STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

Christian Abraham

Mladá Boleslav 2022

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

**Autor: Christian Abraham  
Studijní obor: 18-20-M/01 Informační technologie  
Vedoucí práce: Lukáš Mázl**

Mladá Boleslav 2022

# Obsah

[Obsah 3](#_Toc101801498)

[1 Úvod 6](#_Toc101801499)

[2 Technický úvod 7](#_Toc101801500)

[2.1 Potřebné programy pro koncového uživatele 7](#_Toc101801501)

[3 Program z pohledu uživatele 8](#_Toc101801502)

[3.1 Základní příkazy pro spuštění 8](#_Toc101801503)

[3.1.1 Po připojení na server 8](#_Toc101801504)

[3.1.2 Příkaz /pluginenable 8](#_Toc101801505)

[3.1.3 Příkaz /storyenable 9](#_Toc101801506)

[3.2 Příběh pluginu 9](#_Toc101801507)

[4 Plugin z pohledu programátora 10](#_Toc101801508)

[4.1 Event a příkaz 10](#_Toc101801509)

[4.1.1 Event 10](#_Toc101801510)

[4.1.2 Příkaz 11](#_Toc101801511)

[4.2 Logika programu 11](#_Toc101801512)

[4.2.1 Určení vzálednosti 12](#_Toc101801513)

[4.2.2 Spawnování zombies 12](#_Toc101801514)

[4.2.3 Inventář a nastavení hráče 13](#_Toc101801515)

[4.2.4 Reset pluginu 13](#_Toc101801516)

[4.2.5 Vypisování příběhu a časový limit 13](#_Toc101801517)

[4.2.6 Teleportace po hitu entity 13](#_Toc101801518)

[4.2.7 Konec hry 14](#_Toc101801519)

[5 Závěr 15](#_Toc101801520)

[6 Zdroje 16](#_Toc101801521)

[7 Přílohy 17](#_Toc101801522)

[7.1 Seznam obrázků 17](#_Toc101801523)

**Prohlášení**

Prohlašuji, že jsem svou ročníkovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti zpřístupňování této ročníkové práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Mladé Boleslavi dne podpis:

# Úvod

Cílem této ročníkové práce bylo vytvořit příběhové rozšíření do hry minecraft, dále jen „plugin“, v programovacím jazyce java, který si mohou zahrát hráči hry minecraft. Hlavní částí tohoto pluginu je framework[[1]](#footnote-1) „minecraft developement“, který je určený přímo na vývoj rozšíření do této hry.

Přínos této práce bylo vytvořit rozšíření, které zaujme komunitu hráčů hry Minecraft[[2]](#footnote-2).

Tato práce se zaměřuje na vývoj pluginů ve hře minecraft a řešení problémů se syntaxí daného frameworku, práci s eventy, příkazy a programy potřebné pro vývoj.

# Technický úvod

Programů potřebných pro vývoj a následující používání je celá řada. Tato práce byla dělána v IDE IntelliJ IDEA[[3]](#footnote-3) s frameworkem „minecraft development“. Dále je potřeba minecraftový server, na kterém se bude plugin spouštět. Zde byl použit server od spigotu[[4]](#footnote-4), jelikož je to jeden z nejpoužívanějších serverů pro minecraft a jeho instalace není složitá. V neposlední řadě je potřeba i samotná hra, minecraft. Ta je placená a stojí 19,99 EUR.

## Potřebné programy pro koncového uživatele

Uživatel bude pro použití pluginu potřebovat hru minecraft, minecraftový server a samotný plugin, který je téma ročníkové práce. Server se dá bezplatně stáhnout z internetových stránek spigot.

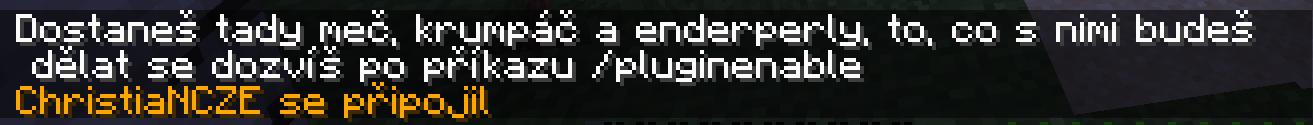
# Program z pohledu uživatele

## Základní příkazy pro spuštění

Po spuštění hry minecraft a připojení se na server bude uživatel nucen napsat následující příkazy pro spuštění samotného pluginu.

### Po připojení na server

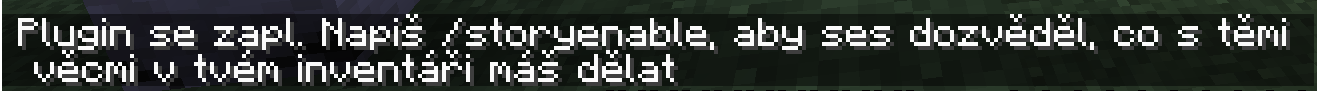
Jakmile se hráč připojí na server, ihned se mu vypíše zpráva, která ho informuje o tom, že dostal meč, krumpáč a enderperly. Věci, které jsou potřebné k dohrání pluginu a všem se zobrazí zpráva, ve které je jméno hráče a oznámení o tom, že se připojil na server.



Obrázek 1: pripojeni\_na\_server

### Příkaz /pluginenable

Po spuštění příkazu /pluginenable se zapne celá logika pluginu (kromě příběhu) a časového limitu. Následně se vypíše zpráva hráči o dalších informacích, viz. obrázek2.



Obrázek 2: prikaz\_pluginenable

### Příkaz /storyenable

Po spouštění příkazu /storyenable se zapne příběh a časový limit. Od této chvíle již hráč může začít hrát.



Obrázek 3: prikaz\_storyenable

## Příběh pluginu

Jelikož tato práce má být příběhový plugin do hry minecraft, byl potřeba vytvořit příběh, který bude hráče provázet do samotného konce. Po spuštění příkazu /storyenable se spustí hlavní příběh. Hlavní příběh začíná tím, že na Evropu byly shozeny atomové bomby a naším úkolem je se dostat v omezeném čase na určitou lokaci ve světě, dále jen „stronghold“, a následně skočit do portálu, abychom se dostali do protiatomového bunkru. Cestou hráče překvapí spoustu ruzných překážek a nebezpečí, které bude muset omezeném čase překonat.

# Plugin z pohledu programátora

## Event a příkaz

K pochopení na jaké bázi plugin funguje, je potřeba pochopit, co je to event a příkaz.

### Event

Event je jako funkce, která spustí metodu, když se něco stane. Ve frameworku minecraft developement jsou jich stovky. Je zde například event „PlayerMoveEvent“, který spustí danou metodu a v ní napsaný kód vždy, když se hráč pohne. Pak už jen stačí na danou třídu přidat listener, který „poslouchá“, co se v té hře děje.

V hlavní třídě se pak musí vždy daný event registrovat, aby správně fungoval.

getServer().getPluginManager().registerEvents(new FoundStronghold(),**this**);

getServer().getPluginManager().registerEvents(new TheEnd(),**this**);

getServer().getPluginManager().registerEvents(new TeleportWhenEnemyHit(), **this**);

Zdrojový kód: Class Main – registrace eventů z tříd FoundStronghold, TheEnd a TeleportWhenEnemyHit

### Příkaz

Tato řočníková práce využívá i příkazy. Příkaz spustí daný kód jamile hráč napíše daný příkaz, co jsme zaregistrovali do chatu ve hře. Ve třídě, kde chceme mít příkaz, se implementuje CommandExecutor[[5]](#footnote-5), který bere několik proměnných. Hráče, který příkaz poslal a co poslal. Příkazy se registrují do souboru .yml následujícím způsobem

commands:

  pluginEnable:

    description: Zapne plugin

    usage: /pluginEnable;

Zdrojový kód: .yml soubor

## Logika programu

Hlavní logika a myšlenka této ročníkové práce je určení polohy strongholdu, hráče a následné vypisování do chatu, zda se přiblížil za určitou vzdálenost či ne. Zde je použit PlayerMoveEvent. Event, který spustí metodu vždy, když se hráč pohne. Pokud se hráč za danou vzdálenost přiblížil, do chatu[[6]](#footnote-6) se vypíše „přihořívá“. Naopak pokud se hráč od původní polohy vzdálil, do chatu se vypíše „samá voda“. Jakmile se hráč přiblíží na vzdálenost 300 bloků od stronholdu, začne se vypisovat přesná vzdálenost. Přiblíží-li se na vzdálenost 100 bloků, vypisování vzdálenosti přestává a kolem hráče se začnou objevovat zombies, které bude muset zabít, aby postoupil dále. Zombies je 11 a objevují se po jednom. Program detekuje, že hráč zombie zabil. Vypíše mu, kolik mu jich ještě zbývá a obejví se další. Následně hráč bude mít za úkol najít portál a skočit do něj, tím se hra ukončí a hráč se objeví v protiatomovém krytu. Na toto má omezený čas, který se vypočítává podle vzdálenosti od strongholdu + 10 minut. Pokud se hráč nestihne do portálu dostat v daný čas, prohrál a plugin se restartuje. Cestou hráče potká jedna nepříjemnost. Jakmile ho udeří, dále jen „hitne“, jakýkoliv enemy, hráče to teleportuje na náhodnou lokaci v okruhu 100 bloků po X a Z souřadnici.

### Určení vzálednosti

Jak již bylo zmíněno výše. Zde byl použit PlayerMoveEvent. Hlavní problém v této oblasti bylo zajistit, aby program detekoval, zda-li se hráč pohnul o nějakou vzdálenost, protože tento Event registruje pohyb i v situaci, kdy hráč pohne myší. K řešení stačila jedna podmínka, kde se porovná, zda se lokace, na kterou se hráč nachází rovná té, na které hráč byl.

**if**((event.getTo().getX() != event.getFrom().getX()) && (event.getTo().getZ() != event.getFrom().getZ())) {

Zdrojový kód: Class PlayerMovement - podmínka pohybu hráče

Poté už stačí jen porovnat poslední uloženou lokaci se současnou.

### Spawnování[[7]](#footnote-7) zombies

Zde je inicializovaný hráč a entita. Podmínkou se dá zjistit, jestli to byl hráč, kdo danou entitu zabil. Zde byl problém určit, jestli entita, kterou hráč zabil, je opravdu zombie, byl spawnut v první řadě. K řešení stačilo označit spawnuté zombies pomocí nametagu[[8]](#footnote-8) a následné podmínce, jestli má daná entita nametag. Volně ve hře se entity s nametagem nespawnují. Tím pádem je zaručené, že entita, která má nametag, je ten uměle zrozený zombie. Poté již stačí jen zjistit lokaci hráče, spawnout dalšího zombie a přičíst zabitého zombie.

**if** (killer instanceof Player) {

**if**(entity.isCustomNameVisible()) {

**if** (overallCount < 10) {

Zdrojový kód: Class FoundStronghold – zombies

### Inventář a nastavení hráče

Když se hráč připojí, nastaví se mu předměty potřebné k dohrání pluginu. Zde byl využit event, který se spustí po připojení hráče. Pokud již hráč na mém serveru hrál, vše se smaže a nastaví znovu.

### Reset pluginu

Zde je CommandExecutor, který resetuje plugin po zadání příkazu /reloadplugin. Tato funkce je zde proto, aby hráč mohl resetovat celou hru, pokud by již nechtěl pokračovat, nebo by nastala by nějaká neobjevená chyba v programu.

### Vypisování příběhu a časový limit

Původně bylo vypisování příběhu řešeno za pomocí regexu odděleným mezerou a Thread.sleep, ale Thread.sleep uspí celý server, což byl velký problém. Konečné řešení je přes funkci z frameworku minecraft development a textu, který jsem si rozdělil po slovech do pole.

Časovač byl vyřešen obdobným způsobem.

### Teleportace po hitu entity

Zde je opět inicializovaný hráč, entita a k tomu potřebná lokace hráče. Zde jen stačilo zjistit, jestli entita hitla hráče, vzít lokaci hráče a přidat k ní náhodnou hodnotu na X a Y souřadnici. Přesto se zde objevili 2 problémy.

Jak zjistit souřadnici výšky na nové lokaci a jak vypnout tento event, jakmile se začnou spawnovat zombies.

#### Jak zjistit souřadnici výšky

Na nové lokaci, kam se hráč bude teleportovat si je nadefinovaný blok, na kterým hráč později bude teleportován. Zde je funkce .getHighestBlockAt, do ní se dá vložit lokace, na kterou se hráč bude teleportovat. Poté stačilo jen přičíst daný blok na Y souřadnici.

Location newlocation = currlocation.add(Math.random() \* 100, 0, Math.random() \* 100);

Block Yheight = player.getWorld().getHighestBlockAt(newlocation);

newlocation.setY(Yheight.getY() + 1);

player.teleport(newlocation);

Zdrojový kód: Class TeleportWhenEnemyHit – určení lokace

#### Vypnutí eventu

Konečné řešení bylo vzít vzdálenost hráče od stongholdu. Jakmile se hráč přiblíží na 120 bloků, event se vypne přes podmínku a již se znovu nezapne.

### Konec hry

Tento event se spustí, když hráč skočí do portálu, tudíž změní svět. Hráče to teleportuje do předem vytovřené místnosti „protiatomového bunkru“ a vypíše se závěrečný text.

# Závěr

Hlavním cílem této práce bylo vytvořit funkční příběhový plugin, který by si hráči rádi zahráli. K tomu bylo potřebné se naučit pracovat s frameworky a efektivněji hledat dokumentaci na internetu.

Na začátku této práce bylo nutné si určit, jak by daný příběh měl vypadat. Inspirace vnikla ze současné situace ve světě. Podle příběhu byl následně programovaný celý plugin.

Přes spoustu překážet a problémů se úspěšně podařilo vytvořit funkční příběhový plugin, který nezatěžuje server a je uživatelsky přátelský.

# Zdroje

Úprava obrázků: CtrlV.link | Fastest ScreenShot and PrintScreen online. CtrlV.link | Fastest ScreenShot and PrintScreen online [online]. Dostupné z: <https://ctrlv.link/>

Inspirace různých částí kódu - Stack Overflow - Where Developers Learn, Share, & Build Careers. Stack Overflow - Where Developers Learn, Share, & Build Careers [online]. Dostupné z: <https://stackoverflow.com/>

Dokumentace k frameworku - Bukkit . Bukkit [online]. Dostupné z: <https://dev.bukkit.org/>

Spigot server - Downloading > spigot-1.18.jar. Get Bukkit | Download CraftBukkit 1.18.2 | Download Spigot 1.18.2 [online]. Copyright © 2022 GetBukkit All Rights Reserved. [cit. 07.04.2022]. Dostupné z: <https://getbukkit.org/get/3d9266f4af6057da7696026b2c8ab59f>

Řešení problémů - Dostupné z: <https://www.spigotmc.org/>

# Přílohy

## Seznam obrázků

Obrázek 1: Připojení na server [vlastní zdroj]

Obrázek 2: Příkaz /pluginenable [vlastní zdroj]

Obrázek 3: Příkaz /storyenable [vlastní zdroj]

1. Softwarová struktura, která slouží jako pomoc při programování a vývoji projektů [↑](#footnote-ref-1)
2. Počítačová hra v otevřeném světě [↑](#footnote-ref-2)
3. Vývojové prostředí pro programování v programovacích jazycích [↑](#footnote-ref-3)
4. Open-source java projekt, na kterém uživatelé mohou provozovat server s rozšířeními (pluginy) [↑](#footnote-ref-4)
5. Poskytuje funkce pro spouštění operátorů [↑](#footnote-ref-5)
6. Textová forma komunikace [↑](#footnote-ref-6)
7. Spawn – zrození se [↑](#footnote-ref-7)
8. Jmenovka (viditelné jméno nad hlavou entity) [↑](#footnote-ref-8)