

# GameSHell

Alessandro Morabito

Epicode

13/10/2023

# Introduction

The goal is to beat the first 10 levels of the GameShell, which can be downloaded at <https://github.com/phyver/GameShell/releases/download/latest/gameshell.sh>.

In order to win, you must understand how some basic shell commands work.

# Level 01

In order to beat level 1, you are requested to reach the "top of the tower". In this context, this translates into changing your current working directory to `~/Castle/Main_tower/` first, and then cd into the next directory until there is none. You can list the current directory, along with hidden files, with `ls -la`.



The screenshot shows a terminal window titled "Terminal - Mission 01" running on a Kali Linux desktop. The user has typed the following commands:

```
[mission 1]$ ls -la
total 20
drwxr-x-- 7 kali kali 4096 Oct 13 03:38 .
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 ..
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 Building
drwxr-x-- 2 kali kali 4096 Oct 13 03:38 Castle
drwxr-x-- 1 kali kali 4096 Oct 13 03:38 Forsil
drwxr-x-- 1 kali kali 4096 Oct 13 03:38 Grotto
drwxr-x-- 1 kali kali 4096 Oct 13 03:38 lessmst
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 Mountain
drwxr-x-- 2 kali kali 4096 Oct 13 03:38 Stalls
[mission 1]$ cd Castle
[mission 1]$ ls -la
total 20
drwxr-x-- 7 kali kali 4096 Oct 13 03:38 .
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 ..
drwxr-x-- 2 kali kali 4096 Oct 13 03:38 Cellar
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 Corridor
drwxr-x-- 4 kali kali 4096 Oct 13 03:38 Main_Building
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 Main_tower
drwxr-x-- 2 kali kali 4096 Oct 13 03:38 Observatory
[mission 1]$ cd Main_tower/
[mission 1]$ ls -la
total 12
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 .
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 ..
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 First_Floor
[mission 1]$ cd First_Floor/
[mission 1]$ ls -la
total 8
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 .
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 ..
drwxr-x-- 3 kali kali 4096 Oct 13 03:38 Secored_Floor
[mission 1]$ cd Top_of_the_tower/
[mission 1]$ ls -la
total 0
[mission 1]$
```

The terminal window is set against a background featuring the Kali Linux logo.

# Level 02

You now want to reach the cellar. In order to do so, you use the same command you used in the previous level. I thought it'd be good being able to visualize the path to your current directory without having to run `pwd`, so I modified the PS1 variable by adding a `\w` right before the dollar sign.



The screenshot shows a terminal window titled "Terminal 2023-09-10T10:23:45Z - 100%". The terminal displays the following session:

```
[mission 2] $ echo $PS1
mission 2] $ \w
mission 2] $ PS1='[mission $1@ghn ~] \w $ '
mission 2] >/Castle/Maze/tower/First_floor/Second_floor/Top_of_the_tower $ ls -v /Castle
Castle/Maze/tower/First_floor/Second_floor/Top_of_the_tower/mission
mission 2] >/Castle/Maze_tower/First_floor/Second_floor/Top_of_the_tower $ cd ~/Castle/Cellar/
mission 2] >/Castle/Cellar $ gnh check
Congratulations, mission 2 has been successfully completed!
```

Below the terminal, a large watermark for "KALI LINUX" is visible, along with the tagline "the quicker you become, the more you are able to hear".

## Level 03

To complete this level you can use up to 20 commands. You are required to reach the throne room. To avoid having any kind of problem given the restriction, considering the low amount of files/directories inside the home, you can autocomplete cd with <TAB>. Alternatively, you can list all directories recursively with ls -laR or tree -fa with the first command, and then cd into the target directory once found. The main difference between the two approaches is that autocomplete ignores hidden files. Hence, in this scenario, the two methods are equivalent.



# Level 04

In order to create the directory `~/Forest/Hut/Chest`, you are required to first create the non-existing parent directory `~/Forest/Hut/`. You can either use the `mkdir` command twice, or the command `mkdir -p` once.



The screenshot shows a terminal window titled "Terminal 2023-09-10T10:42:44Z" with the command history:

```
/Castle/Nox/building/Throne_room  
[mission 4]$ mkdir -p ~/Forest/Hut/Chest  
/Castle/Nox/building/Throne_room  
[mission 4]$ gsh check  
Congratulations, mission 4 has been successfully completed!  
  
| Use the command  
| $ gsh help  
| to get the list of "gsh" commands.  
  
/Castle/Nox/building/Throne_room  
[mission 5]$
```

A large watermark for "KALI LINUX" with the tagline "the quieter you become, the more you are able to hear" is overlaid on the right side of the terminal window.

# Level 05

I chose to use the command `rm *spider*` to beat the fifth level. I used the *wildcard* `*` around the word `spider` because I wanted to delete every file containing such word in its name. Given the low amount of files I had to analyze, I could have reached the same result by indicating the files I wanted to remove one-by-one, with the command `rm spider_1 spider_2 spider_3`.

The screenshot shows a terminal window on Kali Linux with the following session:

```
[mission 5] $ cd ~/Castleville/Cellar/  
[mission 5] $ ls -l  
total 25  
-rwxr-x--- 1 kali kali 123 Oct 13 00:00 barrel_of_spiders  
-rw-r--r-- 1 kali kali 196 Oct 13 00:00 hat_1  
-rw-r--r-- 1 kali kali 196 Oct 13 00:00 hat_2  
-rw-r--r-- 1 kali kali 196 Oct 13 00:00 spider_1  
-rw-r--r-- 1 kali kali 123 Oct 13 00:00 spider_2  
-rw-r--r-- 1 kali kali 223 Oct 13 00:00 spider_3  
[mission 5] $ ls -l | grep set | grep spider  
[mission 5] $ rm spiders  
[mission 5] $ ls -l  
total 17  
-rwxr-x--- 1 kali kali 123 Oct 13 00:40 barrel_of_spiders  
-rw-r--r-- 1 kali kali 196 Oct 13 00:00 hat_1  
-rw-r--r-- 1 kali kali 196 Oct 13 00:00 hat_2  
[mission 5] $ rm check  
[mission 5] $ rm check  
Congratulations, mission 5 has been successfully completed!
```

The terminal then displays a success message and a list of achievements:

```
| _____ |  
| Use the command |  
| I got help |  
| I got the list of "git" commands. |  
| _____ |
```

At the bottom, it says "Castleville" and "KALI LINUX".

# Level 06

To complete the sixth mission you use the `mv` command, indicating the files you want to move first, and leaving the destination path last. Since I wanted to copy more than one file, I decided to use the *wildcard*. You could also simply list the file you wanted to copy one-by-one.



The screenshot shows a terminal window titled "Terminal 2022-09-10T10:45:00Z" with the session number "Session 6". The terminal displays the following commands and output:

```
</Desktop> $ cd ~/Downloads
</Desktop> $ ls -la
total 20
drwxr-xr-x 5 kali kali 4096 Oct 13 06:49 .
drwxr-xr-x 1 kali kali 4096 Oct 13 06:49 ..
-rw-r--r-- 1 kali kali 47 Oct 13 06:49 cain_1
-rw-r--r-- 1 kali kali 47 Oct 13 06:49 cain_2
-rw-r--r-- 1 kali kali 47 Oct 13 06:49 cain_3
-rw-r--r-- 2 kali kali 4860 Oct 13 06:49 cain_pwned
-rw-r--r-- 2 kali kali 4860 Oct 13 06:49 Next
-rw-r--r-- 2 kali kali 4860 Oct 13 06:49 Next_bash

</Desktop> $ mv *cain* ~/forest/NetChest/
</Desktop> $ ls
</Desktop> $ ./get_check
(Congratulations, mission 6 has been successfully completed)

| Use the command | | the easier you become, the more you are able to hear |
| ./get_level | | |
| to get the list of "gdb" commands. | | |

</Desktop> [mission 7] $
```

Levels 07, 08, 09

In the seventh level I had to make sure the *dotfiles* were included in the command. Since I used the *wildcard*, the problem was already solved. Nevertheless, I prepended a `.` to the name of the file I wanted to move. This would mean that *only* the *dotfiles* containing spider in the name be moved to the destination.

To solve the eighth level you do the same, except you change the command to `rm`. The same goes for the ninth mission. I faced no problems here since I already learnt the *wildcard* to select multiple files at once.

# Level 10

This is the last level required as an exercise. I used the ? wildcard after listing the directory containing the files I was required to copy, and noticing that their name all shared the pattern standard\_<single\_digit\_number>. I used the command cp standard\_? ~/Forest/Hut/Chest to copy such files to the desired path. I would have been required more effort to workaround the problem if there were files I wasn't required to copy in the similar form standard\_<character>, since ? accepts any character.

The screenshot shows a terminal window titled "Kali Linux 2021.1 - terminal under Xfce" running on Oracle VM VirtualBox. The terminal displays the following session:

```
[Castleville] [Session 1@] $ cd ~/Castle/Keveat_hall/
[Castleville] [Session 1@] $ ls -la
total 40
drwxr-xr-x 2 kali kali 4096 Oct 13 06:17 .
drwxr-xr-x 2 kali kali 4096 Oct 13 06:17 ..
-rw-r--r-- 1 kali kali 427 Oct 13 06:17 23h_decorative_shield
-rw-r--r-- 1 kali kali 229 Oct 13 06:17 28017_stag_head
-rw-r--r-- 1 kali kali 229 Oct 13 06:17 28017_stag_head
-rw-r--r-- 1 kali kali 47 Oct 13 06:17 standard_1
-rw-r--r-- 1 kali kali 47 Oct 13 06:17 standard_2
-rw-r--r-- 1 kali kali 47 Oct 13 06:17 standard_3
-rw-r--r-- 1 kali kali 47 Oct 13 06:17 standard_4
[Castleville] [Session 1@] $ cp standard_? ~/Forest/Hut/Chest/
[Castleville] [Session 1@] $ gun check
Congratulations, mission 10 has been successfully completed!
```

The terminal then shows a message from the game: "I like the command  
I like the command  
I like the command  
to get the list of "goh" commands.  
The harder you become, the more you are able to hear!"

At the bottom, it says "[Castleville] [Session 1@] \$".

# Level 42

I decided to continue the game with the goal to clear it, and successfully reached the last level.

The file `~/Castle/Main_building/Library/Merlin_s_office/Drawer/secret_message` had been encrypted using a rotational cipher. I used pipe | to redirect the output of `cat secret_message` to standard input, which was then read by `tr`. This command allows to easily apply a rotational cipher.

The first word in the output of `cat secret_message` was `urer`. The first word having the same second and fourth letter would be `here`, hence I applied the rotation from the alphabet `u->z` `a->t` to the alphabet `h->z` `a->g`.



# Conclusion

In order to complete the game I believe I became more proficient with the command line. I also learned about the command `awk` and the *backtick* ```, although I didn't use any of them above and they weren't really required to complete the game.