raspberry pi minecraft server

Lauren Emmenegger, Janik Meier

BBB

Projekttag

Metainformationen

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Autor** | **Bemerkung** |
| 2.12.2020 | 1.0 | L. Emmenegger, J. Meier | Erstellung des Dokumentation-Grundgerüsts |
| 9.12.2020 | 1.1 | L. Emmenegger, J. Meier | Tools, Anforderungsanalyse, Pflichtenheft, Fazit gemacht |
| 16.12.2020 | 1.1 | L. Emmenegger, J. Meier | Entscheidungsmatrix erstellt |
| 6.1.2021 | 1.2 | L. Emmenegger, J. Meier | An der Realisation dokumentiert |
| 13.1.2021 | 1.3 | L. Emmenegger, J. Meier | Testspezifikationen hinzugefügt |
| 20.1.2021 | 1.4 | L. Emmenegger, J. Meier | Testen, Dokumentation des Realisierens |

Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **In Bearbeitung** | **In Prüfung** | **Freigegeben** |
| 2.12.2020 | X |  |  |
| 9.12.2020 | X |  |  |
| 16.12.2020 | X |  |  |
| 6.01.2021 | X |  |  |
| 13.01.2021 | X |  |  |
| 20.01.2021 |  | X | X |

Inhalt

[1 Informieren 4](#_Toc62051102)

[1.1 Projektantrag 4](#_Toc62051103)

[1.2 Projektantrag-E-Mail 5](#_Toc62051104)

[1.3 Idee 6](#_Toc62051105)

[1.4 Anforderungsanalyse 6](#_Toc62051106)

[1.5 Tools 6](#_Toc62051107)

[1.6 Zeitplan 7](#_Toc62051108)

[2 Planen 8](#_Toc62051109)

[2.1 Pflichtenheft 8](#_Toc62051110)

[2.2 Situation 8](#_Toc62051111)

[2.3 Ziele 8](#_Toc62051112)

[2.4 Anforderungen 8](#_Toc62051113)

[2.5 Tätigkeitsliste 9](#_Toc62051114)

[2.6 Testspezifikationen 10](#_Toc62051115)

[3 Entscheiden 13](#_Toc62051116)

[3.1 Entscheidungsmatrix 13](#_Toc62051117)

[3.2 Fazit 13](#_Toc62051118)

[4 Realisieren 14](#_Toc62051119)

[4.1 Raspberry Pi konfigurieren / Minecraft Server aufsetzen 14](#_Toc62051120)

[4.2 Zwischenprojekt: Hamachi Spigot Minecraft Server 15](#_Toc62051121)

[4.3 Zwischenprojekt: Azure Minecraft Server 15](#_Toc62051122)

[4.4 Fazit 16](#_Toc62051123)

[5 Kontrollieren 17](#_Toc62051124)

[5.1 Testprotokoll 17](#_Toc62051125)

[5.2 Testbericht 17](#_Toc62051126)

[6 Auswertung 18](#_Toc62051127)

[6.1 Selbstreflexion 18](#_Toc62051128)

[6.2 Fazit 19](#_Toc62051129)

# Informieren

## Projektantrag

|  |
| --- |
| **Projektname:** |
| Raspberry Pi Minecraft Server |

|  |
| --- |
| **Beschreibung:** |
| Wir wollen einen Minecraft Server auf einem Raspberry Pi laufen lassen. Ob wir einen Spigot oder Paper Minecraft Server machen ist noch unklar.  Der Server soll zuerst über einen Switch im lokalen Netzwerk erreichbar sein, wenn der Server stabil läuft möchten wir noch netzwerkübergreifenden Zugang mit Hamachi einbauen.  Zudem wollen wir mit einer Whitelist in der Lage sein, nur spezifisch ausgewählte Spieler auf den Server zu lassen. Dabei möchten wir uns Kenntnisse zur Bedienung des Raspberry Pis und dem Aufsetzen eines Minecraft Servers aneignen.  Ziele:   * Kenntnisse über die Benutzung eines Raspberry Pis erlangen * Kenntnisse über das Aufsetzen eines Minecraft Servers mit Raspberry Pi erlangen * Kenntnisse über die Verwendung von Hamachi erlangen |

|  |
| --- |
| **Module** |
| - |

|  |
| --- |
| **Zusätzlich benötigtes Material:** |
| Raspberry PI min 4 GB Ram |
| Netzteil |
| SD-Karte min 32GB |
| Tastatur + Maus + Monitor |
| HDMI Kabel, USB-Kabel, Switch, 3 Ethernet Kabel |

|  |  |  |
| --- | --- | --- |
| **Auftraggeber:** | **Weitere Kontaktpersonen:** | **Begleitperson BBB:** |
| N/A | N/A | Sascha Fiechter |

|  |  |
| --- | --- |
| **Projektleiter:** | **Teammitglieder:** |
| Lauren Emmenegger | Janik Meier |

|  |  |
| --- | --- |
| **Termin Projektstart:** | **Termin Projektende:** |
| 02.12.20 | 06.01.21 |

## Projektantrag-E-Mail



## Idee

Beim überlegen, was wir als nächstes Projekt machen könnten, hatten wir die Idee, einen Minecraft Server über einen Raspberry Pi laufen zu lassen. Der Server wird im lokalen Netzwerk erreichbar sein. Da Minecraft Multiplayer über das BBBaden Netzwerk nicht läuft, werden wir zu diesem Zweck ein kleines Netzwerk mit Switch und Ethernetkabeln erstellen. Falls wir den Server zum Laufen bringen und noch Zeit übrig haben, möchten wir, dass man sich auch per Hamachi von der Ferne ins gleiche Netzwerk einschalten kann.

## Anforderungsanalyse

|  |  |
| --- | --- |
| **Nr.** | **Anforderung** |
|  | Wir können uns gegenseitig über den Switch anpingen. |
|  | Wir können den Raspberry Pi per Laptop steuern, anstatt per angeschlossene Tastatur, Monitor und Maus (Headless). |
|  | Wir können dem Minecraft Server joinen, wenn wir im gleichen Netzwerk sind. |
|  | Wir haben konstant 30 Fps auf dem Minecraft Server, wenn zwei Spieler drauf sind. |
|  | Wenn man ein Entity schlägt, wird dies innerhalb einer halben Sekunde angezeigt. |
|  | Der Raspberry Pi verwendet die neueste Version von Raspbian. |

## Tools

|  |  |
| --- | --- |
| **Nr.** | **Tools** |
| 1 | Raspberry Pi 3 |
| 2 | Software PuTTY |
| 3 | Externe Tastatur |
| 4 | Externer Monitor |
| 5 | Maus |
| 6 | Minecraft 1.16.4 |
| 7 | HDMI Kabel |
| 8 | USB-Kabel |
| 9 | Switch |
| 10 | 3 Ethernet Kabel |

## Zeitplan



# Planen

## Pflichtenheft

Wir möchten einen 1.16.4 Minecraft Spigot Server über einen Raspberry Pi laufen lassen. Wir sollten konstant mindestens 30 Fps haben. Den Server werden wir im lokalen Netzwerk laufen lassen, wenn dies funktioniert, wollen wir den Minecraft Server auch von externen Netzwerken erreichbar machen, mithilfe von Hamachi.

## Situation

Ausgangslage

Wir interessieren uns für Minecraft und Einplatinencomputer. Diese Aspekte möchten wir in diesem Projekt zusammenfügen.

Ist-Zustand

Wir haben uns bereits ein paar Tutorials zur Aufsetzung eines Minecraft Servers auf einem Raspberry Pi angeschaut und wissen ungefähr, wie man einen Raspberry Pi bedient.

Soll-Zustand

Das Projekt soll bis am 6ten Januar fertig sein, heisst, wir haben ein lokales Netzwerk mit Switch und Ethernet Kabeln erstellt und haben mit konstant mindestens 30 Fps gleichzeitig zusammen auf dem Minecraft Server gespielt.

## Ziele

* Einen Minecraft Server auf einer Raspberry Pi aufsetzen und laufen lassen.
* Einen Raspberry Pi per Laptop mit Putty steuern können (Headless).
* Wir wissen, wie man sich im Raspberry Pi Interface zurechtfindet.
* Mit Hamachi aus einem anderem Netzwerk auf den Server connecten.

## Anforderungen

Der Server soll über ein lokales Netzwerk und über ein externes Netzwerk mithilfe von Hamachi erreichbar sein. Zudem sollte der Server mit zwei Spielern konstant mit 30 Fps laufen. Der Raspberry Pi soll über die Software PuTTY gesteuert werden können, indem wir uns per IP-Adresse mit dem Pi verbinden.

## Tätigkeitsliste



## Testspezifikationen

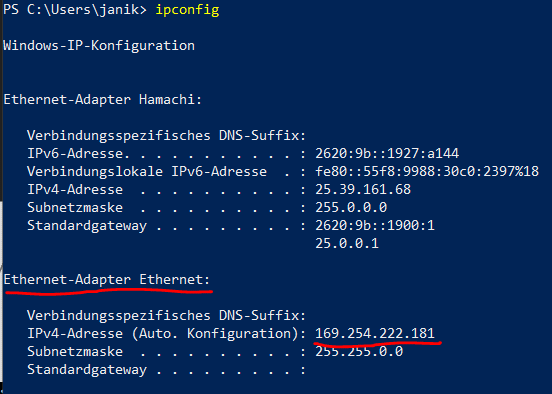
**Nummer: 1**

**Getestete Anforderung:**

Wir können uns gegenseitig über den Switch anpingen.

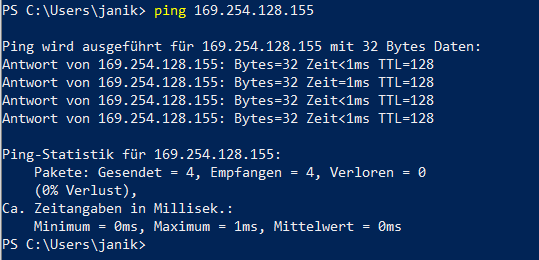
**Voraussetzung:**

Der Switch ist mit Strom versorgt, unsere zwei Laptops sind über ein Ethernet Kabel mit dem Switch verbunden. Man ist mit keinem WLAN verbunden. Man hat zudem seine eigenen IP-Adresse vom Ethernet-Adapter mit der Applikation PowerShell und dem Command «ipconfig» ermittelt.



**Eingabe:**

Man gibt in der Applikation PowerShell «ping» «IP vom Kollege»



**Ausgabe:** Es öffnet sich ein neues Fenster in Firefox.

**Nummer: 2**

**Getestete Anforderung:**

Wir können den Raspberry Pi per Laptop steuern, anstatt per angeschlossene Tastatur, Monitor und Maus (Headless).

**Voraussetzung:**

Der Raspberry Pi hat SSH und Console Autologin aktiviert.

**Eingabe:**

PuTTY starten und im Feld “Hostname (oder IP-Adresse)” die IP-Adresse des Pi eingeben (169.254.207.108). Auf den Knopf “Open” drücken. “pi” eingeben und Enter drücken. “JodokGTA” eingeben und Enter drücken.

**Ausgabe:**

Sie steuern ein Terminal auf dem Raspberry Pi von Ihrem Laptop aus.

**Nummer: 3**

**Getestete Anforderung:**

Wir können dem Minecraft Server joinen, wenn wir im gleichen Netzwerk sind.

**Voraussetzung:**

Testfälle 1 und 2 erfolgreich abgeschlossen und Minecraft ist auf Ihrem Gerät installiert. Auf dem Raspberry Pi ist ein heruntergefahrener Minecraft Server.

**Eingabe:**

Über PuTTY folgenden Befehl auf dem Raspberry Pi ausführen: “java -jar -Xms512M -Xmx1008M server.jar nogui” Sobald ausgegeben wird, dass der Server jetzt läuft, Minecraft starten. Bei Minecraft auf «Multiplayer» klicken, dann auf «Direkt verbinden», dies eingeben: «169.254.207.108:25565» und «verbinden» klicken.

**Ausgabe:**

Sie verbinden sich mit dem Minecraft Server.

**Nummer: 4**

**Getestete Anforderung:**

Wir haben konstant mindestens 30 Fps auf dem Minecraft Server, wenn zwei Spieler drauf sind.

**Voraussetzung:**

Testfall 3 erfolgreich abgeschlossen.

**Eingabe:**

Sich mit dem Minecraft Server verbinden. Wenn man in der Minecraftwelt ist, «F3» klicken.

**Ausgabe:**

Im nun erscheinenden Text links oben in der zweiten Zeile die erste Zahl (FPS Anzeige) ist immer über 30.

**Nummer: 5**

**Getestete Anforderung:**

Wenn man ein Entity schlägt, wird dies innerhalb einer halben Sekunde angezeigt.

**Voraussetzung:**

Testfall 3 erfolgreich abgeschlossen.

**Eingabe:**

Dem Minecraft Server beitreten. Ein Tier oder Monster suchen und dieses mit einem Klick auf der linken Maustaste schlagen.

**Ausgabe:**

Innerhalb einer halben Sekunde blinkt das Tier oder Monster rot auf.

**Nummer: 6**

**Getestete Anforderung:**

Der Raspberry Pi verwendet die neueste Version von Raspbian.

**Voraussetzung:**

Testfall Nr. 2 erfolgreich abgeschlossen

**Eingabe:**

PuTTY starten und im Feld “Hostname (oder IP-Adresse)” die IP-Adresse des Pi eingeben (169.254.207.108). Auf den Knopf “Open” drücken. “pi” eingeben und Enter drücken. “JodokGTA” eingeben und Enter drücken.

Eingeben: cat /etc/os-release

**Ausgabe:**

Text erscheint im Konsolenfenster. Bei «VERSION\_ID» steht «10».

# Entscheiden

## Entscheidungsmatrix

|  |  |
| --- | --- |
| **Vanilla Server**  **Pro:**   * Schnelle Updates * Weniger Bugs   **Contra:**   * Benötigt mehr Rechenleistung * Keine Plugins möglich | **Spigot Server**  **Pro:**   * Benötigt weniger Rechenleistung * Plugins   **Contra:**   * Langsame Updates * Mehr Bugs |



## Fazit

Wir haben uns für den Spigot Server entschieden. Hauptsächlich wegen dem Performance Boost, den man, im Vergleich zum Vanilla Server bekommen würde. Der Fakt, dass Updates für einen Spigot Server nicht so schnell verfügbar sind ist uns egal, da wir nicht sofort die neuste Version benötigen.

# Realisieren

## Raspberry Pi konfigurieren / Minecraft Server aufsetzen

Zuerst haben wir mithilfe vom Raspberry Pi Imager das neuste Raspbian auf die Micro SD-Karte geflashed. Danach führten wir das Setup vom OS aus.

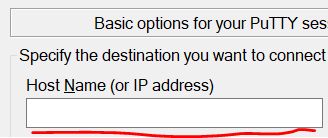
1. Terminal öffnen und folgenden Befehl eingeben:

«sudo apt-get install netatalk screen avahi-daemon»

Dieser Befehl installiert Packages, die man für den Minecraft Server braucht.

1. Im Terminal den Befehl «screen -list» eingeben. Wenn Sie die Nachricht «No Sockets found» bekommen haben, hat es geklappt, wenn «No screen Kommand» angezeigt wird wiederholen sie die Installation vom ersten Schritt.
2. Öffnen sie das Konfigurationstool des Pi mit dem Kommand «sudo raspi-config» und nehmen sie folgende Änderungen vor.

|  |  |
| --- | --- |
| 1. Expand FileSystem | 6. Advanced Options 🡪 A1 Expand Filesystem |
| 2. Change Use Password | 1 System Options 🡪 S3 Password |
| 3. Enable Boot to Desktop/Scratch | 1 System Options 🡪 S5 Boot / Auto Login 🡪 B2 Console AutoLogin |
| 4. Internationalisation Options | 5 Localisation Options 🡪 L2 |
| 5. Overclock | 4 Performance Options 🡪 P1 Overclock |
| 6. Memory Split | 4 Performance Options 🡪 P2 GPU Memory 🡪 Speicher auf mind. 16 setzen |
| 7. SSH | 3 Interface Options 🡪 P2 SSH 🡪 ja |

1. Auf dem Raspberry Pi in der Shell den Kommand «hostname -I» ein, um die Ip-Adresse zu ermitteln.
2. Installieren von Putty auf einem PC, den sie verwenden möchten, um den Raspberry Pi headless zu bedienen.
3. Geben Sie hier  die IP vom Raspberry Pi ein und drücken Sie dann unten auf Open. Melden sie sich an, Sie können nun den Raspberry Pi aus der Shell bedienen.
4. Geben sie nun den Befehl «mkdir minecraft && cd minecraft» ein, damit erstellen Sie ein Ordner mit dem Namen minecraft und begeben sich auch direkt in den Ordner.
5. Laden sie die Server.jar Datei mit dem Befehl «sudo wget <https://launcher.mojang.com/v1/objects/1b557e7b033b583cd9f66746b7a9ab1ec1673ced/server.jar>»
6. Nach dem Download der server.jar Datei geben Sie folgenden Befehl ein «java -jar server.jar» um den Server aufzusetzen.
7. Starten Sie mit dem Befehl «java -Xmx1024M -Xms1024M -jar minecraft\_server.1.16.5.jar nogui» den Server. Dadurch wird eine Datei namens eula.txt erstellt.
8. Öffnen Sie mit dem Befehl «nano eula.txt» und ändern sie eula=false zu eula=true, um die Eula zu akzeptieren
9. Starten Sie den Server erneut mit dem Befehl «java -Xmx1024M -Xms1024M -jar minecraft\_server.1.16.5.jar nogui» nach einer langen Wartezeit sollte die Eingabeaufforderung wiedererscheinen, dies bedeutet, dass der Server einsatzbereit ist.

## Zwischenprojekt: Hamachi Spigot Minecraft Server

Am 09.12.2020 hatten wir am Projekttag Homeoffice und konnten deswegen nicht an unserem Raspberry Pi in der Schule weiterarbeiten. Da hatten wir die Idee, von Zuhause aus einen Spigot Server aufzusetzen. Wir haben den Server auf einem Windows PC erstellt und uns über Hamachi damit verbunden.

**Was ist Hamachi?**

Hamachi ist ein einfach zu konfigurierender, proprietärer VPN-Client. Programme, die sonst nur über lokale Netzwerke funktionieren, lassen sich so auch über das Internet nutzen, zum Beispiel Computerspiele. Wenn man eine Netzwerkgruppe erstellt, simuliert diese eine lokale Netzwerkverbindung zwischen mehreren Geräten.

## Zwischenprojekt: Azure Minecraft Server

Da wir während dem Projekt Homeoffice hatten und am 6.1.2021 nicht mit dem Raspberry Pi arbeiten konnten, weil dieser in der Schule verweilte, bekamen wir vom Lehrer einen Zwischenauftrag.

In diesem Auftrag sollten wir einen Minecraft Server auf einer virtuellen Maschine über das Azure-Portal erstellen. Dazu haben wir eine Ubuntu VM erstellt, in welcher wir Verbindungen über SSH zuliessen. Somit konnten wir uns mit Hilfe von PuTTY mit der VM verbinden und auf dieser den Minecraft Server aufsetzen:

1. Ubuntu auf die neueste Version updaten sudo apt update und sudo apt upgrade
2. Benutzer für den Minecraft Server erstellen sudo adduser minecraftuser
3. Den neuen Benutzer zur sudo-Gruppe hinzufügen sudo usermod -aG sudo minecraftuser
4. Als den neuen Benutzer anmelden su – minecraftuser
5. Das wget Paket installieren sudo apt install wget
6. Das java Paket installieren sudo apt install openjdk-8-jdk -y
7. Erstelle ein Minecraft Verzeichnis und navigiere rein sudo mkdir minecraftdir && cd minecraftdir
8. Installiere die Minecraft Server-Software sudo wget <https://launcher.mojang.com/v1/objects/1b557e7b033b583cd9f66746b7a9ab1ec1673ced/server.jar>
9. Erweitere die Ausführberechtigung der Serverdatei sudo chmod +x server.jar
10. Ändere das eula.txt sudo nano eula.txt. Ersetze «false» mit «true».
11. Starte den Minecraft Server sudo java -Xmx1024M -Xms1024M -jar minecraft\_server.1.16.4.jar nogui
12. Verbinde dich mit dem Server über die VM-IP + Server Port(bei uns 51.103.139.62:25565)
13. Spiele!

## Fazit

Wir haben einen Minecraft Server auf Ubuntu und Windows erstellt, welchen wir mit unseren Accounts von nah und von fern beitreten konnten. Wir haben mit Azure einen virtuellen Ubuntu Server erstellt, mit dem wir uns per SSH mit PuTTY und Minecraft verbinden konnten. Ausserdem haben wir einen Raspberry Pi konfiguriert und darauf auch einen Minecraft Server erstellt, auf welchem wir spielen konnten. Den Raspberry Pi konnten wir des Weiteren auch headless steuern, das heisst, wir konnten von unseren Laptops her auf den Raspberry Pi zugreifen.

# Kontrollieren

## Testprotokoll

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nummer | Resultat | Datum | Tester | Unterschrift |
| 1 | Funktioniert nicht | 20.1.2021 | L. Emmenegger | L. Emmenegger |
| 2 | Funktioniert | 20.1.2021 | L. Emmenegger | L. Emmenegger |
| 3 | Funktioniert | 13.1.2021 | L. Emmenegger | L. Emmenegger |
| 4 | Funktioniert | 13.1.2021 | L. Emmenegger | L. Emmenegger |
| 5 | Funktioniert nicht | 13.1.2021 | L. Emmenegger | L. Emmenegger |
| 6 | Funktioniert | 20.1.2021 | L. Emmenegger | L. Emmenegger |

## Testbericht

Obwohl der Grossteil der Tests erfolgreich verlief, läuft das Produkt nicht wie gewünscht. Der Server stürzt schon nach wenigen Minuten ab, da der Raspberry Pi nicht genügend Leistung hat (nur 1GB Ram). Der Server kann nicht an den Kunden übergeben werden.

# Auswertung

## Selbstreflexion

**Janik Meier**

Mit meiner Leistung im Projekt bin ich zufrieden, da ich mein Vorwissen gut einbringen konnte und viele neue Dinge gelernt habe. Unteranderem weiss ich jetzt, wie man einen Raspberry Pi bedient, einen Minecraft Server in Ubuntu aufsetzt und wie man eine VM auf Azure einrichtet und verwendet.  
  
Das Homeoffice war für unser Projekt nicht sehr vorteilhaft, da die benötigte Hardware in der Schule gelagert wurde. So mussten wir uns kleine Ersatzprojekt für die Tage im Homeoffice ausdenken. Die Zusammenarbeit hat im Homeoffice und in der Schule gut funktioniert. Da unser Team nur aus zwei Personen besteht konnten wir uns immer gut absprechen und Entscheidungen wurden schneller gefällt.

Was ich gut gemacht habe:

* Ich konnte mein Vorwissen gut einbringen.
* Ich habe funktionierende Lösungen zu Problemen gefunden.

Was ich noch besser machen könnte:

* Mein Arbeitstempo während der Dokumentation verbessern.
* Mich weniger von meinem Umfeld ablenken lassen.

**Lauren Emmenegger:**

Ich bin hauptsächlich zufrieden mit meiner Arbeit.   
Ich kann jetzt Minecraft Server aufsetzten und habe meine Kenntnisse mit dem Linux CLI gestärkt. Dies waren meine ersten Erfahrungen mit einem Raspberry Pi und ich weiss, wie ich ihn headless betreiben kann.  
Im Homeoffice-Unterricht konnten wir zwar nicht mit dem Raspberry Pi arbeiten, dafür war die Umgebung ruhiger und wir konnten so konzentrierter arbeiten.

Was ich gut gemacht habe:

* Ich konnte auch im Homeoffice meist effizient arbeiten.
* Ich konnte gut mit meinem Team kommunizieren.

Was ich verbessern kann:

* Ich war manchmal ineffizient, da ich mich ablenken liess.

## Fazit

Im Verlauf von dem Projekt «Raspberry PI Minecraft Server» haben wir viel neues dazu gelernt. Wir wissen nun, was alles benötigt wird, um ein Minecraft Server auf einem Ubuntu Server aufzusetzen zudem können wir nun durch das neu erlangte wissen einen Raspberry Pi mit dem Betriebssystem Raspbian bedienen.

Das Projekt hat uns sehr viel Spass gemacht. Wir konnten uns bei Problemen immer gegenseitig unterstützen. Zudem arbeiteten wir sehr effizient. Das Dokumentieren des Projekts war das Einzige, was uns nicht so viel Spass gemacht hat.

Während unserem Projekt hatten wir zwei Mal Homeoffice, dadurch mussten wir unser Vorgehen etwas Anpassen, da die Hardware in der Schule gelagert wurde. Für die beiden Tage haben wir uns kleine Zwischenprojekte ausgedacht, da es für uns nicht möglich war, an unserem Hauptprojekt weiter zu arbeiten.

Die Zwischenprojekte haben uns auch sehr viel Spass gemacht und uns in gewisser Weise auch im Hauptprojekt weitergeholfen. Z.B. das Aufsetzen des Minecraft Servers auf dem virtuellen Ubuntu Server war sehr ähnlich wie das Aufsetzen des Servers auf dem Raspberry Pi.

Das Homeoffice an sich war kein grosses Problem, wir haben uns ziemlich schnell daran gewöhnt und gelernt, wie man effizient von Zuhause ausarbeitet.