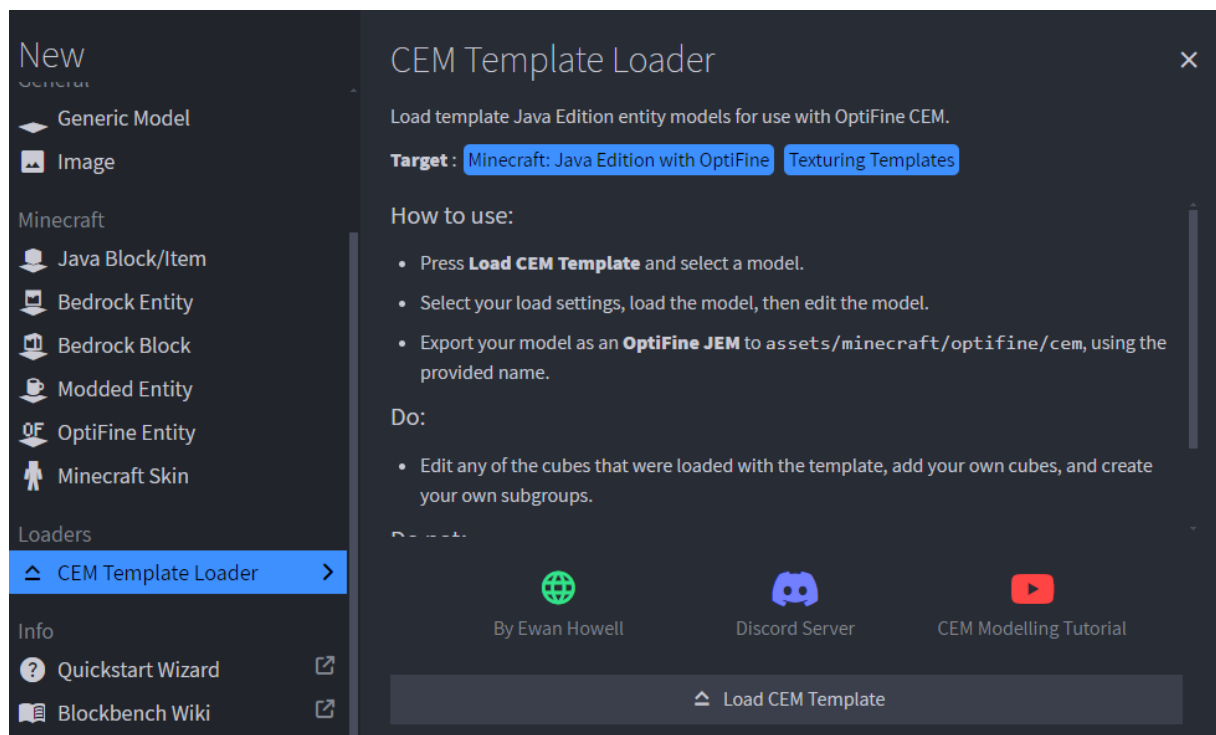
– Rüstungen –

Die Textur

Um eine Custom Rüstung in Minecraft zu erstellen, benötigen wir eine 3D-Textur. Diese erstellen wir in Blockbench.

Dafür öffnen wir einmal das Programm, und suchen nach der Schaltfläche

CEM Template Loader



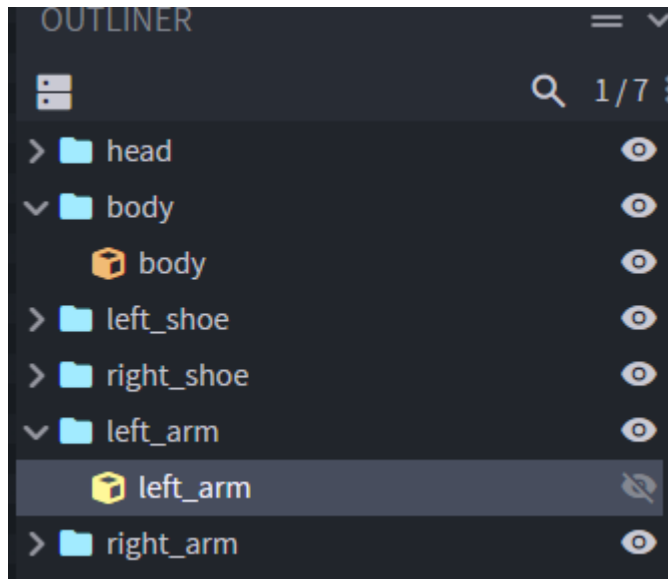
Und klicken dort auf „Load CEM Template“. In dem Fenster wählen wir dann unter „unsupported“ das „Armor“ set aus, wählen zuerst „Armor 1“ und öffnen das Modell, mit der Vanilla Textur!



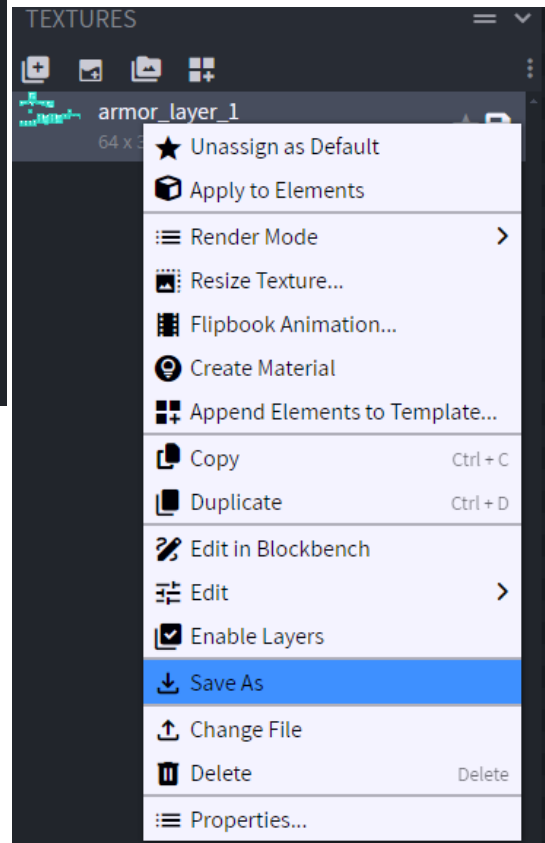
Im Folgenden müssen wir das Modell ein wenig bearbeiten, indem wir oben rechts in der Ecke auf „Paint“ klicken, und so die Textur der Rüstung einfach wie in Paint.net bearbeiten können.

Um an einige stellen besser heran zu kommen, könnt ihr auch einzelne Blöcke des Modells ausblenden, dafür könnt ihr

einfach auf das kleine Augen Symbol neben den Blöcken im Outliner klicken.



Wenn ihr mit der Textur zufrieden seid, exportieren wir die Textur Datei, indem wir auf diese Rechtsklicken, und „speichern unter“ auswählen:



Die Textur speichern wir dann, in unserem resources Ordner, aber diesmal unter textures → models → armor

Mit dem Namen unserer Rüstung kleingeschrieben, gefolgt von „\_layer\_1“ bzw. „\_layer\_2“ für die beiden Layer Dateien. In meinem Beispiel wäre das jetzt

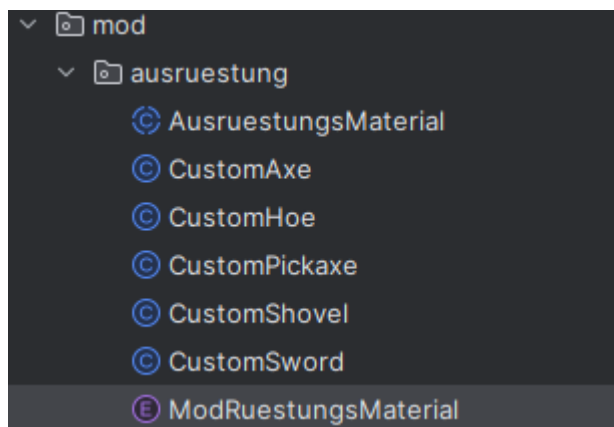
sapphire\_layer\_1

sapphire\_layer\_2

Wenn ihr beide Texturen fertig erstellt und exportiert habt, wollen wir die Rüstung auch in Minecraft noch registrieren.

Registrierung

Um die Rüstung zu registrieren, müssen wir zunächst ein Rüstungsmaterial festlegen. Dafür öffnen wir die Klasse „ModRuestungsMaterial“ in dem Ordner „ausruestung“:



und kopieren uns hier einmal die Zeilen vom RUBY, setzen hinter die letzte Klammer ein Komma, und fügen die kopierte Zeile wieder ein. Dann ändern wir alle Namen zu unseren Namen von unserem Material. Bei mir also zu

SAPPHIRE

Und

sapphire

```

public enum ModRuestungsMaterial implements ArmorMaterial { 6 usages

    RUBY( name: "ruby", durabilityMultiplier: 25, new int[] {3, 8, 6, 3}, enchantability: 19, 4 usages
        SoundEvents.ITEM_ARMOR_EQUIP_DIAMOND, toughness: 2f, knockbackResistance: 0.2f, ()-> Ingredient.ofItems(ModItems.GetItem( name: "ruby")))

    ;

    RUBY( name: "ruby", durabilityMultiplier: 25, new int[] {3, 8, 6, 3}, enchantability: 19, 4 usages
        SoundEvents.ITEM_ARMOR_EQUIP_DIAMOND, toughness: 2f, knockbackResistance: 0.2f, ()-> Ingredient.ofItems(ModItems.GetItem( name: "ruby"))),
    SAPHIRE( name: "saphire", durabilityMultiplier: 25, new int[] {3, 8, 6, 3}, enchantability: 19, no usages
        SoundEvents.ITEM_ARMOR_EQUIP_DIAMOND, toughness: 2f, knockbackResistance: 0.2f, ()-> Ingredient.ofItems(ModItems.GetItem( name: "saphire")))

    ;

```

im Anschluss dazu, gehen wir in die TestMod-Klasse und kopieren uns dort die Zeilen Code von der Rubin Rüstung, und ersetzen alles mit unserem eigenen Material:

```

//Test Custom Rüstung
Mod.RuestungHinzufuegen( name: "ruby_helmet", ModRuestungsMaterial.RUBY, ArmorItem.Type.HELMET);
Mod.RuestungHinzufuegen( name: "ruby_chestplate", ModRuestungsMaterial.RUBY, ArmorItem.Type.CHESTPLATE);
Mod.RuestungHinzufuegen( name: "ruby_leggings", ModRuestungsMaterial.RUBY, ArmorItem.Type.LEGGINGS);
Mod.RuestungHinzufuegen( name: "ruby_boots", ModRuestungsMaterial.RUBY, ArmorItem.Type.BOOTS);

```

In der „ModRuestungsMaterial“-Klasse könnt ihr auch die Attribute der Rüstung anpassen – also einmal den Haltbarkeitswert, die Widerstandswerte-Multiplikatoren für die einzelnen Teile der Rüstung, die Verzauberkeit der Rüstung, das Geräusch der Rüstung, den Härtewert sowie der Rückstoßwiderstand und die Items, mit denen man die Rüstung reparieren kann an einem Amboss:

```

durabilityMultiplier: 25, new int[] {3, 8, 6, 3}, enchantability: 19, 4 usages
ITEM_ARMOR_EQUIP_DIAMOND, toughness: 2f, knockbackResistance: 0.2f, ()-> Ingredient.ofItems(ModItems.GetItem( name: "ruby")))

```

probiert mit den Werten am besten ein wenig herum, und falls ihr mit der Textur nicht zufrieden seid, könnt ihr jederzeit einfach die Textur anpassen, und neu in den resources Ordner exportieren.

Um die Rüstung jetzt auch korrekt im Spiel angezeigt zu bekommen, müssen wir die Model Klasse noch erstellen lassen, dafür gehen wir in den datagen-Ordner, und suchen die „ModModelsProvider“-Klasse, in welcher wir einfach den Code von der Ruby Rüstung kopieren und anpassen:

```

//Rüstungen - siehe Doku [Rüstungen]
itemModelGenerator.registerArmor((ArmorItem) ModItems.GetItem( name: "ruby_helmet"));
itemModelGenerator.registerArmor((ArmorItem) ModItems.GetItem( name: "ruby_chestplate"));
itemModelGenerator.registerArmor((ArmorItem) ModItems.GetItem( name: "ruby_leggings"));
itemModelGenerator.registerArmor((ArmorItem) ModItems.GetItem( name: "ruby_boots"));

```

Wenn ihr die Schritte alle befolgt habt, und alles richtig geschrieben, solltet ihr nun, nachdem ihr einmal die Datagen ausgeführt habt, die Rüstung auch in Minecraft zu finden sein!