



Minecraft Hacking für Anfänger

Peter Hrenka

Linux Tag Tübingen 2019

6. Juli 2019

Über mich

- Linux Anwender seit 1995
- Studium Informatik und Mathematik in Tübingen
- Softwareentwickler C++, python, OpenGL
- regelmäßig auf OpenSource Konferenzen anzutreffen
- Programmiersprachenjunkie
- Vater von 3 Kindern

1 Motivation

- Pro Minecraft
- Contra Minecraft

2 Modding

- Begriff
- Modding von Minecraft
- Modding Frameworks

3 Live Hacking

- Fliegender Besen
- Spiel des Lebens



Warum Minecraft?

- Ist bei raspian schon mit dabei (uralte Version)
- Läuft auch in den neueren Versionen unter Linux
- Wird noch weiterentwickelt
- Läuft auch auf (ur-)alter Hardware
- Stressfreier Kreativmodus
- Fokus auf Erforschen, Ausprobieren, Bauen
- Multikinder/playerfähig
- Halbwegs friedfertig
- Meine Kinder fanden schon die Raspberry Version toll



Minecraft und OpenSource

- Microsoft mag jetzt OpenSource
- Mojang ist jetzt auf github
- <https://github.com/Mojang>
 - Brigadier - Befehlsparser
 - DataFixerUpper - Konvertierungstool
- Nützlich aber nicht essentiell



Minecraft ist doch böse!

- Keine Version ist OpenSource(tm)
- Kostet Geld (aktuell ca. 25 Euronen)
 - Pro gleichzeitigem (!) Spieler
- Die wollen einem eigentlich nur die (nicht-Java) Windows-Version andrehen
 - Die Java-Version gibt es als kostenlose Dreingabe
- Ist voll Mainstream
- Schlechte Grafik

Begriff Modding

- “Modding” = Modifizieren eines vorhandenen, fertiggestellten Spiels, um den Funktionsumfang zu erweitern
- “Patches” = Modifizieren eines Spiels, um Fehler zu beseitigen
- “Cheaten” = Modifizieren eines Spiels, um seinem Spieler einen unfairen Vorteil zu verschaffen
- Modding wird oft von vielen Spiele-Engines durch eine eingebaute Skriptsprache unterstützt
- ... nicht so bei Minecraft
- Aber: Dank Java kann man trotzdem prima Modding betreiben

Modding in Minecraft

- Decompilieren des Minecraft-Codes
- Patches des Decompilierten Codes
 - Einfügen von Callbacks
 - Code wieder lesbarer machen
- Neu kompilieren
- Vorgehen wird von Mojang/Microsoft toleriert

Modding-Frameworks

- Bukkit
 - Beliebtes Server-Framework
 - Wird nicht mehr weiterentwickelt
- Spigot(MC)
 - Nachfolger von Bukkit
 - auch für aktuelle Versionen
- MinecraftForge
 - Client- und ServerMod
 - auch für aktuelle Versionen
 - Erlaubt kombinieren von mehreren Mods
 - <https://github.com/MinecraftForge/MinecraftForge>

Demo


Umgebung einrichten

- `https://files.minecraftforge.net/maven/net/minecraftforge/forge/index_1.14.2.html`
- MDK herunterladen und extrahieren
- `./gradlew genEclipseRuns`
- Läuft ca. 7 min auf meinem Notebook
- Belegt 22 MB
- `./gradlew eclipse`
- `snap run eclipse`
- File → Import → Gradle → existing Gradle Project
- Verzeichnis wählen → (Next)*
- Run → runClient
- Juhu wir haben ein ExampleMod, tut aber nichts

Anpassungen

- `build.gradle`
 - `Version = 0.0.1`
 - `Group = 'org.tuebix.2019'`
 - `archivesBaseName = 'tuebix2019mod'`
 - `query-replace 'example' → 'tuebix2019mod'`
- Kein Unterschied
- `src/main/resources/META-INF/mods.toml` editieren
 - `modid = 'tuebix2019'`
- Auch in `ExampleMod.java @mod` anpassen

Harry und Bibi lassen grüßen

- Neues “Item” Besen
 - Requirement SWRS_0001 : Es muss einen Besen als Item geben
 - Requirement SWRS_0002 : Schönes Bildchen
- 
- Requirement SWRS_0003 : Wenn man auf dem Besen sitzt, kann man fliegen



Implementieren ist schwer

- Wo kann man ein neues Item registrieren?
- Nachbauen, was in `onBlocksRegistry` steht
- Viele Java 'Probleme'
- Probleme-Reiter öffnen, mit Maus über Problem schweben, Quick-Fix
- Tut, aber leider kein Effekt

Fliegen ist schwer

- Funktionalität sollte eigentlich da sein (im Kreativ-Modus)
- Wir bauen uns einen Handler für `WorldTickEvents`
- Durchsuche alle Spieler (`ServerPlayerEntity`)
- Durchsuche `armorInventory`
- Falls Item unser Besen ist
 - Setze Abilites `allowFlying` und `isFlying` auf `true`



- Neues Item “Ankh”
- Requirement SWRS_0004 : <https://bitstorm.org/gameoflife/>
- Requirement SWRS_0005 : Wenn man das “Ankh” in der linken Hand hält, spielen die Blöcke auf dem Boden das Game of Life.
- Requirement SWRS_0006 : Bitte nicht zu schnell



Leben ist auch schwer

- Ähnliche Item-Logik wie beim Besen
- Verwende jedoch `offHandInventory`
- Modifikation der Welt
 - Zentrale Klasse `BlockPos` (immutable)
 - API `World.setBlockPos` und `World.getBlockPos`
- Suche Ebene nach unten, bis nicht-Luft-Block kommt
- Berechne entstehende und sterbende Blöcke : in Listen merken
- Danach an gemerkten Positionen Blöcke erschaffen bzw. mit Luft-Blöcken überschreiben

Persönliches Fazit

- Offizielle Doku ist nicht für Anfänger geeignet
- Nur wenige OpenSource Mods verfügbar, wo man abschauen kann
- Klassenbrowser hilft ungemein
- Auch “spickeln” im Source ist oft notwendig
- Keine stabile Modding API, man muss direkt Minecraft-APIs verwenden
- API in ständigem Fluss (meist nur kosmetisch ItemArmor ↔ ArmorItem)
- QuickFix-Logic sehr praktisch



Vielen Dank!

Fragen?