

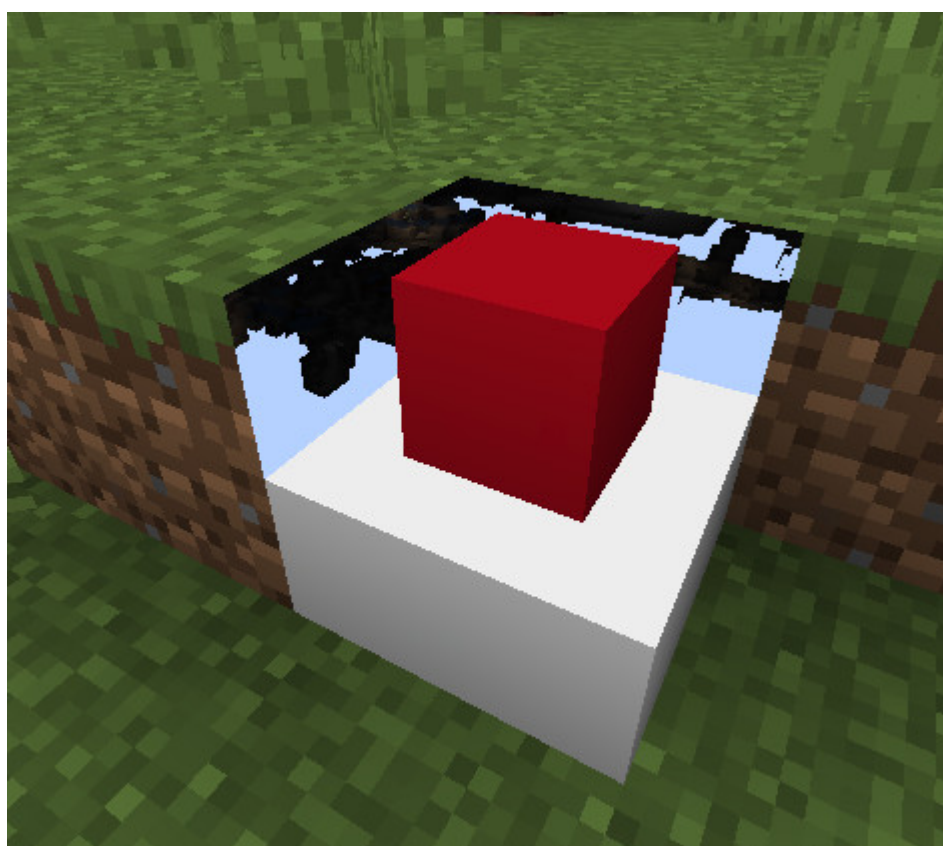
20 Přidání vlastního modelu bloku

V minulé lekci jsme si tvořili vlastní model bloku a nyní si ukážeme, jak jej přidat do hry a co je potřeba ještě nastavit.

Než si budeme moci přidat model do hry, tak si ještě musíme nastavit jednu věc, která souvisí s vykreslováním bloku.

Průhledné stěny bloku

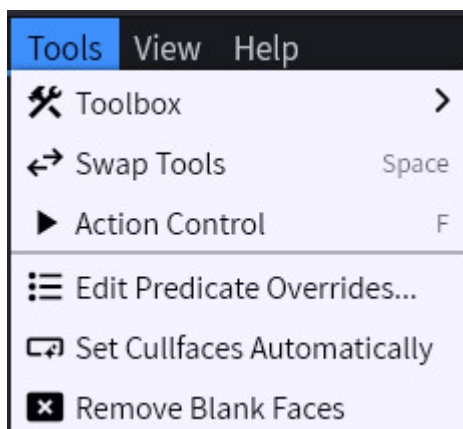
Když přidáváme vlastní model bloku, tak se nám může po přidání stát, že některé stěny okolních bloků budou průhledné.



Pokud se přepnete do spectate módu a dostanete se do nějakého bloku, tak nevidíte stěny všech bloků. Hra totiž vykresluje pouze stěny, které jsou vidět. Kdyby totiž hra měla vykreslovat stěny všech bloků, tak by moc dobře nefungovala a potřebovali bychom na to výkonný počítač. Tomuto se říká *culling* a můžeme si to nastavit i u našich bloků.

Toto chování ale musíme nastavit zvlášť pro každý blok v souboru s modelem `.json`. Pokud vytvoříme nějaký blok pomocí Blockbench, tak v základu toto nastavené není a vykreslují se všechny stěny. Blockbench má možnost toto nastavení vytvořit automaticky. To uděláme tak, že označíme všechny části bloku `Ctrl + A` a poté v menu zvolíme *Tools* → *Set Cullfaces Automatically*. Tím se nám ke každé stěně, které je až ke kraji bloku přidá

vlastnost `cullface` . Nemusíte řešit, jak přesně se tyto hodnoty nastavují - stačí jen vědět, že bychom to měli nastavit.



Přidání bloku do hry

Model a textury

Model (`json` soubor) budeme vkládat do složky `src/main/resources/assets/custom-block/models/block` .

Textury (`png` soubory) budeme vkládat do složky `src/main/resources/assets/custom-block/textures/block` .

Pro vložení bloku tedy stačí otevřít stažený `zip` soubor z Blockbench, kde máme uložený model a texturu bloku. A poté přesunout soubory do příslušných složek. Mělo by se nám zobrazit okno s informací, jaký soubor kam kopírujeme. Tam stačí zvolit *Refactor*. Stejně vložíme i druhý soubor.

Více textur

Pokud jste při tvorbě modelu použili více textur (třeba na různé části modelu), tak musíte zkopírovat všechny `png` soubory do složky s texturami.

Blockstate

Jaký model se má v základu použít pro zobrazení bloku.

Ve složce `src/main/resources/assets/custom-block/blockstates` vytvoříme soubor `custom_model.json` .

Obsah souboru bude vypadat následovně:

```
src/main/resources/assets/custom-block/blockstates/custom_model.json
```

```
1  {  
2    "variants": {
```

```

3      "": {
4          "model": "custom-block:block/custom_model"
5      }
6  }
7  }

```

Model itemu

Jak má vypadat item bloku (pokud item přidáváme).

Ve složce `src/main/resources/assets/custom-block/models/item` vytvoříme soubor `custom_model.json`.

Obsah souboru bude vypadat následovně:

src/main/resources/assets/custom-block/models/item/custom_model.json

```

1  {
2      "parent": "custom-block:block/custom_model"
3  }

```

Přidání bloku do hry

Jako poslední krok ještě musíme v kódu pro mód registrovat blok do hry. To uděláme v souboru `ModBlocks.java`. Toto už jsme dělali v lekcí [13 Přidání bloku - Vytvoření a nastavení bloku](#)

Musíme přidat klíč, pod kterým chceme blok (a item) registrovat. To je zároveň id bloku/itemu.

src/main/java/com/example/block/ModBlocks.java

```

1  public static final RegistryKey<Block> CUSTOM_MODEL_KEY = RegistryKey.of(
2      RegistryKeys.BLOCK,
3      Identifier.of(CustomBlock.MOD_ID, "custom_model")
4  );

```

Dále registrujeme samotný blok a případně změníme jeho nastavení.

src/main/java/com/example/block/ModBlocks.java

```

1  public static final Block CUSTOM_MODEL = register(
2      new Block(AbstractBlock.Settings.create()),
3      CUSTOM_MODEL_KEY,
4      true
5  );

```

Nyní bychom ve hře měli mít blok i item s texturou, kterou jsme si vytvořili.