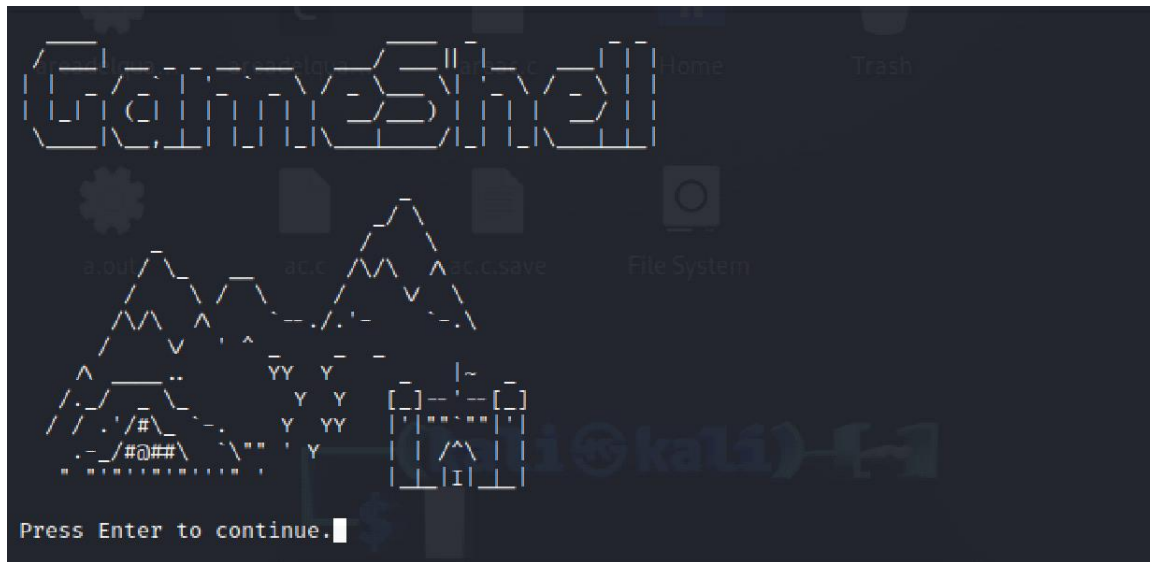


# Gameshell



## 1. installazione di gameshell

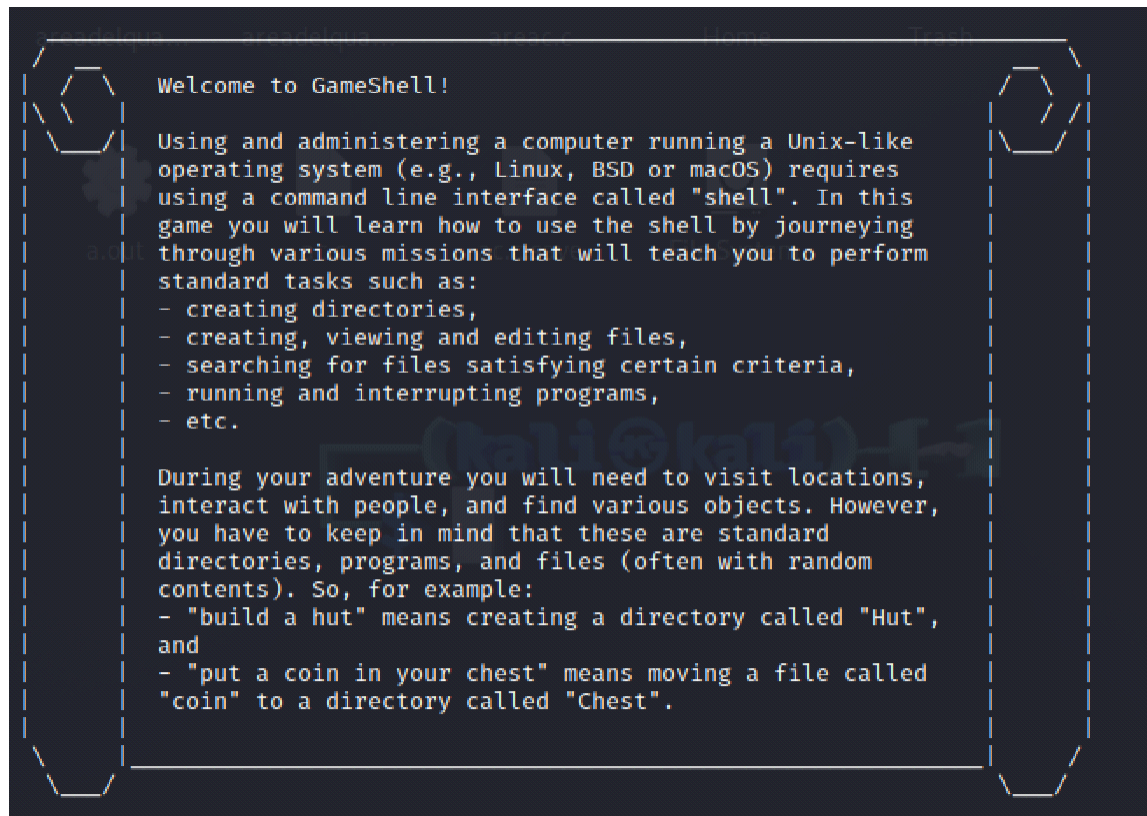
Per installare il gioco "Gameshell", eseguire in ordine i seguenti comandi :

- sudo apt update
- sudo apt install gettext man-db procps psmisc nano tree bsdmainutils x11-apps wget
- wget <https://github.com/phyver/GameShell/releases/download/latest/gameshell.sh>

### 1.1 Lanciare il gioco

Una volta eseguiti tutti i comandi sopraindicati , si può lanciare il gioco con il comando "bash gameshell.sh"

Premere "enter" per poter iniziare la partita!



## 1.3 Strutta del gioco

Il gioco è composto da più livelli, l'avanzamento di livello richiede il completamento di determinate missioni da portare a termine.

Per ogni livello si può controllare la missione e il suo stato con i comandi:

**#gsh goal** mostra la missione per il livello corrente

**#gsh check** controlla lo stato della missione attuale

```
|  
+-----+  
| Run the command  
|   $ gsh goal  
| to discover your first mission.  
|  
| You can check the mission has been completed with  
|   $ gsh check  
|  
| The command  
|   $ gsh help  
| displays the list of available (gsh) commands.  
+-----+  
|  
[mission 1] $
```

## Missione1

```

[mission 1] $ cd
[mission 1] $ pwd
/home/kali/gameshell/World
[mission 1] $ ls
Castle Forest Garden Mountain Stall
[mission 1] $ cd Castle
[mission 1] $ ls
Cellar Great_hall Main_building Main_tower Observatory
[mission 1] $ cd Main_tower
[mission 1] $ ls
First_floor
[mission 1] $ cd First_floor
[mission 1] $ ls
Second_floor
[mission 1] $ cd Second_floot
bash: cd: Second_floot: No such file or directory
[mission 1] $ cd Second_floor
[mission 1] $ ls
Top_of_the_tower
[mission 1] $ cd Top_of_the_tower
[mission 1] $ ls
[mission 1] $ gsh check

```

Congratulations, mission 1 has been successfully completed!

```

|_|
--+-----+--
| Use the command |
| $ gsh help      |
| to get the list of "gsh" commands. |
--+-----+--
|_|

```

Glossario dei comandi:

**CD**=(change directory) questo comando ti permette di spostarti tra le directory

**LS**=permette all'utente di elencare file o directory

**PWD**=(print working directory) permette all'utente di scrivere il percorso completo della tua directory

## Missione 4

```

Castle  Forest  Garden  Mountain  Stall
areadelqua... areadelqua... areac.c... Home  T
~
[mission 4] $ cd Forest

~/Forest
[mission 4] $ ls

~/Forest
[mission 4] $ mkdir "Hut"

~/Forest
[mission 4] $ ls
Hut

~/Forest
[mission 4] $ cd Hut

~/Forest/Hut
[mission 4] $ mkdir Chest

~/Forest/Hut
[mission 4] $ ls
Chest

~/Forest/Hut
[mission 4] $ gsh check

Congratulations, mission 4 has been successfully completed!

|                                     |
--+-----+-----+
| Use the command                    |
| $ gsh help                        |
| to get the list of "gsh" commands. |
--+-----+-----+
|                                     |

```

Glossario:

**mkdir**=crea una o più directory

## Missione 5

```
Mission goal
=====

Go back to the cellar and get rid of all the spiders. Leave the bats alone: they
appear on the castle's coat of arms and are said to confer luck.

Useful commands
=====

rm FILE1 FILE2 ... FILEn
Delete the files (permanently).
Remark: "rm" is an abbreviation for "remove".

~/Forest/Hut
[mission 5] $ pw
pw: command not found

~/Forest/Hut
[mission 5] $ pwd
/home/kali/gameshell.1/World/Forest/Hut

~/Forest/Hut
[mission 5] $ cd
~
[mission 5] $ pwd
/home/kali/gameshell.1/World

~
[mission 5] $ cd Castle/Cellar

~/Castle/Cellar
[mission 5] $ ls
barrel_of_apples  bat_1  bat_2  spider_1  spider_2  spider_3

~/Castle/Cellar
[mission 5] $ rm spider_1 spider_2 spider_3

~/Castle/Cellar
[mission 5] $ ls
barrel_of_apples  bat_1  bat_2

~/Castle/Cellar
[mission 5] $
```

Glossario:

**RM**=elimina tutti i file all'interno della cartella, poi la cancella.

## Missione 11

```
Mission goal
=====

The tapestries in the castle's great hall are also particularly
beautiful. Put a copy of each in your chest.

Useful commands
=====

cp FILE1 FILE2 ... FILEn DIRNAME
Copy the files to the directory.
Remark: "cp" is an abbreviation of "copy".

Shell patterns
=====

*
The "*" character stands in for any sequence of characters
(including an empty sequence).

?
The "?" character stands in for any single character.
```

glossario:

**CP**=copiare file da una posizione a un'altra

# Missione12

```
Mission goal
-----

Nostradamus predicted a spectacular star conjunction on the 07-14-1963.
But what will the day of the week be on that date?

When you have it, run the command ``gsh check``.

Useful commands
-----

cal
Print a calendar for the current month.
cal YEAR
Print a calendar for the given year.
```

Glossario:

**CAL**=permette di aprire il calendario del mese corrente

**CAL YEAR**=permette di aprire il calendario annuo

# Missione15

```
Mission goal
-----

Create a file named "journal.txt" in your chest and write a short message in it.
You can use this file to record your notes and solutions for the upcoming missions.

Details
-----

``nano`` is a command-line text editor. You can use it whenever you need to edit a file from the shell.

Useful commands
-----

nano FILE
Edit the file from the shell.
(If the file does not exist, it will be created.)

Keybindings are listed at the bottom of the screen (the ``^`` symbol means "Control"). The most important
ones are:
  Control-x    quit
  Control-o    save
  Control-w    search for a string

Remark: do not use Control-s or Control-z!
```

glossario:

**NANO**=permette di creare un file ed aprirlo

# MISSIONE25

```
Mission goal

The old hermit notices your interest for potion recipes, and sees promise in your
ability to lookup lists of ingredients. He challenges you to find the steps for
the elixir of Youth.

In order to validate the mission, you need to be in the cave with Servillus
**and** your last command prior to `gsh check` must show the steps for the
recipe and nothing else.

Note: you shouldn't alter the content of the book of potions.

Useful commands

cat FILE1 FILE2 ... FILEn
Display the contents of the files in order.

tail
Print the last 10 lines sent on the standard input.

tail -n K
Print the last K lines sent on the standard input.

COMMAND1 | COMMAND2
Run the two commands, feeding the "standard output" of the former into the
"standard input" of the latter.
Remark: by analogy with plumbing "|" is called "pipe".

Explanations
inter you become, the more you are able

Many of Unix commands process text: they receive text as input and produce text
as output.

It is common for those commands to write their output to their "standard output",
which means that (by default) the output is written into the terminal.

Most of those commands can receive input either through files (given as arguments)
or from their "standard input". For example:
- `head FILE` reads its input from the file,
- `head` reads its input on the standard input.

By default, data from the standard input is read from the keyboard, but a pipe
can change that.
```

## GLOSSARIO

**CAT**=egge i file che gli sono specificati come parametri