

FABRIC 1.20 × 我的世界 MOD

第 8 课

让方块记住放置方向

HorizontalFacingBlock 与四向模型



本课成就 放置方块时自动面向玩家，并让同一个模型正确显示东、南、西、北四个朝向。

建议 120-140 分钟 | [simon-minecraft-template-1.20](#) | Java 21

01 本课任务：只增加一个新能力

接着第 7 课的 Property 思路，把整数状态扩展为方向状态

第 7 课让属性保存 0-3。方向也可以看成四个有限值：NORTH、EAST、SOUTH、WEST。我们让方块在放置时读取玩家朝向，再由 blockstates 旋转同一个模型。

本课三颗知识钻石

- 使用 HorizontalFacingBlock.FACING 保存四个水平方向。
- 在 getPlacementState 中读取玩家面向并决定方块正面。
- 用 blockstates 的 y 旋转复用同一个模型。

1

继承方向方块

2

设置默认朝向

3

读取玩家方向

4

加入状态表

5

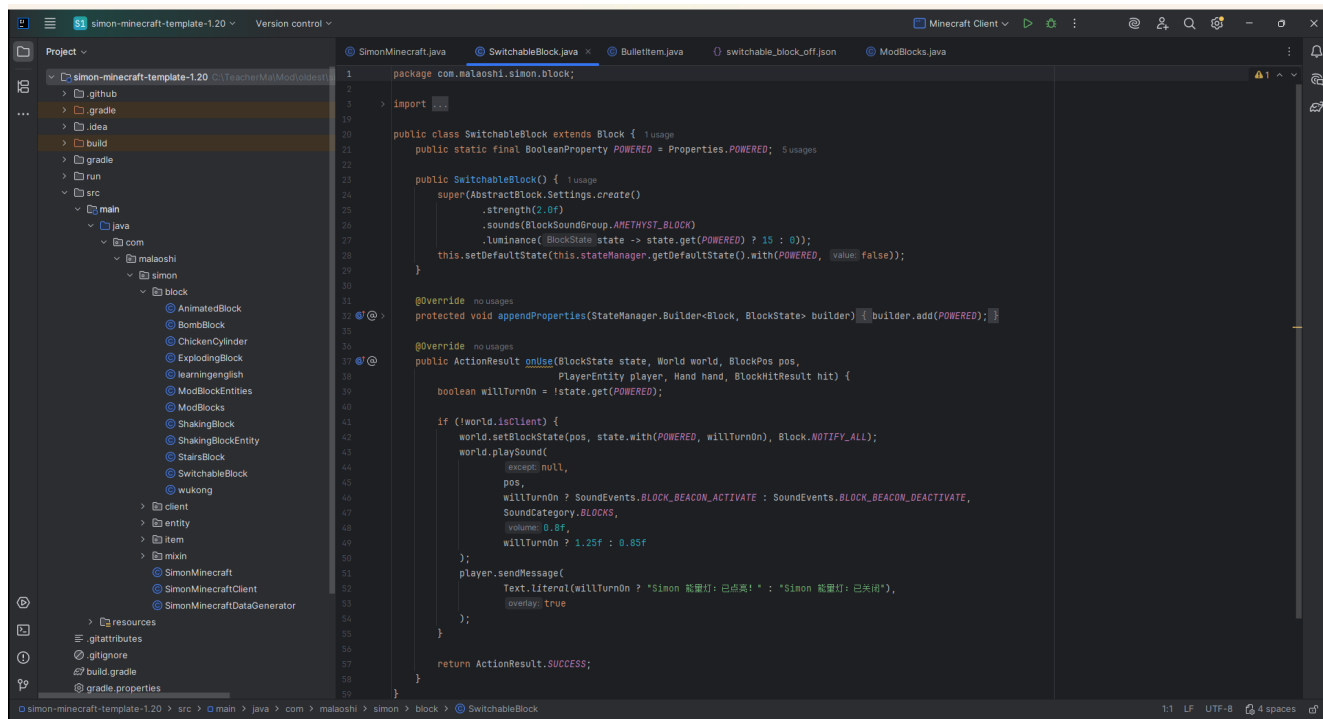
旋转模型

顺序感

先只处理水平方向，不加入 UP 和 DOWN。四个状态验证通过后，再考虑六向方块或与 level 组合。

02 在真实 IntelliJ 工程中定位文件

同一 simon-minecraft-template-1.20 工程界面，不是生成图



真实 IntelliJ 截图 · 左侧项目树 + Java 编辑器 + Gradle 工程

本课只打开这些位置

- src/main/java/com/malaoshi/simon/block/DirectionalMachineBlock.java
- src/main/java/com/malaoshi/simon/block/ModBlocks.java
- assets/simon-minecraft/blockstates/directional_machine.json
- assets/simon-minecraft/models/block/directional_machine.json
- assets/simon-minecraft/textures/block/directional_machine.png

版本锁定

Minecraft 1.20 · Yarn 1.20+build.1 · Fabric API 0.83.0+1.20。当前模板 fabric.mod.json 要求 Java 21。

03 定义方向属性并记录放置者朝向

先读懂，再逐行输入；不要整段复制后跳过理解

```
DirectionalMachineBlock.java

public class DirectionalMachineBlock
    extends HorizontalFacingBlock {

    public DirectionalMachineBlock(Settings settings) {
        super(settings);
        setDefaultState(getStateManager()
            .getDefaultState()
            .with(FACING, Direction.NORTH));
    }

    @Override
    public BlockState getPlacementState(
        ItemPlacementContext context) {
        return getDefaultState().with(
            FACING,
            context.getHorizontalPlayerFacing()
                .getOpposite());
    }

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

程序怎么读

FACING 已由 HorizontalFacingBlock 提供。getOpposite 让机器正面朝向放置它的玩家，而不是背对玩家。

04 同一个模型旋转成四个方向

Java 与资源文件必须使用同一个 MOD_ID 和注册名

```
blockstates/directional_machine.json

{
  "variants": {
    "facing=north": {
      "model": "simon-minecraft:block/directional_machine"
    },
    "facing=east": {
      "model": "simon-minecraft:block/directional_machine",
      "y": 90
    },
    "facing=south": {
      "model": "simon-minecraft:block/directional_machine",
      "y": 180
    },
    "facing=west": {
      "model": "simon-minecraft:block/directional_machine",
      "y": 270
    }
  }
}
```

连接点

模型文件只有一份，blockstates 根据 facing 添加 y 轴旋转。先确认模型原始正面朝北，再填写 90 度倍数。

05 从玩家视线到模型旋转

放置上下文 → DirectionProperty → BlockState → JSON 变体

玩家方向

getHorizontalPlayerFacing 读取玩家水平朝向。

取反

getOpposite 让方块正面朝向玩家。

BlockState

FACING 把方向保存在世界中的方块状态里。

blockstates

facing=north/east/south/west 选择旋转角度。

组合数量

若再加入 level=0-3，将出现 $4 \times 4 = 16$ 种状态组合。

课堂检查

模型看似转错 90 度时，先检查模型原始正面，而不是反复修改 Java。方向错误通常发生在资源旋转约定。

06 进入游戏验证结果

真实 Fabric 客户端截图，由 Minecraft ScreenshotRecorder 保存



游戏实拍 · 四个同类模型按 NORTH、EAST、SOUTH、WEST 四向放置

检查 1

从四个方向放置，机器正面始终朝向玩家。

检查 2

保存并重进世界后方向不改变。

检查 3

方块掉落后再次放置会重新读取玩家方向。

检查 4

四个变体只复用一個模型文件。

实机证据

截图来自本地 runClient 的复制测试世界；课程生成后临时截图代码已撤销，模板重新执行 Gradle build 成功。

07 故障地图与通关挑战

全部朝北

确认 `appendProperties` 添加 `FACING`，且 `getPlacementState` 被覆盖。

总是背对玩家

在玩家朝向后调用 `getOpposite`。

东西方向颠倒

检查 `blockstates` 中 `east/west` 的 `y` 旋转。

放置时崩溃

默认状态必须先包含合法的 `FACING` 值。

Boss 挑战

把第 7 课 `LEVEL` 与本课 `FACING` 放进同一个方块，列出 16 种组合，并用测试表逐项验证等级与方向都能保存。

下一课

第 9 课不再等待玩家右键，而是监听邻居变化，让红石信号自动控制方块。