Gymnázium, Praha 6, Arabská 14

Programování

ROČNÍKOVÁ PRÁCE

2024

Jana Šrámková

Arabská 14, Praha 6, 160 00

ROČNÍKOVÁ PRÁCE

Předmět: Programování

Téma: Minecraft Mod

Autorka: Jana Šrámková

Třída: 4. E

Školní rok: 2023/2024

Vedoucí práce: Mgr. Jan Lána

Třídní učitel: PhDr. Markéta Šlegerová

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V dne

Anotace

Jedná se o mód pro hru Minecraft, který bude obsahovat zcela nového moba, který bude moct být ochočen a i osedlán, a krom toho bude mít všechny funkce obyčejných Minecraft mobů. Krom přidám itemy, které bude možné ze zabití moba získat, a brnění a zbraně, které se z nich dají vyrobit. K ochočení moba bude také potřebován specifický item, který rovněž přidám jakožto i způsob, jak ho vytvořit.

Zadání

Vytvořím mód, který bude obsahovat nového moba s vlastním modelem, chováním a animacemi vytvořenými v programu Blockbench.

Krom toho přidám i několik dalších věcí:

Itemy, které je možné získat ze zabití zmíněného moba, a věci z nich vyrobené. Tyto itemy budou mít pixelovou grafiku převedenou do 3D a speciální funkce.

Dvě nové tvořící stanice - v podstatě továrny. První stanice bude mít vlastní uživatelské rozhraní, a bude v určitých intervalech kombinovat některé předměty a vytvářet z nich nové. Druhá tvořící stanice/entita, bude postrádat uživatelské rozhraní a bude se jednat o jakousi továrnu s vlastním modelem a animacemi opět z programu Blockbench.

Zbytek obsahu módu bude tvořit několik dalších předmětů soustředěných na přidanou bytost. Ty budou mít pixelové, a výjimečně 3D modely a zvláštní funkce.

Podrobnosti o jednotlivých předmětech a jejich funkcích budou rozepsány v dokumentaci.

Obsah

Úvod

Tato práce je dokumentací ke tvorbě mé ročníkové práce. Jedná se o mód pro hru Minecraft verze 1.20.X s použitím Forge. Minecraft módy jsou projekty postavené na základním programu samotného Minecraftu. Některé do hry přidávají zcela nové věci, funkce, či bytosti, zatímco jiné mění nebo rozšiřují již existující části hry.

V případě mé práce se jedná o mód soustředěný kolem přidání zcela nové bytosti, jejích interakcí s hráčem, a vytvoření a implementace předmětů, které se díky této bytosti dají zužitkovat.

Kvůli nedostatku českých výrazů budou v této dokumentaci občas použit anglické výrazy nebo jejich počeštěné verze. K takovýmto výrazům bude vždy poskytnuto vysvětlení

Minecraft

Minecraft je velmi populární počítačová hra vyvinutá v roce 2009 Markusem Perssonem. Zároveň s Minecraftem byla založena i společnost Mojang Studios, která hru dále rozvíjí i po té co byla spolu se samotným Minecraftem odkoupena společností Microsoft.

Hra se setkala ve světě s obrovským úspěchem. Jedná se o nejprodávanější počítačovou hru v historii, s více než 250 miliony prodaných kopií.

Kvůli relativní jednoduchosti byly na Minecraft za dobu jeho existence vytvořeny celé stovky módů, mnohé z nichž složitostí a obsahem daleko přesahují hru samotnou.

Použité nástroje

Forge

Spolu se svým dvojčetem Fabric, Forge je jedním s nejpoužívanějších nástrojů pro implementování módů do Minecraftu. Jedná se o bezplatný program, který sám o sobě s módy nic nedělá, pouze spouští jejich kód spolu se základní hrou.

JDK – Java Development Kit

JDK je soubor základních nástroj pro vývoj aplikací v jazyce Java vytvořený společností Oracle Corporation. Jelikož samotný Minecraft tento nástroj používá, většina módů – včetně mého – s ním taktéž pracuje.

Gimp

Každý předmět ve hře má v inventáři hráče svou vlastní ikonu, tradičně v rozměrech 16x16 pixelů. Pokud není v kódu řečeno jinak, tato ikona je pak převedena do 3D podoby a použita i jako podoba daného předmětu ve hře. Pro vytváření těchto ikon do svého módu používám právě nástroj Gimp, což je bezplatný grafický editor.

Blockbench

Blockbench je nástroj pro vytváření 3D modelů, vytvořený speciálně pro Minecraft, i když se v něm dají tvořit i jiné modely. Právě v něm jsem vytvořila veškeré 3D modely pro svůj mód.

Příprava prostředí

Celý program je napsaný v jazyce Java v prostředí IntelliJ. Krom toho je potřeba balíček JDK neboli Java Development Kit, a také základní kód z oficiální stránky Forge ve správné verzi, což je v mém případě 1.20.1. Tento kód přidá do vybavení IntelliJ základní knihovny Minecraftu a poskytne základ který dovolí módu pracovat se hrou a poskytne možnost spouštět mód přímo skrz IntelliJ.

Základní kód

EventBus

EventBus je takzvaná sběrnice událostí, která umožňuje zachytit reagovat na události jak ze základní hry, tak z módu.

Hlavní taková sběrnice používaná pro většinu událostí se nachází na MinecraftForge#EVENT_BUS. Existuje další sběrnice událostí pro události specifické pro mody umístěné na FMLJavaModLoadingContext#getModEventBus, která se používá pouze ve specifických případech.

Každá událost je spuštěna na jedné z těchto sběrnic, většina událostí je spouštěna na hlavní sběrnici událostí kódu Forge, ale některé jsou spouštěny na sběrnicích událostí specifických pro daný mod. Takové musí být označené pomocí Mod id. Obsluha události je pak metoda, která byla zaregistrována do sběrnice událostí.

Mod id

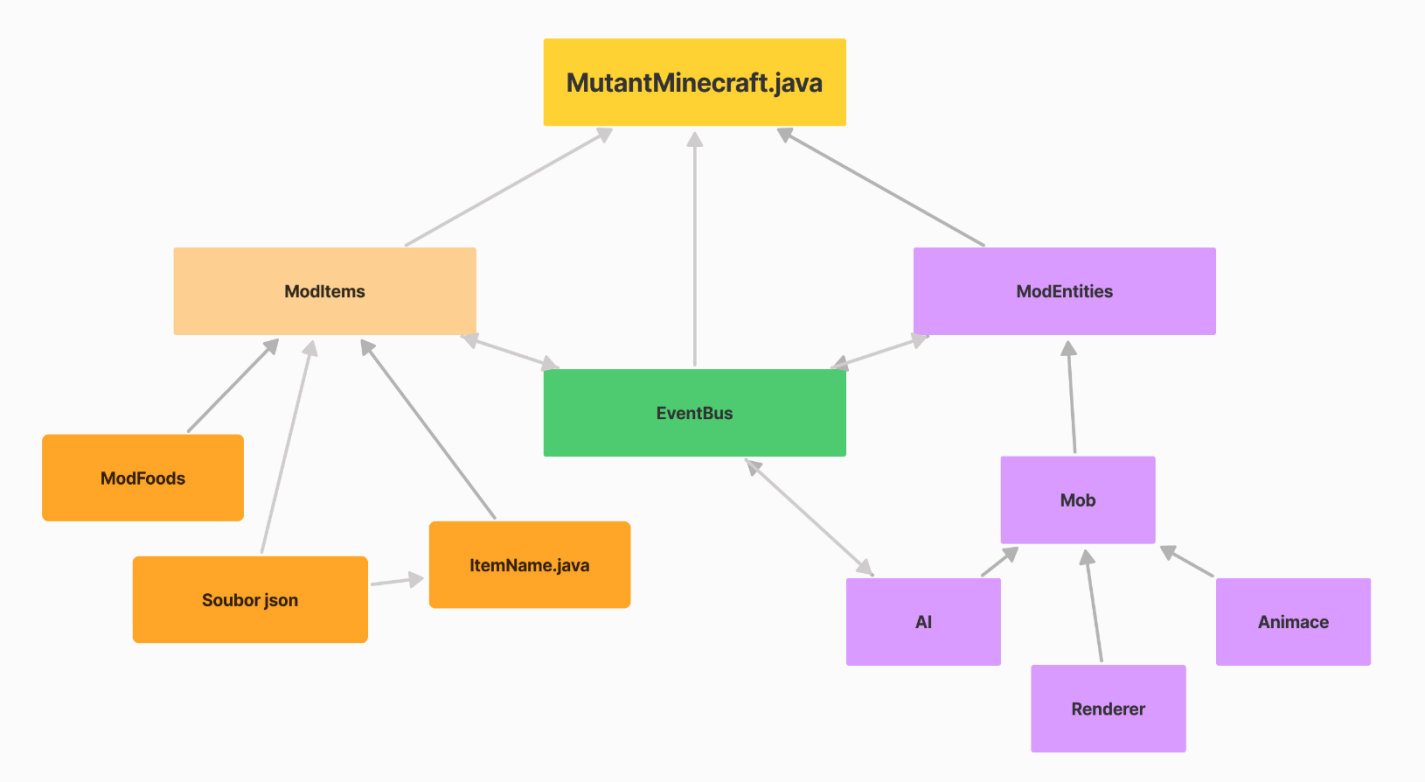
Jedná se o jakousi stopu, která je specifická pro každý mód. S pomocí Mod id se například příkazy přesměrovat ze souborů základní hry do identicky pojmenovaných souborů módu.

En_lang

Jedná se o jednoduchý textový soubor, který zajišťuje překlad jmen jednotlivých aspektů módu do podoby čitelné pro lidi. K obecnému jménu každé věci je zde přiřazen název, pod kterým se objeví ve hře.

MutantMinecraft.java

Jedná se o hlavní třídu módu, ve které jsou registrovány všechny ostatní třídy. Pokud by zde nějaká třída zaregistrovaná nebyla, nezáleželo by na tom, že je zcela správně naprogramovaná, mód by ji nepoužíval.



Obrázek 1: Poněkud zjednodušený graf představující základní propojení módu

Přidané věci

Hlavním cílem mého módu bylo přidat do hry několik věcí, které se dají rozdělit do následujících kategorií.

Mod Creative tab

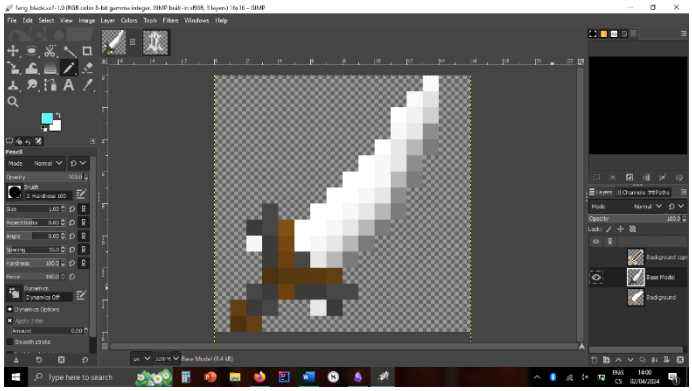
Jedná se o záložku v herním inventáři, odkud je možné všechny itemy dostat bez potřeby použití příkazů. Do této záložky jsou itemy iniciovány ve třídě *ModCreativeModTabs*. Samotné itemy jsou naprogramovány v jiných třídách, v této se pouze pomocí jména přidávají do daných záložek. Záložek je možno vytvořit identickými metodami s jiným jménem i více, ale v mém módu je potřeba jen jedna.

Itemy

Jedná se o základní komponent módu. Itemy jsou věci, které může hráč přidat do svého inventáře, a nějak s nimi interagovat. Většina věcí, které přidávám, má nějaký item, který se k nim váže, tudíž dává hráči možnost ji přenášet v inventáři.

Ikony

Každý item potřebuje ikonu, kterou bude ve hře reprezentován. S touto ikonou se objeví například v inventáři hráče, a u mnoha předmětů jsou také použity jako základ pro jejich 3D reprezentaci. Tyto ikony jsem vytvořila v programu Gimp. Jedná se o jednoduché obrázky rozměru 16x16 pixelů. Teoreticky by mohli být i větší, ale jelikož 16x16 je velikost ikon základní hry, ikony s větším rozlišením vypadají nepatřičně a hrozí, že budou přesahovat jednotlivé buňky inventáře hráče. Zobrazit některé předměty s pouze 16ti pixely je poněkud těžký úkol, ale po několika pokusech se většina předmětů celkem vydařila.



Obrázek 2: Ikona jednoho z itemů v programu Gimp

Json

Každý předmět musí být opatřen svým vlastním souborem json. Tento soubor obsahuje informace o tom, o jaký předmět se jedná, jeho jméno, a cesta k místu v kódu, kde je uložená jeho ikona. Jméno itemu v tomto souboru je potom použito pro propojení různých částí kódu daného itemu.

Code | Blame | 6 lines (6 loc) · 118 Bytes | Code 55% faster with GitHub Copilot

```json
{
  "parent": "item/template_spawn_egg",
  "textures": {
    "layer1": "mutminecraft:item/direwolf/direwolf_egg"
  }
}
```

Obrázek 3: Příklad json souboru itemu Direwolf Spawn Egg

Itemy, které mají speciální vlastnosti, mají pak také každý svou vlastní třídu, ve kterých jsou tyto vlastnosti naprogramované.

Jídlo

Kód itemů, které jsou ve hře použité jako jídlo, je rozdělený do dvou tříd, *ModFoods*, kde jsou definované jednotlivé vlastnosti daného jídla pomocí knihovní metody, a *ModItems*, kde je předmět jako takový registrován do módu pomocí jména, které je k danému jídlu přiřazené v předchozí třídě. Zbytek kódu je pak identický s ostatními itemy

```java
public class ModFoods {
    public static final FoodProperties RAW_MEAT = new FoodProperties.Builder().meat().nutrition(3).saturationMod(0.2F)
            .effect(() -> new MobEffectInstance(MobEffects.CONFUSION, 150), 0.3F).build();
    public static final FoodProperties COOKED_MEAT = new FoodProperties.Builder().meat().nutrition(9).saturationMod(0.9F)
            .effect(() -> new MobEffectInstance(MobEffects.DAMAGE_BOOST, 200), 0.6F).build();
```

Obrázek 4: Příklady metod definujících jídlo

Brnění

Tato část módu bohužel kvůli času nebyla dokončena. Ve třídě *ModArmorMaterials* je knihovními metodami definováno, jaké zvuky, odolnost a vlastnosti má kus brněný přidaného mým módem mít, a item brnění má i ikonu a místo v záložce módu, technicky tudíž naprosto funguje. Brnění ale nemá 3D model, takže když si ho hráč oblékne, vypadá jen jako základní model s černo fialovou čtvercovou texturou.

Obrázek 5: Brnění bez textury

Recepty

Recepty definují způsob, jakým se z určité kombinace itemů módu a základní hry dají vytvořit jiné. Nachází se ve své vlastní složce *recipes*, a každý recept musí mít svůj vlastní json soubor. V každém soubor jsou definovány ingredience pro daný recept a druh receptu.

Druhů receptů je mnoho a záleží v nich na tom, s jakou tvořící stanicí jsou použity. V mém módu se vyskytují například recepty druhu crafting_shaped a crafting_shapeless, které se od sebe liší tím, že prvně zmíněný má přesně danou šablonu, do které musí být ingredience vloženy, zatímco u druhého na tom nezáleží.

Podle druhu receptu jsou v každém souboru json definovány další potřebné hodnoty, jako je čas pro tvorbu, počet vytvořených předmětů, a v případě crafting_shaped také jeho šablona. V takovém případě je zároveň ke každé již definované ingredienci přiřazen jednoznakový klíč, se kterým je pak šablona vytvořena.

Tagy

Pro některé účely, jako je třeba použití v receptech, nebo aby se nemuselo velké množství itemů opakovaně vypisovat, je možné libovolný počet itemů seskupit pod takzvaným tagem. Ve třídě *ModTags* se zadá název tagu, specifikuje se, o jaký druh itemů se jedná, a samozřejmě se vloží itemy jako takové a tag je pak možno používat jako plošný název pro všechny itemy v něm

```
{
  "type": "minecraft:crafting_shaped",
  "category": "misc",
  "key": {
    "M": {
      "tag": "mutminecraft:treat_usable_meats"
    },
    "B": {
      "item": "minecraft:bone"
    }
  },
  "pattern": [
    "MMM",
    "MBM",
    "BMM"
  ],
  "result": {
    "item": "mutminecraft:treat",
    "count": 1
  }
}
```

Obrázek 6: Příklad použití tagu v receptu

Mob

Mob je hlavní součástí mého módu. Jedná se o žijící entitu, která se pohybuje herním světem nezávisle na hráči díky svému vlastnímu AI. Každý mob musí být takzvaně zaregistrován ve třídě *ModEntities*, a její kód je pak následně rozdělen do několika souborů

AI

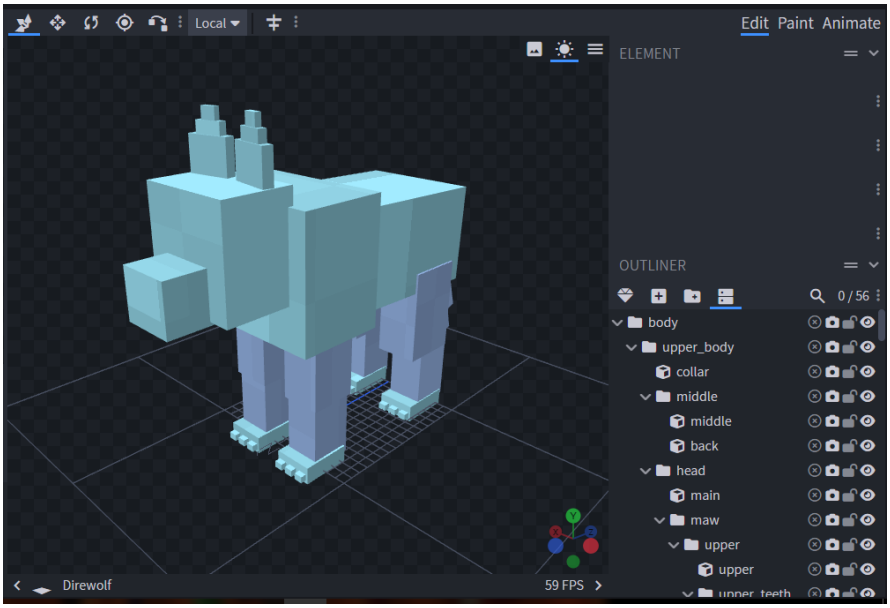
V souborech AI se předvídatelně nachází program, který definuje chování moba. Může se zde ukládat jak kód pro jednotlivé moby, tak plošné metody, které jde později implementovat pro každého moba bez potřeby vytvářet stále nové třídy.

Třída *AttackGoal*, která se v této složce nachází, definuje jak a za jakých podmínek má mob zaútočit. Objevuje se zde několik hodnot, jako například hodnoty private int attackDelay a *attackCooldown*, nebo *private boolean countTillNextAttack.* V metodě start se k těmto hodnotám přiřadí čísla a metoda a pokud je vyhověno všem podmínkám, *countTillNextAttack* se přepne na true. Jakmile pak odpočet dosáhne hodnoty rovné *attackCooldown*, cíl je poškozen, a odpočet začne znovu. Celý proces může být kdykoliv ukončen metodou stop.

Kdy se má metoda start nebo stop spustit je definováno ve třídě náležící každému mobovi zvlášť, v tomto případě tedy ve třídě DirewolfEntity.java. Zde je taky určené intervaly mezi jednotlivými animacemi útoku. Ty je podle toho, jak dlouhá je animace udělaná, vypočítat tak, aby se viditelný úder shodoval s časem, kdy je cíl poškozen. Tyto odmlky jsou počítané v ticích, což je jedna dvacetina vteřiny.

Client

Ve složce klient se řeší model moba a to, jak bude vypadat ve hře, čili jeho velikost, velikost stínu, nebo jaká jeho část se bude pohybovat při sledování hráče. Většina práce je zde vykonaná mimo IntelliJ v programu Blockbench, kde byl model vytvořen. Části moba se zde musí seskupit pod takzvané kosti, se kterými se pak hýbe při tvorbě animace. Jednu tuto kost je ve třídě *ModAnimationDefnitions* potřeba vybrat jako hlavu, která sleduje okolí, a zbytek program udělá sám při exportu modelu.



Obrázek 7: Model během modelování

Ve třídě *DirewolfRenderer* je pak k modelu připojena cesta k místu, kde je v programu uložena jeho textura, a také jak velký má mob být v normální dětské verzi. Touto třídou se zajistí, že se mob objeví ve hře.

Obrázek 8: Mob ve hře, dospělá i dětská varianta v porovnání s obyčejným vlkem

Animace

Animace jsou samy o sobě pouze matematické funkce, ve kterých se podle souřadnic určuje, jak daleko se má která část modelu přesunout v jaký čas. I když je lze programovat ručně, je mnohem jednodušší použít program, ve kterém je vytvořený model moba i pro vytvoření animací a ty pak exportovat. Program tak animace sám převede do potřebné podoby a výsledný kód je pak už jen potřeba zkopírovat do třídy *ModAnimationDefinitions*. Tvorba dobrých animací i tak trvá dlouho, a při tvorbě mého modelu jsem první verzi omylem vytvořila ve špatném formátu. Když jsem pak model přetvořil do správné podoby, animace některých končetin zřejmě bezdůvodně nefungovaly tak, jak by měly. Oprava tohoto problému zabrala několik hodin, a i přes všechnu mou snahu se animace pohybu moba stále v některých úsecích chová nepatřičně.

Custom

Ve složce klient se nachází třídy obsahují kód týkající se pouze jednotlivých mobů. Jsou zde základní statistické údaje moba, jako počet životů, síla útoku, a jakou rychlostí se pohybuje.

Jelikož je můj mob ochočitelný, je zde definován i způsob, jak ho ochočit a jak se touto akcí změní jeho statistiky.

Pomocí metody setup *AnimationStates* se zde iniciují jednotlivé animace a k těm potřebným, jako je v mém případě animace útoku, se přidává délka prodlevy, opět definovaná v ticích.

Velkou částí třídy jsou pak *registerGoals*, kde jsou knihovní metody, které dávají mobovi některé základní cíle, jako náhodné procházení po světě, čím může být mob nalákán, aby sledoval hráče, a v případě ochočitelných mobů jako je ten můj, jestli má hráč schopnost přikázat mobovi, aby někde čekal, nebo jestli má za svého majitele mob bojovat, když je na něj útočeno. Každý cíl má pak danou důležitost v podobě celého kladného čísla. Čím nižší číslo je, tím větší prioritu daný cíl má.

V metodách obsahujících slovo sound je definováno, jaký zvuk má hrát při nejrůznějších akcích moba. Přidávat vlastní zvuky je celkem jednoduché, pokud je člověk umí vytvořit, ale kvůli nedostatku času jsem použila zvuky jiného moba ze hry, jelikož můj mob má být jeho zmutovanou variantou.

Metoda *wantsToAttack* dává další podmínky pro to, na co je mob ochotný zaútočit, a k jakým mobům je agresivní i bezdůvodně. Jedná se o jednoduchou sérii metod *if*.

Metoda *mobInteract* definuje, jak mob reaguje na určité itemy, se kterými s ním hráč interaguje. Pomocí metod *if* se jednoduše přidá název daného itemu a jak na něj má mob reagovat. V této metodě by také mělo být definováno, jak na moba nasednout, jelikož to byla jedna z jeho plánovaných schopností, ale jelikož se tato jedna podmínka pere s podmínkou základní hry pro to, jak říct mobovi aby někde čekal, ani jeden tento příkaz ve výsledku nefunguje.

Závěr

Sice jsem ani zdaleka nestihla vše, co jsem měla v plánu – například na tvořící stanice vůbec nedošlo a mnoho z přidaných itemů není zcela dokončených – to, co jsem stihla, se ale velmi vyvedlo, a s výsledkem jsem spokojená. Vyskytly se sice komplikace, které proces výrazně zpomalily, ale díky nim teď programování módů rozumím mnohem lépe než dřív, a další práce na mém módu bude o to jednodušší. Pokud budu mít v budoucnu čas, hodlám se k tomuto projektu vrátit a řádně ho dokončit.

Zdroje

https://codakid.com/guide-to-minecraft-modding-with-java/

https://codakid.com/guide-to-minecraft-modding-with-java/

https://www.youtube.com/watch?v=sfl0GUL0vtk

https://parchmentmc.org

https://www.blockbench.net/

https://courses.kaupenjoe.net/p/modding-by-kaupenjoe-forge-modding-for-minecraft-1-20-x?coupon_code=CARNIVAL24

https://www.infoworld.com/article/3296360/what-is-the-jdk-introduction-to-the-java-development-kit.html

Seznam obrázků