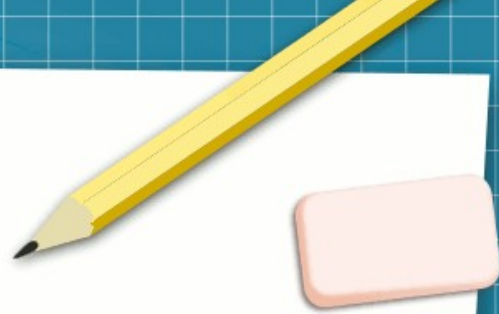




Unobfuscated Food

Einführung von Essen

Enums



- Abzählbare Typen ohne Ordnung
 - Farben von Gummibärchen
 - Lego-Bausteine
 - Vogelarten
- Der Name allein definiert den enum (eindeutig)

```
public enum EnumColor {  
    RED,  
    YELLOW,  
    GREEN  
}
```

```
uno [uno main]
src/main/java
de.jottifyan.minecraft.gta
  block
  feature
  item
    Food.java
```

Eigenes Essen

- Neues Essen, das $\frac{1}{2}$ Hunger stillt
- Definition der Essens-Eigenschaften als Enum
 - Enum erweitern, damit er zusätzlich zum Namen Essens-Attribute hat
- Unter Paket items

```
public enum Food {
    DELICIOUS_STUB(new FoodProperties.Builder()
        .nutrition(1).saturationModifier(0.1f)
        .build());

    private final FoodProperties component;

    private Food(FoodProperties component) {
        this.component = component;
    }

    public FoodProperties get() {
        return component;
    }
}
```

Stub um Essenseigenschaft erweitern



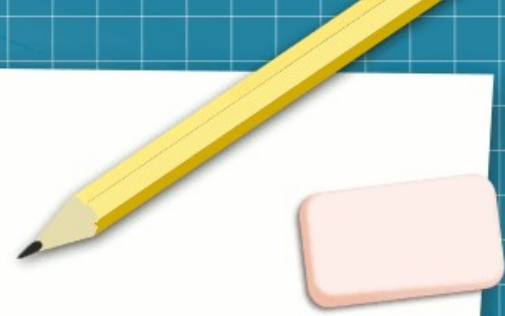
- Erweiterte Sessings mit **food** beim Definieren
 - In UnolItems:

```
private static final Item registerStubItem(Identifier identifier) {  
    return registerItem(identifier, new Item.Properties()  
        .food(Food.DELICIOUS_STUB.get()), StubItem::new);  
}
```

- Food.DELICIOUS_STUB.get() gibt definierte FoodProperties

Arbeit sichern

- Git commit und push



Herausforderung

- Rezept für Pommes
 - Rezeptkonflikt → geschnittene Kartoffel
 - Icon abmalen
 - Icon für geschnittene Kartoffel malen
 - Icons anlegen
 - Backrezept anlegen
- Ausprobieren und schmecken lassen





This work is licensed under a Creative Commons
Attribution-ShareAlike 3.0 Unported License.
It makes use of the works of Mateus Machado Luna.

