

13 Přidání bloku

Už jsme si ukázali, jak přidat do hry item, dále si ukážeme, jak přidat blok. Později si pro blok vytvoříme i vlastní model a přidáme mu nějakou funkčnost.

Jako první si přidáme hlínu, která bude vyrobená z 9 hlíny. Takže to bude takový pevný blok hlíny.

Vytvoření projektu

Na stránkách fabricmc.net/develop/template si vytvoříme projekt s následujícím nastavením:

- Mod name
 - **Custom Block**
- Package Name
 - `com.example.block`
- Minecraft Version
 - `1.21`

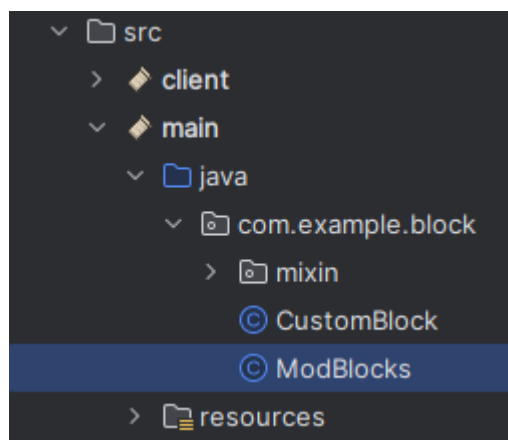
Zbytek necháme v základním nastavení a stáhneme si template. Ten po stažení extrahujeme a otevřeme soubor `build.gradle`. Počkáme, až se nám projekt nastaví a poté projekt zavřeme a znovu otevřeme, aby se dokončilo nastavení.

Přidání bloku

Stejně jako jsme měli u itemu třídu `ModItems`, kde jsme měly všechny itemy, tak i zde si vytvoříme třídu, kde budeme mít všechny bloky. Tuto třídu nazveme `ModBlocks`.

Vytvoření třídy

Třídu vytvoříme kliknutím pravým tlačítkem myši na složku a zvolíme *New > Java Class*



Stejně jako u itemu si i pro bloky vytvoříme metodu, která nám blok registruje a v tomto případě si přidáme i možnost, že nám k bloku přidá item. Ne všechny bloky totiž musí mít i item. Například bloky portálu nemají item a můžete je položit jen pomocí příkazu `setblock`.

Ve třídě `ModBlocks` si tedy vytvoříme metodu `register`, která bude vracet vytvořený blok (typ `Block`) a jako parametry budeme mít:

- blok, který chceme registrovat
 - `Block block`
- klíč (identifikátor), pod kterým chceme blok registrovat
 - `RegistryKey<Block> blockKey`
- zda chceme přidat i item nebo ne
 - `boolean registerItem`

ModBlocks.java

```
1 public static Block register(Block block, RegistryKey<Block> blockKey,
    boolean registerItem)
```

Registrace itemu k bloku

V metodě nejprve registrujeme item, pokud je hodnota `registerItem` rovna `true`.

ModBlocks.java

```
1 if (registerItem)
```

Dále si vytvoříme klíč, pod kterým chceme item registrovat a uložíme si jej do proměnné `itemKey`. Tento klíč vytvoříme pomocí metody `RegistryKey.of` a jako argumenty uvedeme typ klíče `RegistryKeys.ITEM` a dále identifikátor, pod kterým chceme item přidat. Tento identifikátor bude stejný jako u bloku a ten už máme v parametru `blockKey` - hodnotu získáme pomocí `blockKey.getValue()`.

ModBlocks.java

```
1 var itemKey = RegistryKey.of(RegistryKeys.ITEM, blockKey.getValue());
```

Vytvoříme si item k bloku (typ `BlockItem`), který bude vycházet z přidávaného bloku (parametr `block`) a bude mít základní nastavení `new Item.Settings()`.

ModBlocks.java

```
1 BlockItem blockItem = new BlockItem(block, new Item.Settings());
```

Na závěr item přidáme do registrů pomocí metody `Registry.register`. Jako argumenty uvedeme

- typ registru
 - `Registries.ITEM`
- klíč, pod kterým chceme item přidat
 - proměnná `itemKey`
- a samotný item, který chceme přidat
 - proměnná `blockItem`

ModBlocks.java

```
1 Registry.register(Registries.ITEM, itemKey, blockItem);
```

Celý kód s registrací itemu bude vypadat následovně

ModBlocks.java

```
1 if (registerItem) {
2     var itemKey = RegistryKey.of(RegistryKeys.ITEM,
    blockKey.getValue());
3     BlockItem blockItem = new BlockItem(block, new Item.Settings());
4     Registry.register(Registries.ITEM, itemKey, blockItem);
5 }
```

Registrace bloku

Blok registrujeme stejně jako item, jen zvolíme registr `Registries.BLOCK`. Tuto hodnotu vrátíme pomocí slova `return`.

ModBlocks.java

```
1 return Registry.register(Registries.BLOCK, blockKey, block);
```

Celá metoda pro registraci bloku (a itemu) bude vypadat následovně:

ModBlocks.java

```
1 public static Block register(Block block, RegistryKey<Block> blockKey,
    boolean registerItem) {
2     if (registerItem) {
3         var itemKey = RegistryKey.of(RegistryKeys.ITEM,
    blockKey.getValue());
4         BlockItem blockItem = new BlockItem(block, new
    Item.Settings());
5         Registry.register(Registries.ITEM, itemKey, blockItem);
6     }
7
8     return Registry.register(Registries.BLOCK, blockKey, block);
9 }
```

Metoda `initialize`

Dále potřebujeme přidat metodu, která zajistí, že se všechny bloky v této třídě vytvoří. V této metodě zatím nebude žádný kód a ani nebude vracet žádnou hodnotu.

ModBlocks.java

```
1 public static void initialize() { }
```

Vytvoření a nastavení bloku

Pro přidání bloku si nejprve musíme vytvořit klíč, pod kterým chceme blok registrovat.

Bude to proměnná, která nebude měnit svoji hodnotu `public static final` typu `RegistryKey<Block>` (klíč pro registraci bloku) s názvem `CONDENSED_DIRT_KEY`.

Klíč vytvoříme pomocí metody `RegistryKey.of`. Jako argumenty uvádíme

- typ klíče
 - `RegistryKeys.BLOCK`
- identifikátor, pod kterým chceme blok přidat
 - identifikátor je hodnota `id-módu:název` (např. `minecraft:dirt`)
 - metoda `Identifier.of` s argumenty `id módu` (to máme uložené v `CustomBlock.MOD_ID`) a samotný název bloku `condensed_dirt`

ModBlocks.java

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

Nyní už máme vše potřebné pro registraci bloku, takže si jej můžeme vytvořit pomocí naší metody `register` a uložit si jej do proměnné `public static final Block CONDENSED_DIRT`. Jako argumenty metody uvedeme

- blok, který chceme registrovat
 - vytvoříme pomocí `new Block`
 - jako argument uvádíme nastavení, které získáme ze třídy `AbstractBlock`
 - zatím nám stačí základní nastavení `AbstractBlock.Settings.create()`
- klíč, pod kterým chceme blok registrovat
 - proměnná `CONDENSED_DIRT_KEY`
- informaci, zda chceme vytvořit i item nebo ne
 - hodnota `true` (ano) nebo `false` (ne)

ModBlocks.java

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

Tím máme veškeré věci na přidání bloku hotové, ale když hru spustíme, tak se blok ještě nepřidá. To si vyřešíme v následující lekci.

Celý kód třídy `ModBlocks`

ModBlocks.java

```
1     public class ModBlocks {  
2         public static final RegistryKey<Block> CONDENSED_DIRT_KEY =  
3         RegistryKey.of(  
4             RegistryKeys.BLOCK,  
5             Identifier.of(CustomBlock.MOD_ID, "condensed_dirt")  
6         );  
7         public static final Block CONDENSED_DIRT = register(  
8             new Block(AbstractBlock.Settings.create()),  
9             CONDENSED_DIRT_KEY,  
10            true  
11        );  
12  
13        public static void initialize() { }  
14  
15        public static Block register(Block block, RegistryKey<Block> blockKey,  
16        boolean registerItem) {  
17            if (registerItem) {  
18                var itemKey = RegistryKey.of(RegistryKeys.ITEM,  
19                blockKey.getValue());  
20                BlockItem blockItem = new BlockItem(block, new  
21                Item.Settings());  
22                Registry.register(Registries.ITEM, itemKey, blockItem);  
23            }  
24            return Registry.register(Registries.BLOCK, blockKey, block);  
25        }  
26    }
```