



# Gameshell Linux

Ernesto Robles

**Gameshell è un dispositivo portatile open-source progettato per gli appassionati di giochi e gli sviluppatori, ha un design modulare che consente agli utenti di personalizzarlo e modificarlo per adattarlo alle proprie esigenze di gioco e sviluppo inoltre è basato su un sistema operativo Linux**

**Il gioco seguente ha un ambiente fantasy che ha come scopo l'apprendimento dei comandi Shell Linux mediante il completamento di livelli che ne richiedono l'uso per il completamento del medesimo.**

**Il livello 1 richiede l'uso dei comandi cd e ls.**  
cd ci permette di cambiare directory in questo caso lo scopo è raggiungere il tetto della torre e grazie ad ls che ci permette di elencare i file e le directory presenti in una specifica directory del sistema possiamo seguire il percorso adatto

```
[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_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!
```

**Il livello 2 richiede l'uso del comando ls, cd .. e cd.**  
cd .. ci permette di tornare indietro alla directory padre della directory in qui ci si trova in questo caso /World

```
[mission 2] $ pwd
/home/kali/gameshell/World
[mission 2] $ ls
Castle  Forest  Garden  Mountain  Stall
[mission 2] $ cd Stall
[mission 2] $ ls
[mission 2] $ pwd
/home/kali/gameshell/World/Stall
[mission 2] $ cd ..
[mission 2] $ cd Castle
[mission 2] $ ls
Cellar  Great_hall  Main_building  Main_tower  Observatory
[mission 2] $ cd Cellar
[mission 2] $ gsh check

Congratulations, mission 2 has been successfully completed!
```

**Il livello 3 richiede l'uso del comando cd.**  
Con si cd torna alla directory di partenza per poi spostarci successivamente in un'altra directory

```
[mission 3] $ cd
[mission 3] $ cd /home/kali/gameshell/World/Castle/Main_building/Throne_room
[mission 3] $ gsh check

Congratulations, mission 3 has been successfully completed!
```

**Il livello 4 richiede l'uso del comando mkdir.**

**mkdir ci permette di creare una nuova directory dentro quella corrente in questo caso con l'uso di cd le creiamo all'interno di quelle d'interesse per il livello**

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

~/Forest
[mission 4] $ cd Hut

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

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

Congratulations, mission 4 has been successfully completed!
```

**Il livello 5 richiede l'uso del comando rm.**  
**rm ci permette di eliminare i file o directory una volta rimosso non è possibile recuperarlo direttamente, in questo livello eliminiamo i file all'interno della directory /Cellar**

```
~/Forest/Hut
[mission 5] $ cd /home/kali/gameshell/World/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] $ gsh check

Congratulations, mission 5 has been successfully completed!
```

**Il livello 6 richiede l'uso del comando mv.**  
**mv ci permette di spostare o rinominare file e directory, in questo livello spostiamo i file selezionati nella directory finale /Chest specificando tutto il path**

```
~/Garden
[mission 6] $ ls
Flower_garden  Maze  Shed  coin_1  coin_2  coin_3

~/Garden
[mission 6] $ mv coin1 coin2 coin3 /home/kali/gameshell/World/Forest/Hut/Chest
mv: cannot stat 'coin1': No such file or directory
mv: cannot stat 'coin2': No such file or directory
mv: cannot stat 'coin3': No such file or directory

~/Garden
[mission 6] $ mv coin_1 coin_2 coin_3 /home/kali/gameshell/World/Forest/Hut/Chest

~/Garden
[mission 6] $ gsh check

Congratulations, mission 6 has been successfully completed!
```

Il livello 7 richiede l'uso del comando ls -A, Tab ed mv.

ls -A ci permette di visualizzare i file nascosti e Tab ci aiuta a filtrarli in base alla stringa che inseriamo per una ricerca più veloce, successivamente li spostiamo nella directory di destinazione /Chest

```
~/Garden
[mission 7] $ mv .15247_coin_2 .5
.51933_coin_3 .52376_coin_1

~/Garden
[mission 7] $ mv .15247_coin_2 .51933_coin_3 .52376_coin_1 ~/Forest/Hut/Chest

~/Garden
[mission 7] $ gsh check

Congratulations, mission 7 has been successfully completed!
```

Il livello 8 richiede l'uso del comando ls, rm e della wildcard \*.

In questo caso con la wildcard \* si eliminano tutti quei file che hanno nel loro nome la stringa spider

```
~/Castle/Cellar
[mission 8] $ ls
10158_spider_23  14329_spider_26  17893_spider_41  21838_sp
1181_spider_28   14594_spider_37  18651_spider_29  22084_sp
12506_spider_12  14730_spider_25  18923_spider_3   22262_sp
13156_spider_7   17346_bat_2      19820_spider_21  24091_sp
13352_spider_49  17361_spider_11  20806_spider_6   24254_sp
1429_spider_40   17699_spider_20  2168_spider_17   24577_sp

~/Castle/Cellar
[mission 8] $ rm *spider*

~/Castle/Cellar
[mission 8] $ gsh check

Congratulations, mission 8 has been successfully completed!
```

**Il livello 9 richiede l'uso del comando ls -A, rm e \*.**

**Il comando è simile a quello del livello 8 con la differenza che l'aggiunta del "." permette di eliminare tutti i file che sono nascosti e hanno nel nome la stringa spider**

```
~/Castle/Cellar
[mission 9] $ ls -A
.10005_spider_7      .13854_spider_24    .18906_bat_3        .20943
.10895_spider_6      .1396_spider_15     .19689_spider_28     .21125
.11099_spider_47     .14351_spider_42    .19842_bat_5         .21848
.11639_spider_49     .14461_spider_46    .19971_spider_30     .22839
.1212_spider_20      .14498_spider_45    .20026_bat_4         .23555
.13370_bat_2         .15300_spider_5     .20316_spider_12     .25432
.13469_spider_50     .1636_spider_10     .20848_spider_17     .26346

~/Castle/Cellar
[mission 9] $ rm .*spider*

~/Castle/Cellar
[mission 9] $ gsh check

Congratulations, mission 9 has been successfully completed!
```

**Il livello 10 richiede l'uso del comando cp. cp ci permette di copiare file in un'altra directory.**

**In questo caso tutti i file che hanno nel nome la stringa standard vengono copiati nella directory finale /Chest con lo stesso nome che avevano nella directory /Great\_hall**

```
~/Castle/Great_hall
[mission 10] $ cp standard* ~/Forest/Hut/Chest

~/Castle/Great_hall
[mission 10] $ gsh check

Congratulations, mission 10 has been successfully completed!
```