



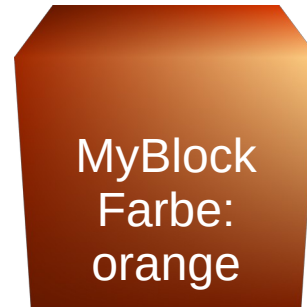
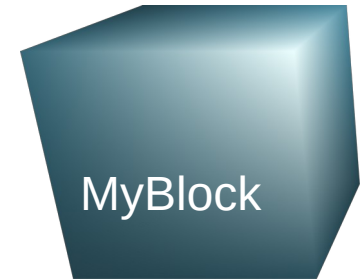
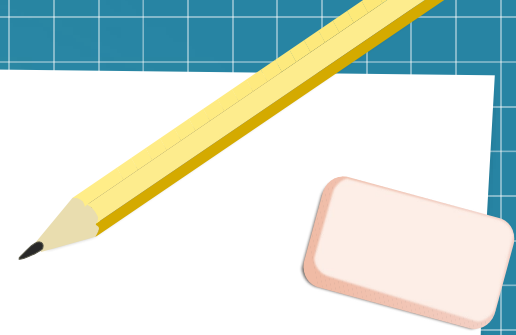
Block States and Item Stacks

Block States

Item Stacks

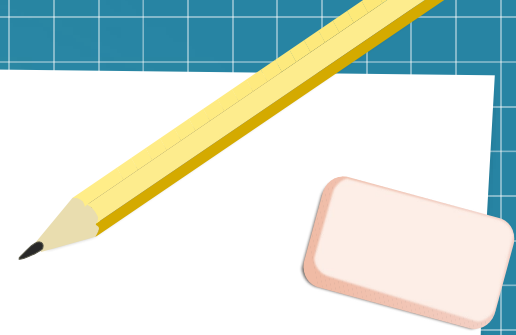
Block States

- Definiert die Eigenschaften einer Block-Instanz
- Ist eindeutig pro Instanz, wird also eher nicht geteilt
- Kann unterschiedliche Eigenschaften halten



Neuer RubyBlock

- Neue Klasse RubyBlock anlegen
- Property in Klasse definieren: ACTIVATED
- Property registrieren (appendProperties)
- Property benutzen in onUse



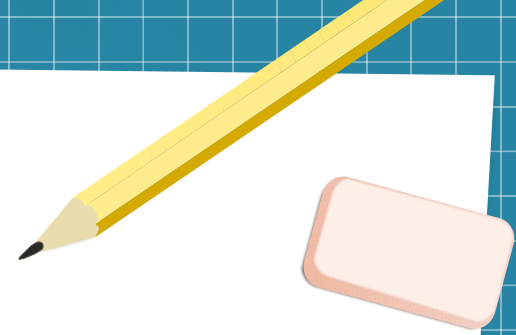
```
public class RubyBlock extends Block {
    public static final BooleanProperty ACTIVATED = BooleanProperty.of("activated");

    public RubyBlock(Settings settings) {
        super(settings);
        setDefaultState(getDefaultState().with(ACTIVATED, false));
    }

    @Override
    protected void appendProperties(Builder<Block, BlockState> builder) {
        builder.add(ACTIVATED);
    }

    @Override
    protected ActionResult onUse(BlockState state, World world, BlockPos pos, PlayerEntity player) {
        if (!world.isClient()) {
            world.setBlockState(pos, state.cycle(ACTIVATED));
        }
        return ActionResult.SUCCESS;
    }
}
```

RubyBlock registrieren



- Registrierung von RubyBlock anpassen
- Leuchtkraft je nach Property ACTIVATED einstellen

```
public class ModBlocks {

    public static final Block RUBY_BLOCK = registerRubyBlock(Identifier.of(GTAGDP.MOD_ID, "ruby_block"),
        AbstractBlock.Settings.create().strength(4f).requiresTool().sounds(BlockSoundGroup.AMETHYST_BLOCK)
        .luminance(state -> state.get(RubyBlock.ACTIVATED) ? 15 : 0));
    public static final Block RUBY_ORE = registerBlock(Identifier.of(GTAGDP.MOD_ID, "ruby_ore"),
        AbstractBlock.Settings.create().strength(3f).requiresTool());

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

Item Stacks

- Sammlung von gleichen Items mit gleichen Eigenschaften
- Kann in Hand gehalten oder Kisten gelegt werden
- Hat eine Stacksize, normalerweise 64

