



Blocks and Recipes

Einführung von eigenen Blocks

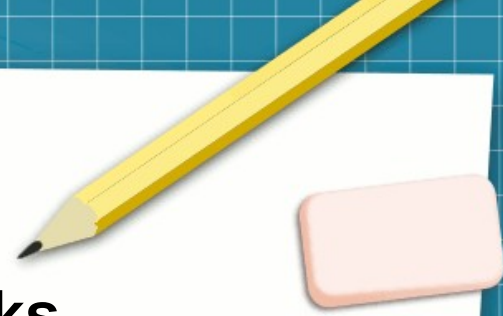
Block Definition und Instanzen

- Einmal definieren, immer wieder benutzen
 - Blöcke sind eindeutig definierte Objekte in Minecraft, gibts nur einmal
 - Instanzen (Kopien) davon gibt es jede Menge in der Minecraft-Welt

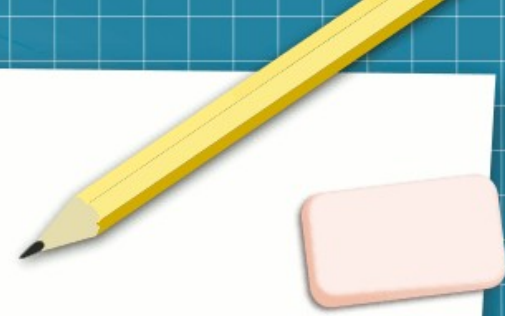


Mein erster eigener Block

- Neues Paket **block**, darin neue Klasse **ModBlocks**
 - Block anlegen
 - Block registrieren
 - Registrierung einbinden



Block anlegen



- BlockSettings – definieren Block-Eigenschaften
- In de.jottyfan.minecraft.**block**.ModBlocks

```
public static final Block RUBY_BLOCK = registerBlock(  
    Identifier.of(Gtamfmd.MOD_ID, "ruby_block"),  
    AbstractBlock.Settings.create().strength(4f)  
        .requiresTool().sounds(BlockSoundGroup.AMETHYST_BLOCK)  
);
```


Block registrieren

- Methode **registerBlock** schreiben
 - Block als **Block-Definition** und als **BlockItem**
 - In `de.jottyan.minecraft.block.ModBlocks`



```
private static Block registerBlock(Identifier identifier,
                                   Block.Settings settings) {
    Block block = new Block(settings.registryKey(
        RegistryKey.of(RegistryKeys.BLOCK, identifier)));
    registerBlockItem(identifier, block, new Item.Settings());
    return Registry.register(Registries.BLOCK, identifier, block);
}
```

Block und Blockitem registrieren



- BlockItem registrieren
 - In de.jottifyan.minecraft.**block**.ModBlocks

```
private static void registerBlockItem(Identifier identifier,
                                     Block block,
                                     Item.Settings settings) {
    Registry.register(Registries.ITEM, identifier,
        new BlockItem(block, settings.useItemPrefixedTranslationKey()
            .registryKey(RegistryKey.of(RegistryKeys.ITEM, identifier))));
}
```

Blöcke sichtbar machen



- Blöcke auch im Kreativ-Modus in der Blockübersicht sehen
 - In `de.jottifyfan.minecraft.block.ModBlocks`

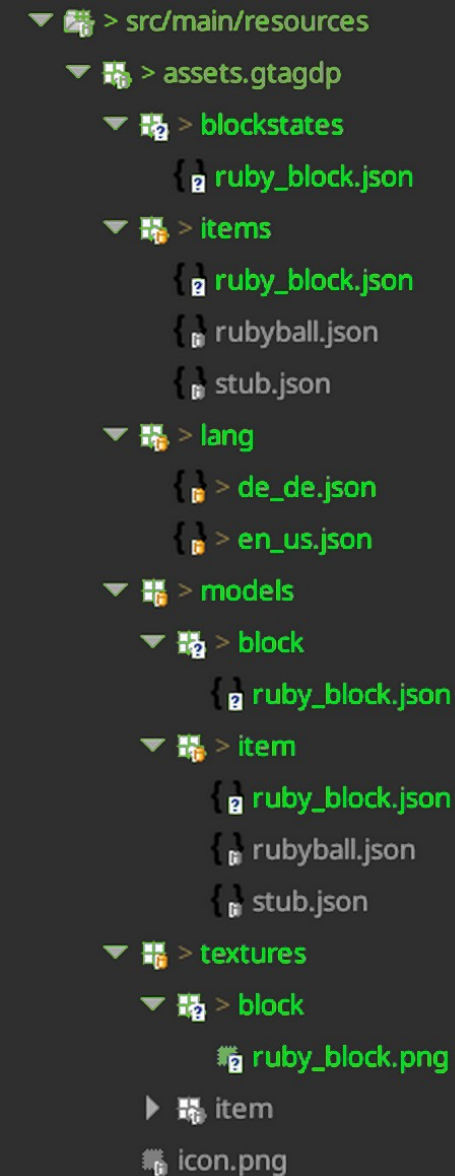
```
public static void registerModBlocks() {  
    Gtamfmd.LOGGER.info("Registering Mod Blocks for {}", Gtamfmd.MOD_ID);  
    ItemGroupEvents.modifyEntriesEvent(ItemGroups.BUILDING_BLOCKS)  
        .register(entries -> {  
            entries.add(RUBY_BLOCK);  
        });  
}
```

Registrierung bei Minecraft



- Registrierung beim Starten von Minecraft aktivieren
 - In de.jottyan.minecraft.Gtamfmd

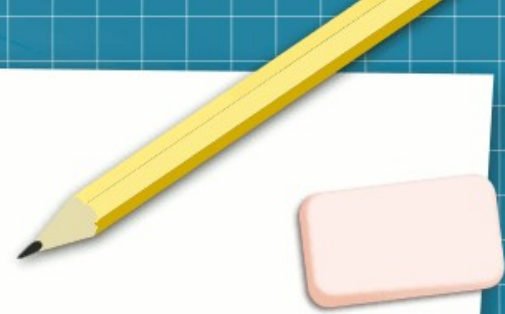
```
public class Gtamfmd implements ModInitializer {  
    public static final String MOD_ID = "gtamfmd";  
    public static final Logger LOGGER = LoggerFactory.getLogger(MOD_ID);  
  
    @Override  
    public void onInitialize() {  
        ModItems.registerModItems();  
        ModBlocks.registerModBlocks();  
    }  
}
```

JSON-Dateien und Bilder

- Blockstates
 - Aktueller Anzeigezustand eines Blocks
- Items
 - Damit Blöcke in der Hand sichtbar werden
- Übersetzung
- Models
 - Damit Blöcke in der Welt sichtbar werden
- Texturen
 - Damit Blöcke schick aussehen

Blockstates



- In assets.gtamfmd.**blockstates**
 - Neue JSON-Datei **ruby_block.json**

```
{  
  "variants": {  
    "": {  
      "model": "gtamfmd:block/ruby_block"  
    }  
  }  
}
```

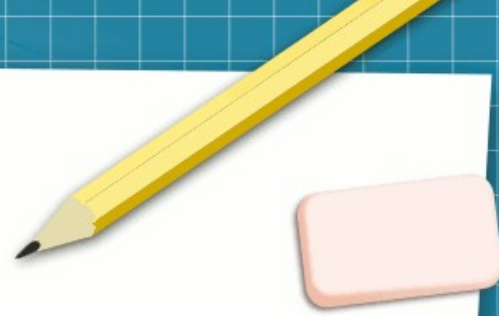
Items



- In `assets.gtamfmd.items`
 - Neue JSON-Datei `ruby_block.json`

```
{  
  "model": {  
    "type": "minecraft:model",  
    "model": "gtamfmd:block/ruby_block"  
  }  
}
```

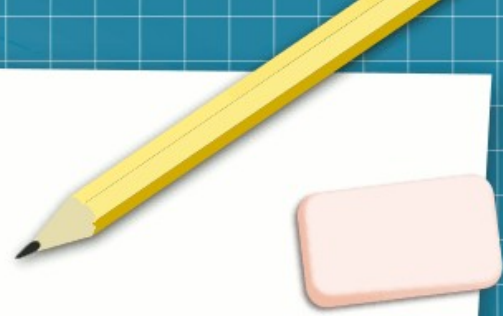
Models (Blocks)



- In assets.gtamfmd.**models.block**
 - Neue JSON-Datei **ruby_block.json**

```
{  
  "parent": "minecraft:block/cube_all",  
  "textures": {  
    "all": "gtamfmd:block/ruby_block"  
  }  
}
```

Models (Items)



- In `assets.gtagdp.models.item`
 - Neue JSON-Datei `ruby_block.json`

```
{  
  "parent": "gtamfmd:block/ruby_block"  
}
```


Übersetzungen

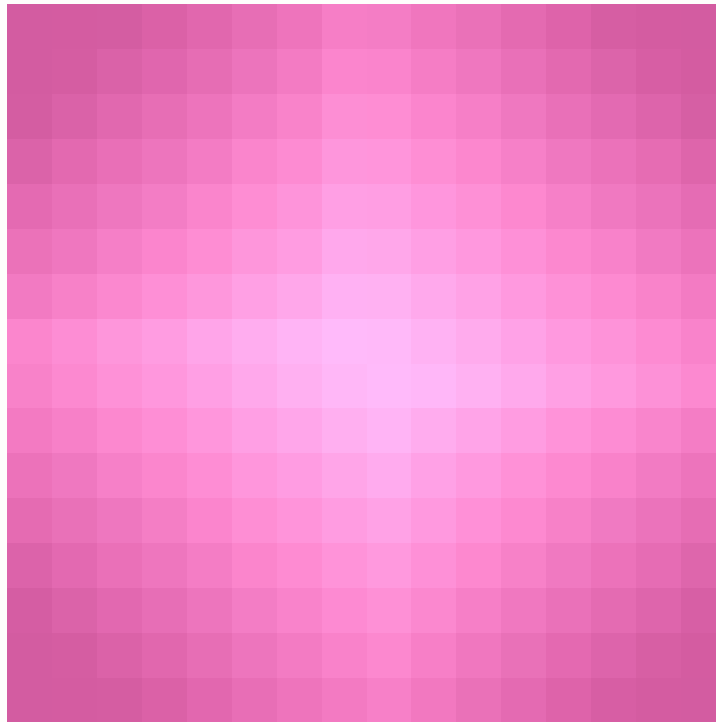
- In `assets.gtagdp.lang.de_de.json` bzw. `en_us.json`

```
{  
  "item.gtamfmd.stub": "Stummel",  
  "item.gtamfmd.ruby": "Rubin",  
  "item.gtamfmd.ruby_block": "Rubinblock"  
}
```

```
{  
  "item.gtamfmd.stub": "Stub",  
  "item.gtamfmd.ruby": "Ruby",  
  "item.gtamfmd.ruby_block": "Ruby block"  
}
```

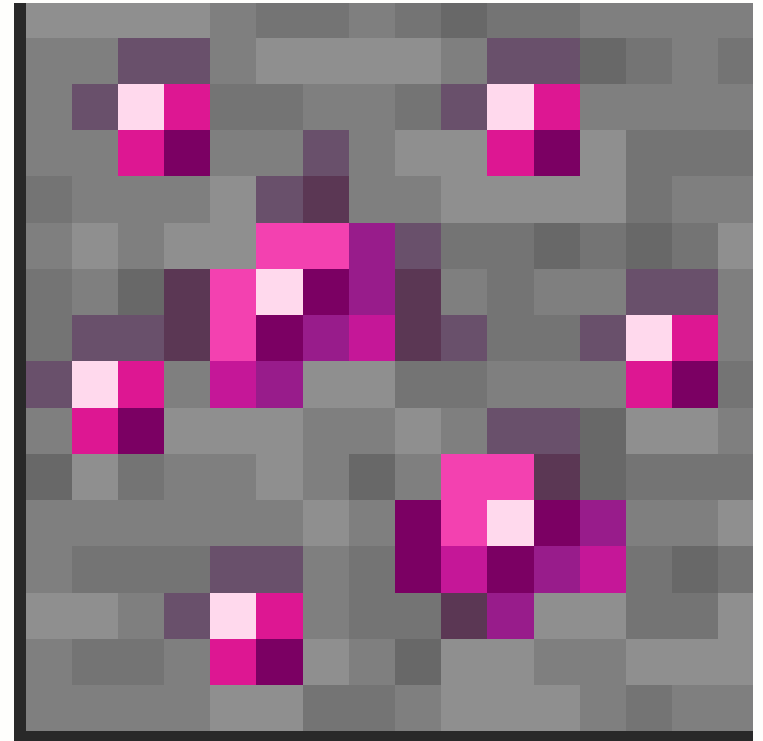
Textur anlegen

- assets/gtamfmd/textures/**block/ruby_block.png**



Erstelle einen 2. Block und binde ihn ein

- Git commit und push – erst dann mit neuen Features starten
- Namen überlegen (z.B. ruby_ore)
- Block-Bild malen
- In ModBlocks einbetten
- Testen
- Git commit und push





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

