
M.R in Minecraft
David Smith
@revodavid

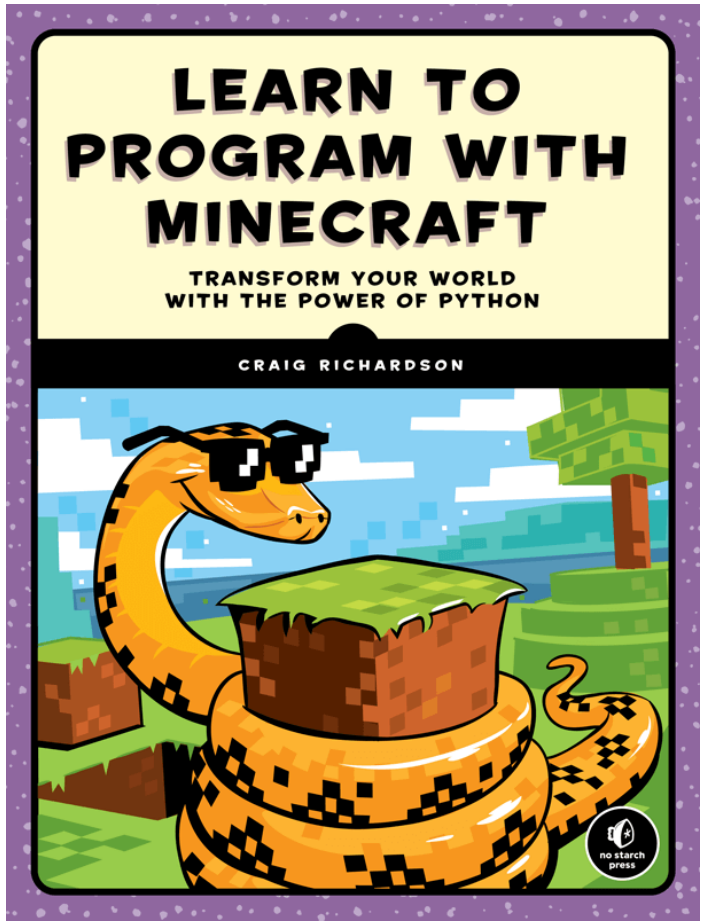
What is Minecraft?

- A 3-D exploration and building game
  - In “Survival” mode, the player crafts resources while defending against monster attacks
  - In “Creative” mode, it’s more like an immersive virtual Lego session
- Developed by Mojang
  - Acquired by Microsoft in 2014
- Hugely popular with under-15s

Microsoft

Why R in Minecraft?

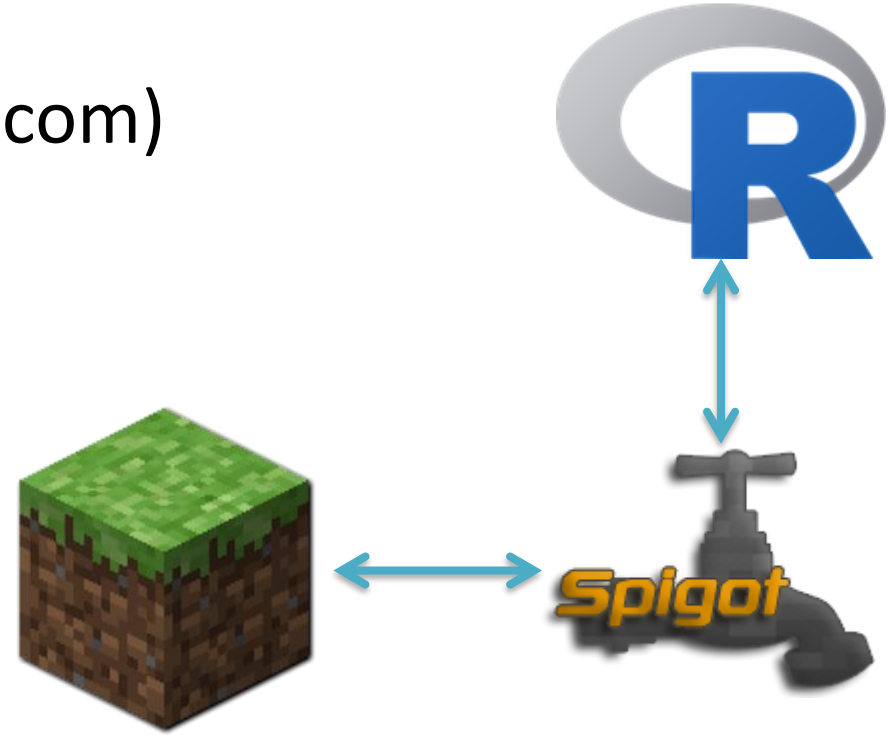
- Teach kids the R language
  - motivated by automating tedious tasks in Minecraft
- Extensible Client/Server architecture
  - World can be modified via server extensions
  - Language API provides a higher-level interface



Microsoft

You Will Need:

- Minecraft desktop (requires purchase)
  - Java version NOT Windows Store or Xbox/Mobile version
- Minecraft account (account.mojang.com)
- Spigot Minecraft server (free)
  - w/ RaspberryJuice plugin
- Desktop client running R
  - “miner” and “craft” packages



Microsoft

R packages

- **miner**: Basic functions for interfacing with a Minecraft server:
  - connect to Spigot server
  - inspect / modify world area (by blocks)
  - monitor player actions (movement, block interaction)
  - github.com/ropenscilabs/miner

- **craft**: Higher-level functions built on basic tools
  - Render a message / image / maze in the world
  - github.com/ropenscilabs/craft
  - Accompanying book: ropenscilabs.github.io/miner_book

R Programming with Minecraft

Brooke Anderson, Karl Broman, Gergely Daróczi, Mario Inchiosa, David Smith, and Ali Zaidi

2017-06-05

1 Introduction

2.3.2 The `miner` Dockerfile

The `miner` package includes a Dockerfile, which is a plain text file that gives Docker the recipe for setting up an appropriate container.

This file specifies the following steps that are needed to set up the required environment and run a Spigot Minecraft Server with the RaspberryJuice plug-in:

- Creates a directory called “minecraft” for the Minecraft server
- Downloads all required files to build a Spigot server (https://www.spigotmc.org) and saves them in the “minecraft” direction
- Builds the Spigot server
- Symlink for the built Spigot server?
- Accepts the End User License Agreement for Minecraft (“eula”) (see here to see what you are agreeing to with this step)
- Downloads the RaspberryJuice plugin (which we’re using for API access) to a subdirectory of the “minecraft” directory called “plugins”
- Install the RaspberryJuice plugin
- Open up the ports required to access the game (port 25565) and the API (4711)
- Start the Minecraft server, [explain options we’re using for that]

This Dockerfile is included in the `miner` package. To find it on your computer once you’ve installed the `miner` package, you can run:

DIY Docker build with VM in Azure

- Launch Ubuntu Data Science Virtual Machine
  - open ports 4711,25565 in/out (in Networking)

```
git clone https://github.com/ropenscilabs/miner
cd miner/inst
docker build -t minecraft .
docker run -ti --rm -p 4711:4711 -p 25565:25565 minecraft
```

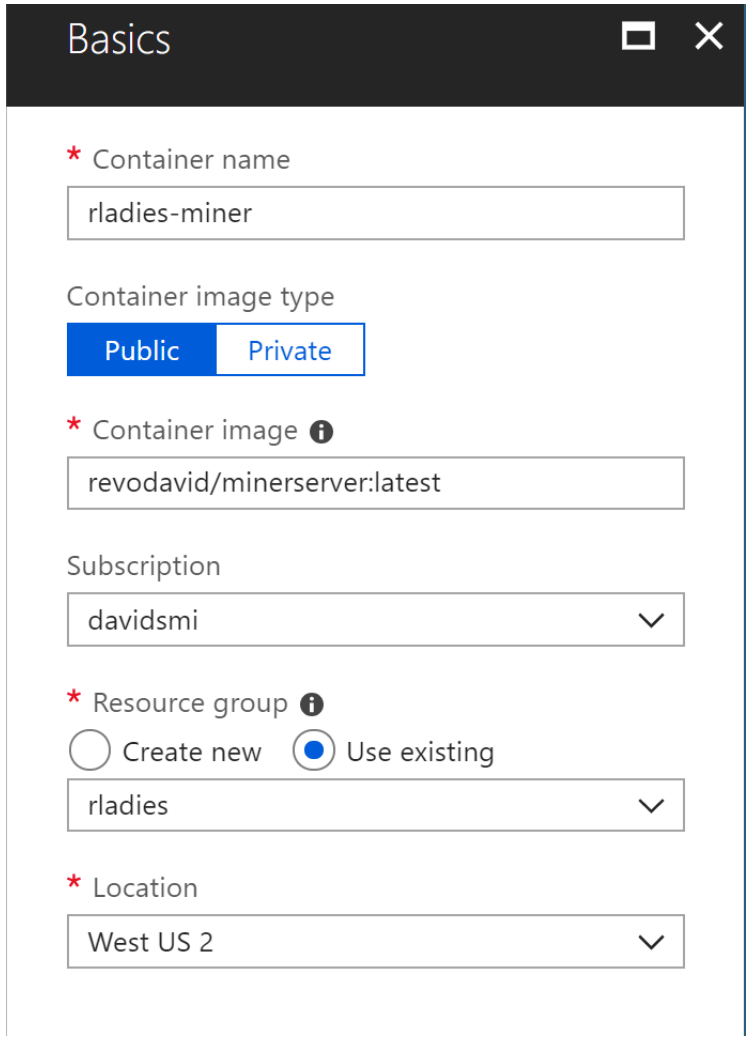
```
> defaultmode creative
> op [your_username]
```

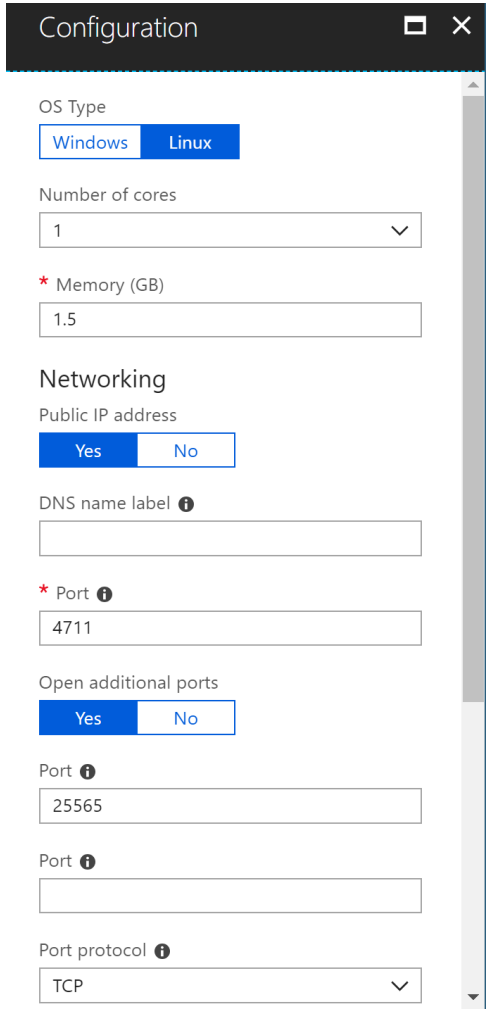
Docker pull

- In any Ubuntu docker machine or VM
  - (still need to open ports 4711,25565 in/out)

```
sudo docker run -ti --rm -p 4711:4711 -p 25565:25565 \
  revodavid/minerserver:latest
```

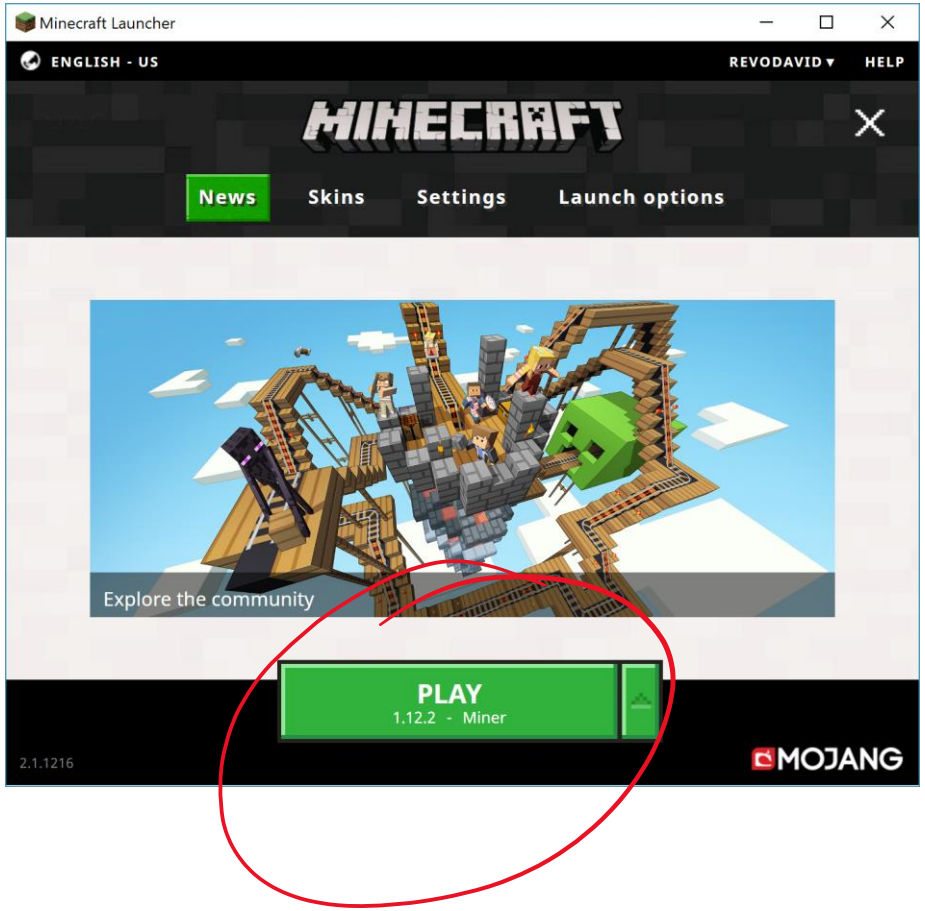
Azure Container Instances

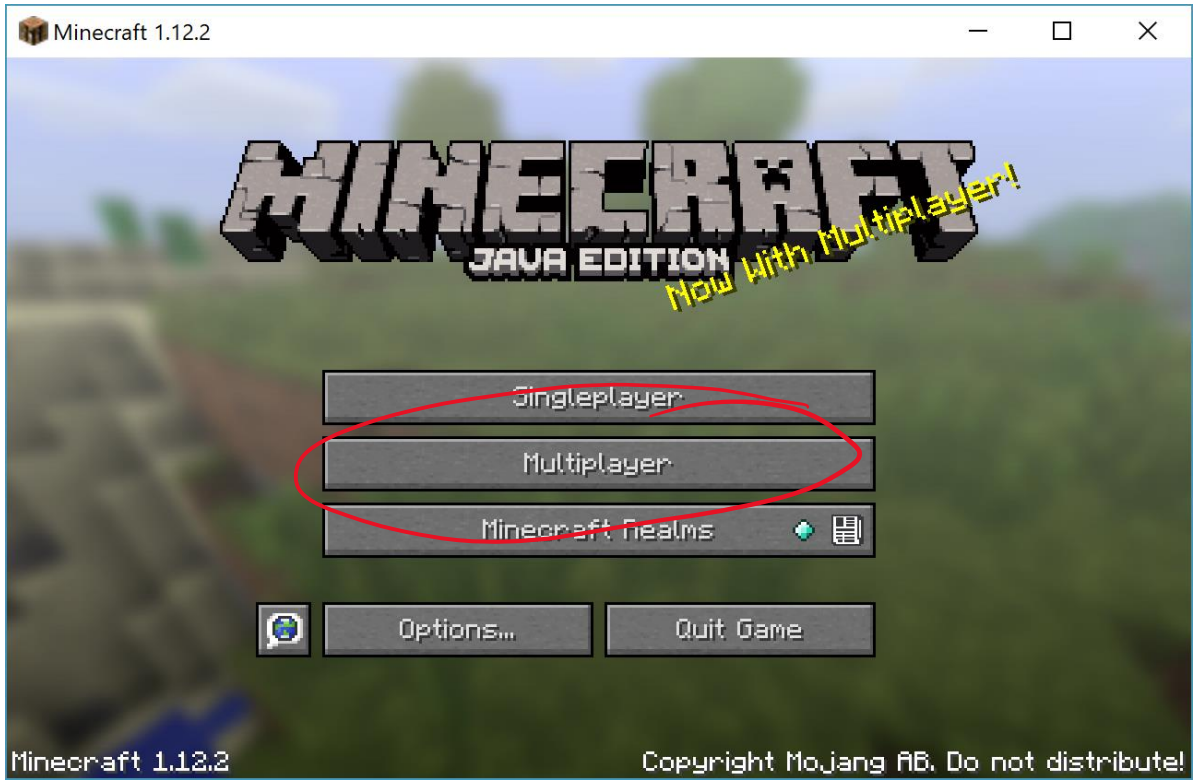




Microsoft

Launch Minecraft





Microsoft

Connect using miner package

- In R:

```
library(devtools)
install_github("ropenscilabs/miner")
install_github("ropenscilabs/craft")
library(miner)
mc_connect("52.170.156.222")
chatPost("Connection Test")
```

- Connections expire after about 10 mins, so reconnect as needed

Modifying the World

```
library(craft)
v <- getPlayerPos()
write_text("Welcome to" , v+c(0,12,0),
 font="8x12", dir="west")
write_text("RLadies" , v+c(0,1,0),
 font="8x12", dir="west")
```

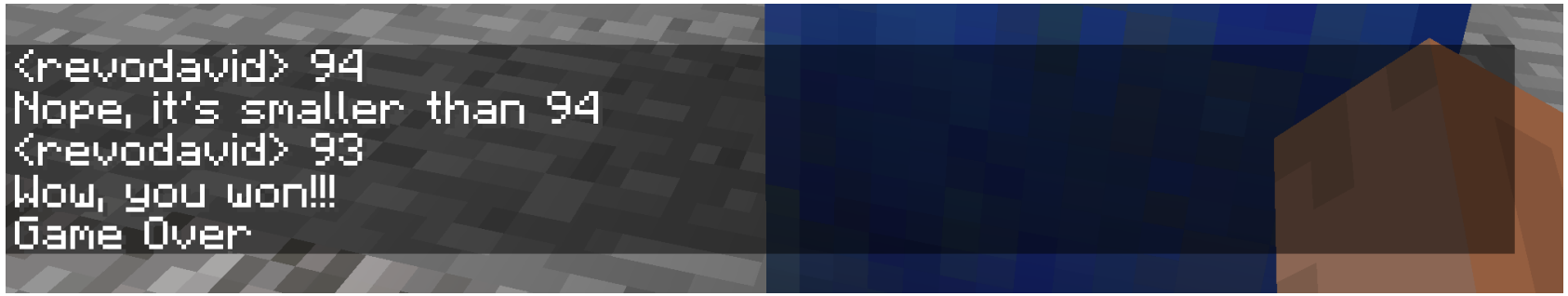

mc_connect()
getPlayerPosition()
setBlocks()
getBlockHits()


library(craft)
myid <- tail(getPlayerIds(),1)
elsafy(myid)

Interacting with players via chat

- Send a message: `chatPost()`

- Players can also chat (press T), which allows us to interact
- Get recent messages: `getchatPosts()`



Microsoft


chatPost('Hi, I thought a number between 1 and 100.')
msg <- getChatPosts()
Minecraft 1.11.2
Nope, it's greater than 70
<rstatsguy> 71
Wow, you won!!!
Hi, I thought a number between 1 and 100. Can you guess it?
Type numbers in the chat and whoever finds out the number
first, will receive a foobar!
<rstatsguy> 50
Nope, it's greater than 50
<rstatsguy> 75
Nope, it's smaller than 75
<rstatsguy> 60
Nope, it's smaller than 60
<rstatsguy> 55
Nope, it's greater than 55
<rstatsguy> 57
Nope, it's greater than 57
<rstatsguy> 59
Nope, it's smaller than 59
<rstatsguy> 58
Wow, you won!!!

Maze Solving

Microsoft

Resources

miner package:
github.com/ropenscilabs/miner

"R Programming with Minecraft":
ropenscilabs.github.io/miner_book

craft package (examples):
github.com/ropenscilabs/craft

The miner team:

- Brooke Anderson
- Karl Broman
- Gergely Daróczi
- Mario Inchiosa
- David Smith
- Ali Zaidi

Join my server: 13.66.184.220

Microsoft