

22 Získání jména hráče, vytvoření exploze

V této lekci si ukážeme, jak do chatu vypsát jméno hráče, jak změnit místo, kde se zpráva zobrazí. Dále si ukážeme, jak na místě bloku vytvořit explozi. Blok si poté přidáme do kreativního inventáře a přidáme si k bloku název.

Zobrazení jména ve zprávě

V předchozí lekci ([21 Nastavení hitboxu a kliknutí na blok](#)) jsme si ukazovali, jak hráči zobrazit zprávu v chatu. Nyní si ukážeme, jak do zprávy přidat i jméno hráče.

Jméno získáme pomocí metody `getName()` ze třídy `PlayerEntity`. Abychom ale jméno mohli zobrazit, tak jej potřebujeme převést z typu `Text` na typ `string` (textový řetězec). Dále už jej můžeme používat jako jakýkoliv jiný text, který chceme zobrazit ve zprávě.

```
src/main/java/com/example/block/ExplodingBlock.java
```

```
1 player.sendMessage(Text.empty()  
2     .append(Text.literal("Ahoj, ").formatted(Formatting.RED,  
    Formatting.UNDERLINE))  
3     .append(Text.literal(player.getName().getString()).formatted(Formatting.AQ  
4     .append(Text.literal(", vítěj!").formatted(Formatting.GOLD)));
```

Zobrazení zprávy nad hotbarem

Pokud bychom chtěli zprávu zobrazit nad hotbarem a ne v chatu, tak to můžeme udělat jednoduše přidáním parametru `true` za zprávu v metodě `sendMessage`

```
src/main/java/com/example/block/ExplodingBlock.java
```

```
1 player.sendMessage(Text.empty()  
2     .append(Text.literal("Ahoj, ").formatted(Formatting.RED,  
    Formatting.UNDERLINE))  
3  
    .append(Text.literal(player.getName().getString()).formatted(Formatting.AQ  
4     .append(Text.literal(", vítěj!").formatted(Formatting.GOLD)), true
```



Přidání exploze

Vytvoření exploze už jsme si ukazovali v kapitole [09 Úprava chování hůlky - Vytvoření exploze](#).

Oproti použití s hůlkou tady změníme pár věcí.

Jako první parametr se uvádí entita, která výbuch vyvolala. To v tomhle případě není žádná entita, protože zdroj je blok, takže můžeme použít hodnotu `null` (reprezentuje žádnou hodnotu).

Dále uvádíme pozice, kde se má exploze vytvořit. V tomto případě máme pozice bloku dostupné v parametru `pos`.

Další parametr uvádíme sílu exploze. TNT má sílu 4, charged creeper 6.

Dále uvádíme, zda se má po výbuchu vytvořit oheň - `true` (ano), `false` - ne.

Jako poslední uvádíme zdroj exploze. Což je v tomto případě blok (`World.ExplosionSourceType.BLOCK`).

Celý kód s explozí

```
src/main/java/com/example/block/ExplodingBlock.java
```

```
1  @Override
2  protected ActionResult onUse(BlockState state, World world, BlockPos pos,
   PlayerEntity player, BlockHitResult hit) {
3      if (world.isClient) {
4          return ActionResult.PASS;
5      }
6
7      player.sendMessage(Text.empty()
8          .append(Text.literal("Ahoj, ").formatted(Formatting.RED,
   Formatting.UNDERLINE)))
```

```

9
    .append(Text.literal(player.getName().getString()).formatted(Formatting.AQUA))
10    .append(Text.literal(", vítěj!").formatted(Formatting.GOLD)),
    true);
11
12    world.createExplosion(null, pos.getX(), pos.getY(), pos.getZ(), 10f, true,
    World.ExplosionSourceType.BLOCK);
13
14    return super.onUse(state, world, pos, player, hit);
15 }

```

Dokončení akce

Jako poslední musíme z metody použití bloku vrátit informaci, zda akce proběhla úspěšně. V tomto případě na konci vrátíme, že akce proběhla úspěšně `ActionResult.SUCCESS`.

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

```

1  @Override
2  protected ActionResult onUse(BlockState state, World world, BlockPos pos,
    PlayerEntity player, BlockHitResult hit) {
3      if (world.isClient) {
4          return ActionResult.PASS;
5      }
6
7      player.sendMessage(Text.empty()
8          .append(Text.literal("Ahoj, ").formatted(Formatting.RED,
    Formatting.UNDERLINE))
9
10     .append(Text.literal(player.getName().getString()).formatted(Formatting.AQUA))
11     .append(Text.literal(", vítěj!").formatted(Formatting.GOLD)),
    true);
12
13     world.createExplosion(null, pos.getX(), pos.getY(), pos.getZ(), 10f, true,
    World.ExplosionSourceType.BLOCK);
14
15     return ActionResult.SUCCESS;
16 }

```

Přidání do kreativního inventáře

Jak blok přidat do kreativního inventáře už jsme si ukazovali v kapitole [14 Přidání bloku do inventáře, textura bloku - Přidání do kreativního inventáře](#).

Blok si přidáme do skupiny funkčních bloků `ItemGroups.FUNCTIONAL` a přidáme si jej na konec `group.add()`.

V metodě `initialize` ve třídě `ModBlocks` přidáme následující kód. Můžete klidně zkopírovat kód na přidání předchozího bloku a jen upravit.

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

```
1  public static void initialize() {
2      ItemGroupEvents.modifyEntriesEvent(ItemGroups.NATURAL)
3          .register(group -> {
4              group.addAfter(Items.DIRT, CONDENSED_DIRT.asItem());
5          });
6
7      ItemGroupEvents.modifyEntriesEvent(ItemGroups.FUNCTIONAL)
8          .register(group -> {
9              group.add(CUSTOM_MODEL.asItem());
10         });
11 }
```

Překlad bloku

Jak přidat překlad itemu/bloku jsme si ukazovali v kapitole [04 Přidání itemu část 2 - Název itemu](#).

Do souboru `src/main/resources/assets/custom-block/lang/en_us.json` si přidáme následující hodnotu:

```
1  {
2      "block.custom-block.condensed_dirt": "Condensed Dirt",
3      "block.custom-block.custom_model": "Exploding Block"
4  }
```