Contents

[**Requirements** 2](#_Toc208706150)

[**Installation of Java and adding to path** 3](#_Toc208706151)

[**First-Time Instructions** 5](#_Toc208706152)

[**Usage instructions** 9](#_Toc208706153)

[**GUI** 9](#_Toc208706154)

[**Alternative way to create Turing machine instructions- editing the text files** 10](#_Toc208706155)

[**Lookup Table** 11](#_Toc208706156)

[**Minecraft plugin** 12](#_Toc208706157)

## **Requirements**

Microsoft Windows as the operating system

A valid Minecraft Java Edition license and account.

A computer capable of running Minecraft Java Edition.

Internet connection.

Java installed and added to path.

Basic understanding of Turing machines.

## **Installation of Java and adding to path**

If you don’t have Java installed and added to path in environment variables (meaning Windows being able to run certain new console commands that comes with Java), the software isn’t able to run. If you already have Java installed and added to path, this can be skipped.

1.

Download Java Development Kit installer for Windows, for example from

https://www.oracle.com/java/technologies/downloads/#jdk24-windows

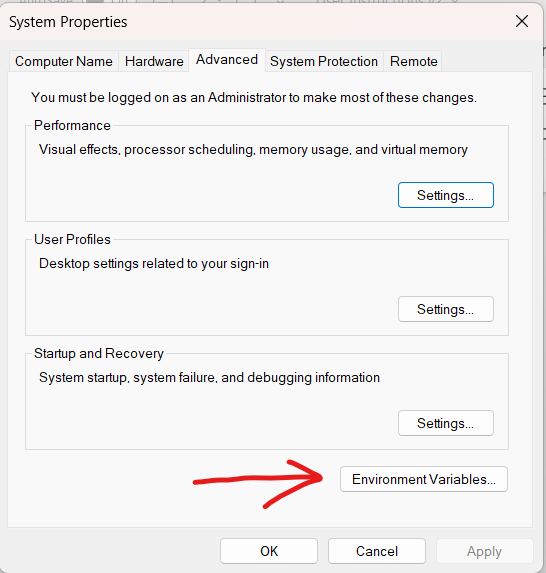
Run the installer. This will launch an installation program. Follow the instructions and either write down or keep memorized the file location you install to.

2.

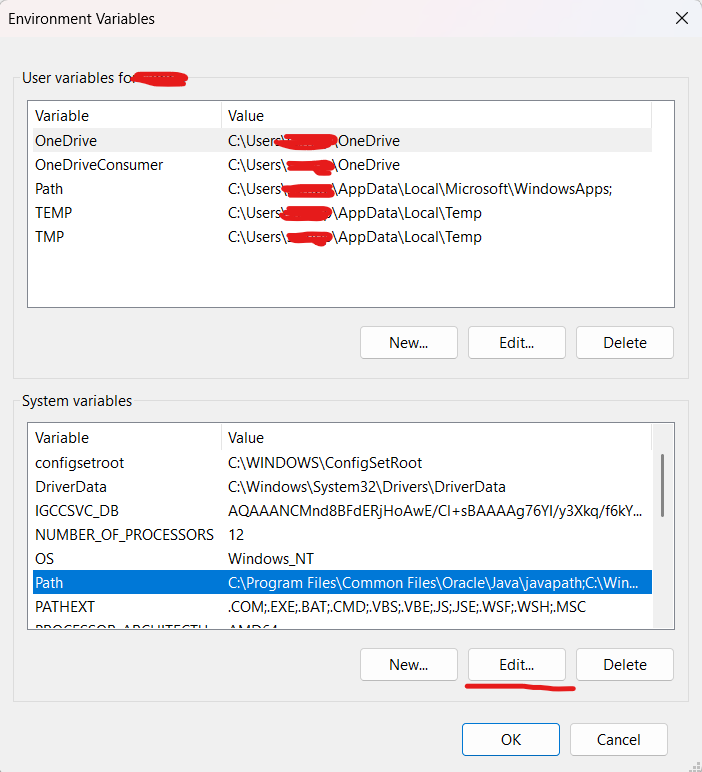
Once it has finished installing, go to the folder Java was installed in, and navigate to the “bin” folder. Then, type into Window’s Search field **environment variables** and select “Edit the system environment variables”.

3.

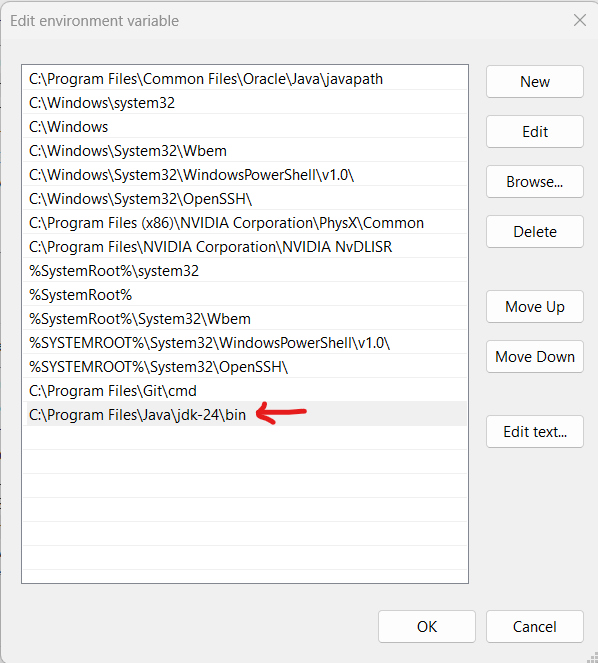
In this new “System Properties” window, press “Environment Variables” button.



Then, in the “System variables” window, double click Path, or select it and then press “Edit...” button.



Then, go back to the “bin” folder, and copy the file path of the folder. Return to “Edit environment variable” window, press “New” button and paste the file path.



You may now close all the windows.

## **First-Time Instructions**

1.

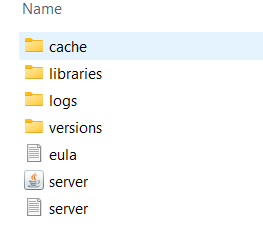
Download Bukkit Server, for example from

https://serverjar.org/download-version/bukkit.

The software of this project was tested on Bukkit server versions for Minecraft 1.21.5 to 1.21.8, and they are currently confirmed to work. Others are untested but expected to work.

2.

Move the "server.jar" file that was just downloaded into the file location of your choosing, and execute the file. It will then unpack necessary files, which might take several minutes. Unpacking is finished when four new folders and two new text files appear in the same folder as “server.jar”:



3.

Open "eula.txt" and edit the value of "eula" to **true**. Save and close the file.

4.

Open "server.properties" file and replace the value of "enable-command-block" with **true**. Also, replace the value of “level-type” with **minecraft\:flat**. These are the only obligatory changes, but it is recommended to edit other values too, like setting "spawn-monsters" to **false, “**difficulty” to **peaceful** and "gamemode" to **creative**. Save and close the file.

5.

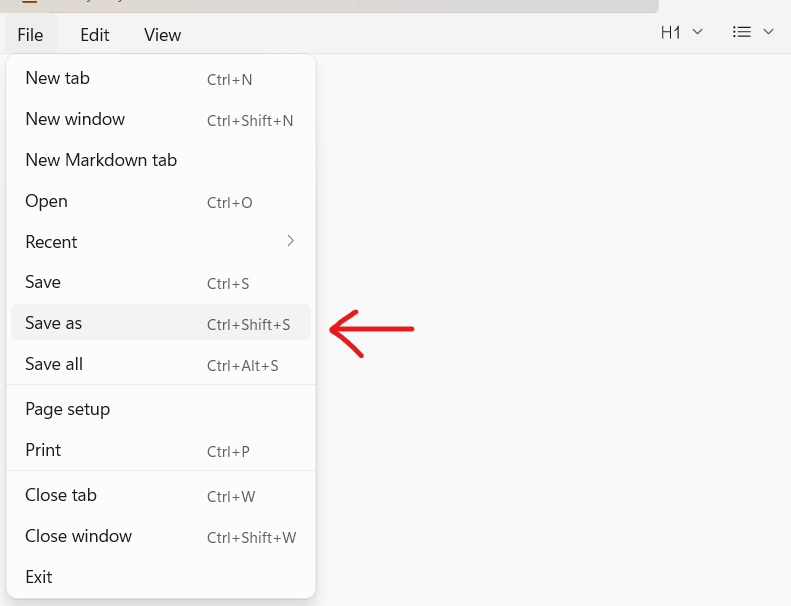
Next, we need to create a “run.bat” file, but creating .bat files is not a default Windows action. To create it, we need to use a detour:

Create a temporary text file in the same folder, any name. Open it. Into it, write

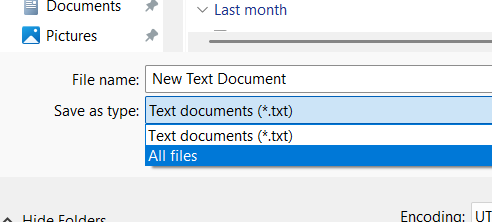
**java -jar server.jar**

**PAUSE**

Then, go to the drop menu File -> Save as.



There, change the "Save as type" to All files. This is necessary, because we want a file of type .bat, but otherwise it would save the file as .txt.



Finally, name the file **run.bat**. You may delete the temporary text file if you want, it serves no further use.

6.

Launch the server by running "run.bat".

7.

Once the server has finished the start-up process, type into the server's console **op Your\_Minecraft\_Username**. For example, if your Minecraft username is VillageTerror123, you would type **op VillageTerror123**. This grants the user operator rights in the server. Close the server by typing in the server's console **stop**.

8.

Open the file “spigot.yml”, and change the value of “timeout-time” to some very large number, like **1000000000000000000**. This number ensures the server is unlikely to engage the safety crash when running certain necessary commands that take a lot of time to execute, such as force loading chunks. Save and close the file.

9.

Download "Lookup Table.txt", “Settings.txt”, "TM\_GUI.jar" and "Turing\_machine.jar" files from

https://github.com/SamiKautto/TM\_Minecraft

and move them into the folder named "plugins".

Optionally, you can also download some of the example Turing machine instructions from the same site, such as “Rainbow.txt” and “Zigzag.txt”, and place them in the same folder.

10.

Launch both the Minecraft client in the same version as the server, and the server again by running "run.bat". These can be done in parallel.

11.

In the Minecraft client, go to Multiplayer -> Direct Connection, and enter the IPv4 address of the host. If you don't know the IPv4 address, here's how to find it: In the host machine or other computers connected to the same network, type into Windows' Command Prompt **ipconfig**, and look for the IPv4 address.

12.

In the first use, type **/initialize** into the Minecraft console. This will clear the operation area in the game, read the settings, force load a large number of chunks that make up the game world, as well as teleport the player into the center of the work area. This might take several minutes.

# **Usage instructions**

## **GUI**

1.

Launch the GUI by executing "TM\_GUI.jar".

2.

Enter the name of the Turing machine, the alphabet you want to be used, and the number of states, and then press the "Update instructions" button. Notice that due to how this code works, “,” is obviously not a valid alphabet character.

3.

Enter the values for multiple choice elements in the blank field and in the cells. once you’re satisfied with the selection, press the “Generate the text file” button. Pressing it will generate a text file - containing the machine instructions - into the “plugins” folder. The Minecraft plugin will read these files in order to generate a Turing machine within game.

4.

Either create other Turing machines, or close the program.

## **Alternative way to create Turing machine instructions- editing the text files**

1.

Go to the plugins-folder, and either create a new text file or open one of the text files other than "Lookup Table.txt" or “Settings.txt”.

2.

First, either write or ensure there is a line **blank:**, followed by the symbol you want to be the blank symbol (or the symbol that is used as the default symbol in the tape – extending the tape is done by placing this into the air block’s place). For example, if you want to use 0 as the blank symbol, write **blank:0**.

3.

Each instruction is a comma-separated series of values, with the following format:

- First is a positive whole number, representing the current state. Note: state 1 is the default starting state, if it is missing the machine will not start without you changing the state with console commands.

- Second is the tape symbol that is read from the current cell.

- Third is a positive whole number, representing the next state.

- Fourth is the tape symbol that is to be written in the current cell.

- Fifth is the direction the tape is moved, with 0 representing moving it right or 1 moving it left.

Each row is treated as a separate instruction (what is done for each combination of current state and input symbol).

It is important the format is followed exactly, otherwise the Minecraft plugin isn’t able to generate a valid Turing machine.

Once finished, save the file and close it.

## **Lookup Table**

For the Minecraft plugin to be able to convert the alphabet into blocks, it is important to maintain the lookup table.

1.

Go to the plugins-folder, and open "Lookup Table.txt".

2.

Each entry in the table is a pair of text, separated by a comma in the following format:

-First is the alphabet symbol

-Second is the Minecraft identifier for the block, of format **minecraft:canonical\_name**.

Only use valid identifiers, otherwise the Minecraft client doesn't recognize the block. If you don't know the identifier, either find it out using the autofill feature of the Minecraft console, or consult the Minecraft wiki: https://minecraft.wiki/w/Block.

The software and the machine is also able to recognize and place blocks with specific block states. If you want to use block states, they can be accessed by using the square brackets, [], after the identifier without a space in between. The states are pairs of name and value separated by the equality sign, =. For example, if you want a redstone lamp that is on, the identifier would be **minecraft:redstone\_lamp[lit=true]**. Additional block states are separated with a simple comma: [state1=value1, state2=value2, etc.]. Make sure to only use valid names and values, otherwise the Minecraft plugin isn’t able to generate a valid machine.

It is important each used tape symbol is presented in the table, otherwise the Minecraft plugin isn't able to generate a valid Turing machine.

Once finished, save the file and close it.

## **Minecraft plugin**

1.

Launch Minecraft in the same version as the server, and the server by running "run.bat". These can be run in parallel.

2.

Once both are finished, in the Minecraft client, go to Multiplayer -> Direct Connection, and enter the IPv4 address of the host. If you don't know the IPv4 address, here's how to find it: In the host machine or other computers connected to the same network, type into Windows' Command Prompt **ipconfig**, and look for the IPv4 address.

3.

Typing **/initialize** will clear the working area around the origin, or (0,0) coordinates in x- and z-directions, as well as force load chunks. This will also read the settings, so if you want to change the “move limit” value in “Settings.txt”, you need to run this command for changes to take effect.

Typing **/generateturing Name\_Of\_Instructions** will generate the Turing Machine in Minecraft. For example, if the name you gave the Turing machine is “Rainbow”, enter **/generateturing Rainbow**.

Typing **/clearturing Name\_Of\_Instructions** will clear the Turing Machine. For example, if the name you gave the Turing machine is “Rainbow”, enter **/clearturing Rainbow**. It is necessary to enter the name as well, so that the plugin can correctly clear the Minecraft scoreboards as well. The scoreboards are used for things like keeping track of the state the machine is in. Please clear the previous Turing machine before generating another one, otherwise the machines are generated overlapping and are unlikely to function as wished.

Pressing the "Start" button will start or continue the Turing machine execution.

Pressing the "Pause" button will pause the execution.

Pressing the "Reset" button will reset the tape's values into the air blocks (equivalent of empty tape cell) and set the state into the initial state, or 1.

4.

Once you've finished using the program, disconnect in the Minecraft client and type **stop** into the server console.