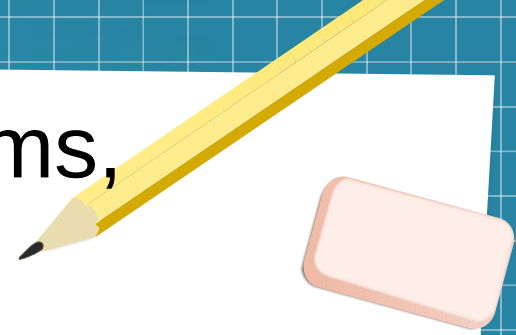




Armor

Rüstungen

Rüstungen sind besondere Items,



- die man tragen kann
- die beim Kampf einen Einfluss haben (Schutz)
- die man nur 1x in einem Itemstack haben sollte
- die eine zusätzliche Textur für den Spieler benötigen
- die zusätzlich für den Avatar registriert werden müssen
- die mit der Zeit kaputt gehen (Abnutzung)
- die man erweitern kann
 - machen wir hier nicht, ist zu komplex

Armor-Typen für menschlichen Avatar

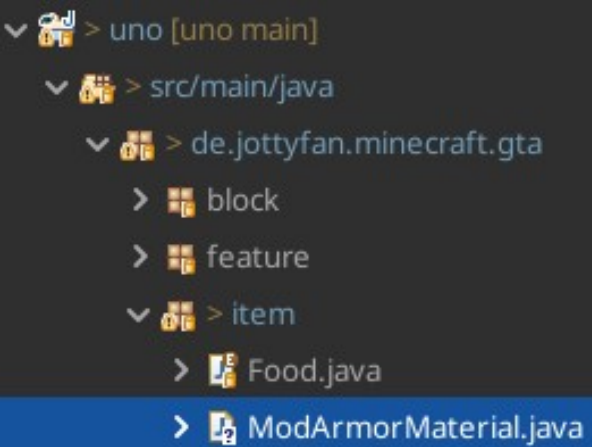
- Helmet
- Chestplate
- Leggings
- Boots



← Item

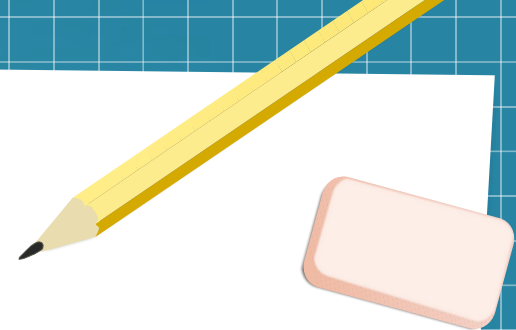
Equipment →





Armor material

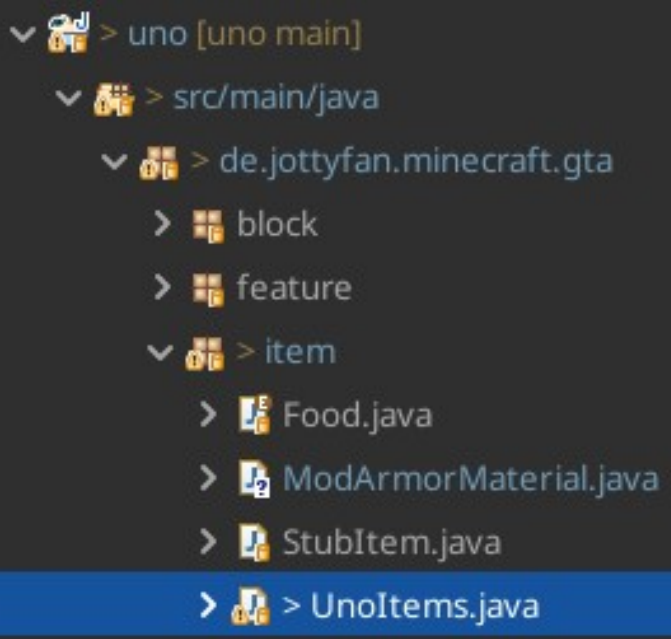
Rüstungseigenschaften definieren



```
public class ModArmorMaterial {
    public static ResourceKey<? extends Registry<EquipmentAsset>> REGISTRY_KEY = ResourceKey
        .createRegistryKey(Identifier.withDefaultNamespace("equipment_asset"));

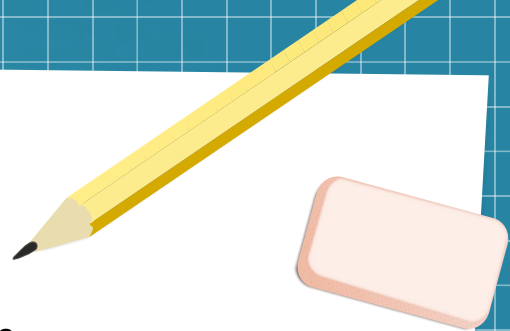
    public static final ResourceKey<EquipmentAsset> RUBY_KEY = ResourceKey.create(REGISTRY_KEY,
        Identifier.fromNamespaceAndPath(Unomod.MOD_ID, "ruby"));

    public static final ArmorMaterial RUBY_ARMOR_MATERIAL = new ArmorMaterial(500,
        Util.make(new EnumMap<>(ArmorType.class), map -> {
            map.put(ArmorType.BOOTS, 3);
            map.put(ArmorType.LEGGINGS, 4);
            map.put(ArmorType.CHESTPLATE, 10);
            map.put(ArmorType.HELMET, 3);
            map.put(ArmorType.BODY, 5);
        }), 20, SoundEvents.ARMOR_EQUIP_DIAMOND, 0, 0, null, RUBY_KEY);
}
```



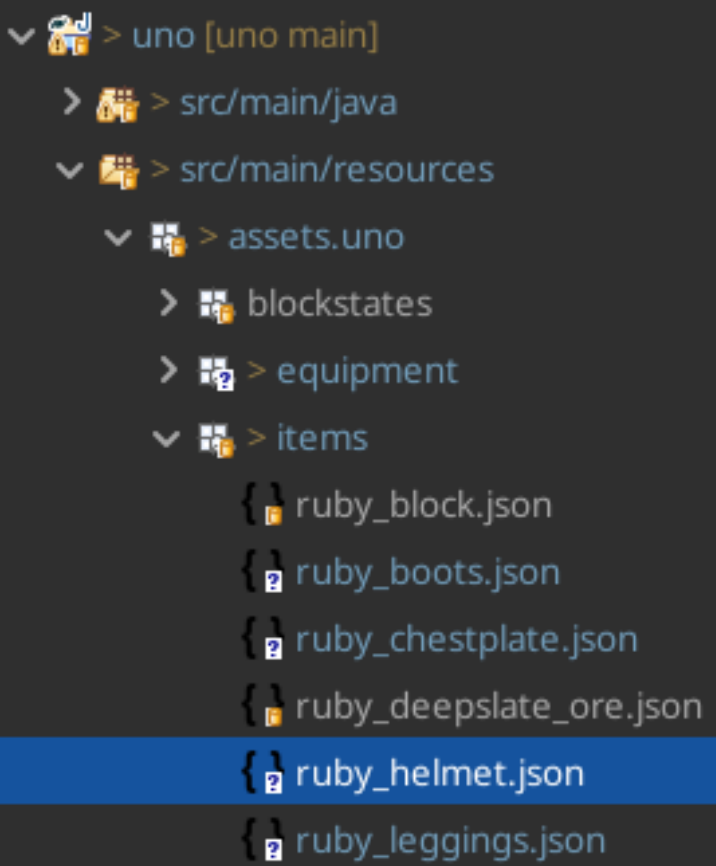
Items anlegen

Neue Methode **registerArmorItem**, um allgemeine Rüstungseigenschaften zu definieren

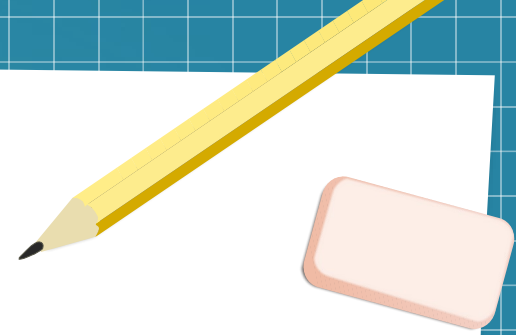


```
private static final Item RUBY_BOOTS = registerArmorItem("ruby_boots", ArmorType.BOOTS);
private static final Item RUBY_HELMET = registerArmorItem("ruby_helmet", ArmorType.HELMET);
private static final Item RUBY_CHESTPLATE = registerArmorItem("ruby_chestplate", ArmorType.CHESTPLATE);
private static final Item RUBY_LEGGINGS = registerArmorItem("ruby_leggings", ArmorType.LEGGINGS);

private static final Item registerArmorItem(String name, ArmorType type) {
    Item.Properties properties = new Item.Properties()
        .humanoidArmor(ModArmorMaterial.RUBY_ARMOR_MATERIAL, type);
    return registerItem(name, properties);
}
```



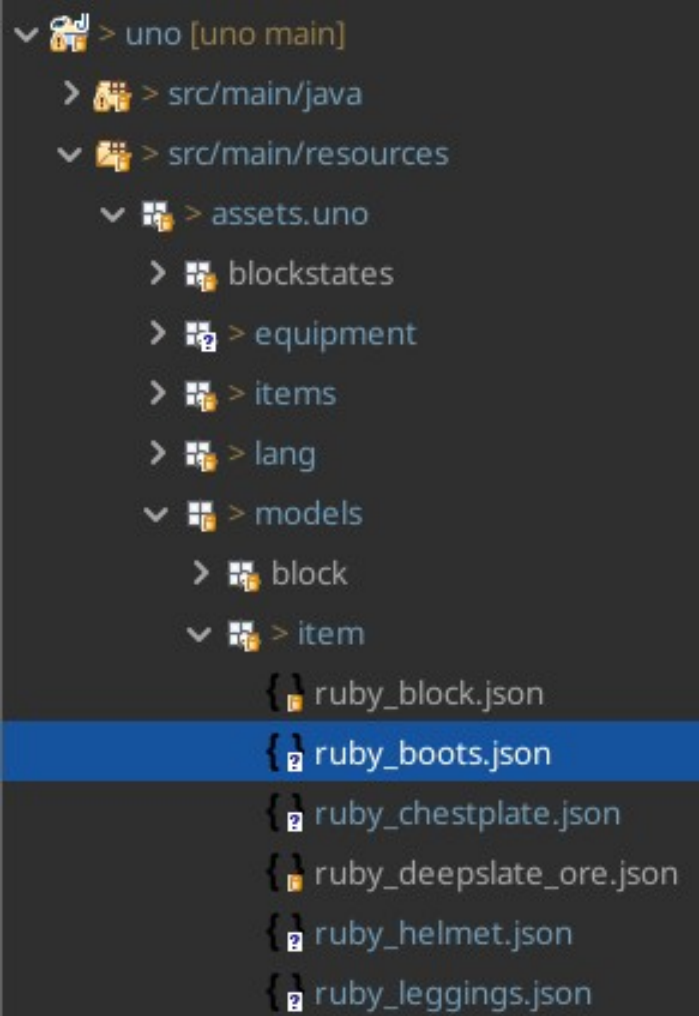
Assets für items



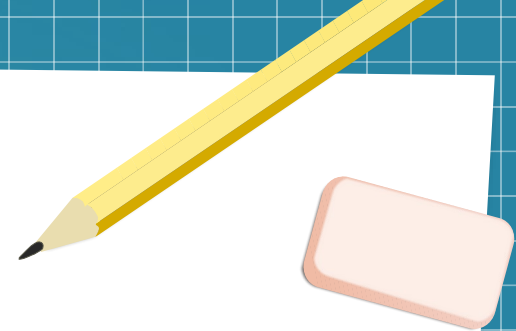
Immer alles für:

helmet, chestplate,
leggings, boots

```
{  
  "model": {  
    "type": "minecraft:model",  
    "model": "uno:item/ruby_helmet"  
  }  
}
```



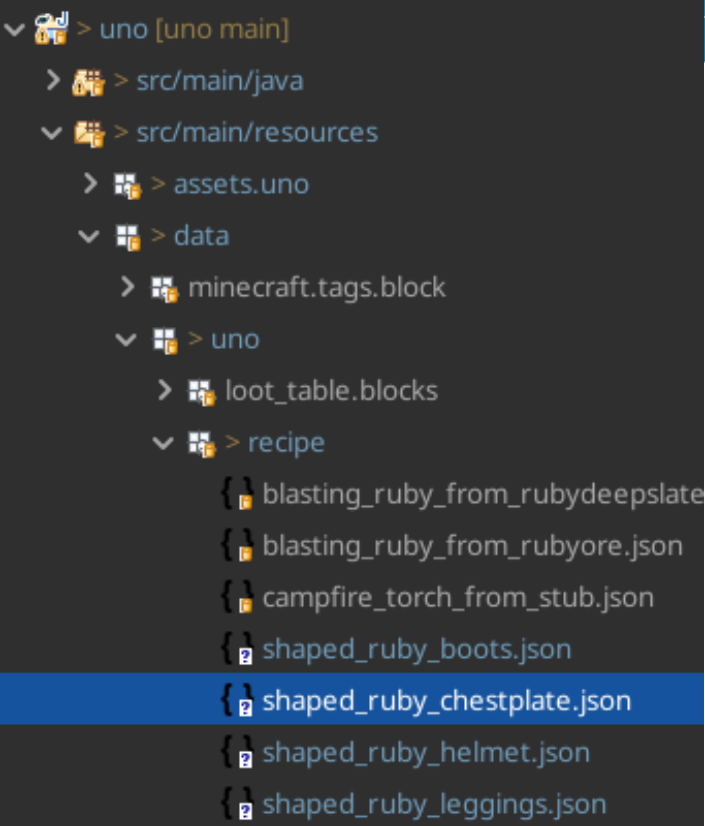
Assets models



Immer alles für:

helmet, chestplate,
leggings, boots

```
{  
  "parent": "item/coal",  
  "textures": {  
    "layer0": "uno:item/ruby_boots"  
  }  
}
```




Immer alles für:

helmet, chestplate,
leggings, boots

Crafting-Rezepte

```
{  
  "type": "minecraft:crafting_shaped",  
  "pattern": [  
    "t t",  
    "ttt",  
    "ttt"  
  ],  
  "key": {  
    "t": "uno:ruby"  
  },  
  "result": {  
    "id": "uno:ruby_chestplate",  
    "count": 1  
  }  
}
```

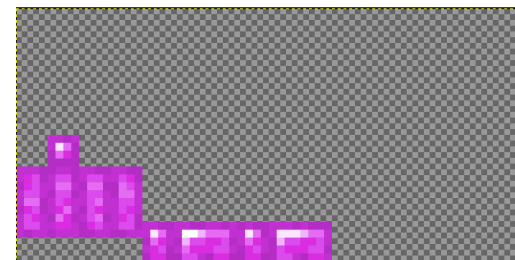
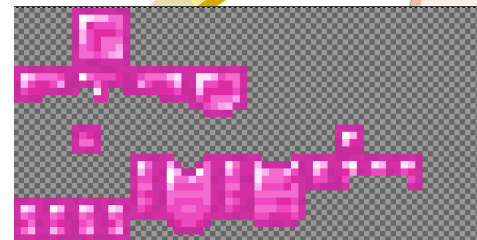

▼  > uno [uno main]
 >  > src/main/java
▼  > src/main/resources
 ▼  > assets.uno
 >  blockstates
 ▼  > equipment
 {  } ruby.json

Equipment registration

```
{  
  "layers": {  
    "humanoid": [  
      {  
        "texture": "uno:ruby"  
      }  
    ],  
    "humanoid_leggings": [  
      {  
        "texture": "uno:ruby"  
      }  
    ]  
  }  
}
```

Texturen

- textures/entity/equipment/humanoid/ruby.png
- textures/entity/equipment/humanoid_leggings/ruby.png
- textures/item



ruby_boots
.png

ruby_chest
plate.png

ruby_helm
et.png

ruby_leggi
ngs.png

Übersetzungen nicht vergessen

