**《Linux实训》实验报告**

**实验2**

**1.作业要求概述**

本课程会在每一节实验结束部分介绍一个有趣的 Linux 命令，这一节介绍一个可以输出图形字符的命令banner，toilet，figlet

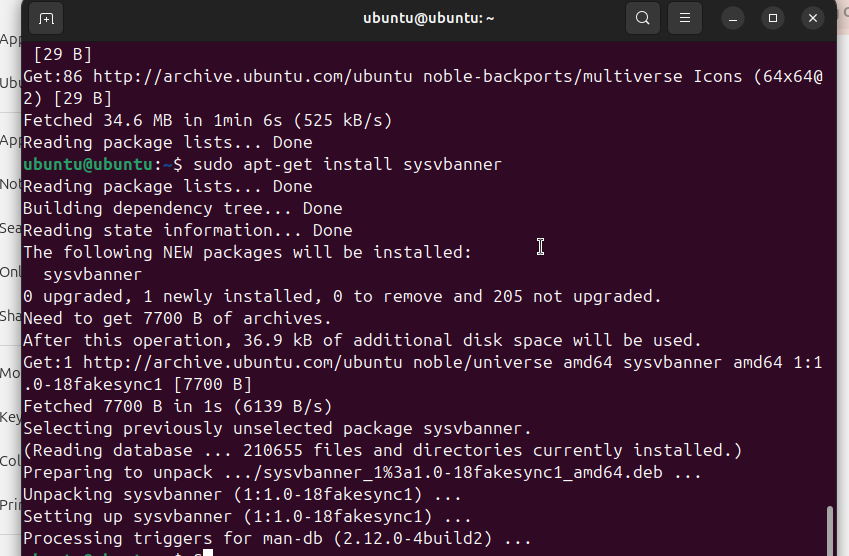
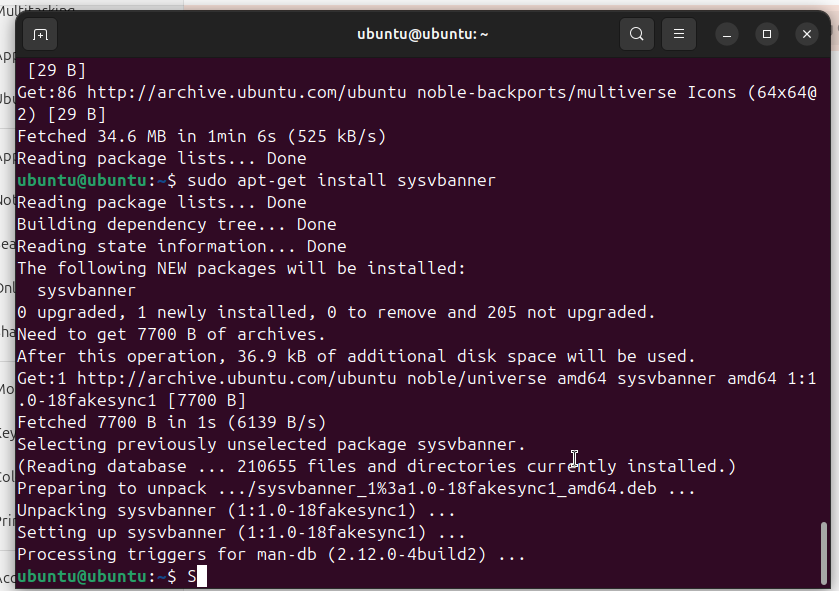
1. **操作截图和简要说明**

**banner**

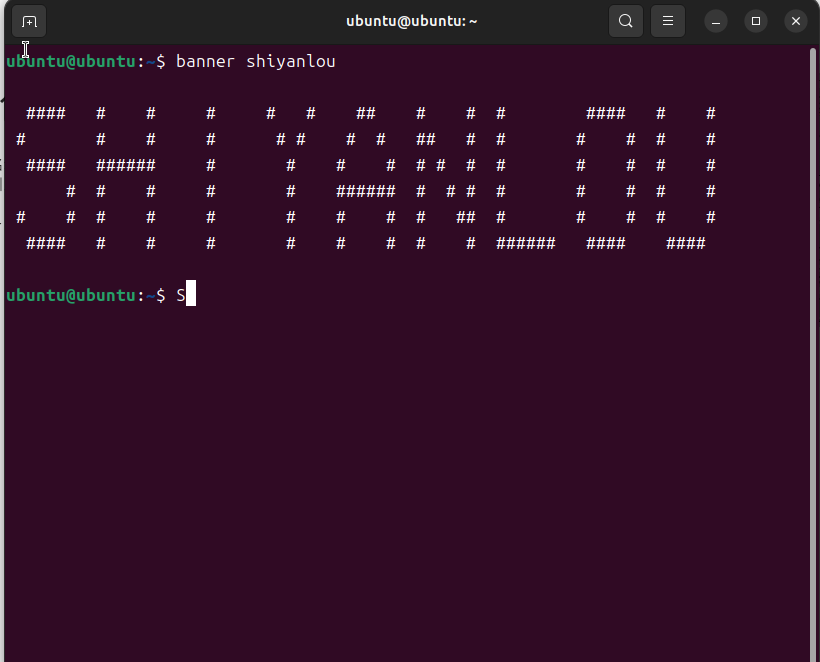
1.先使用如下命令安装：

sudo apt-get update

sudo apt-get install sysvbanner



1. 然后banner shiyanlou

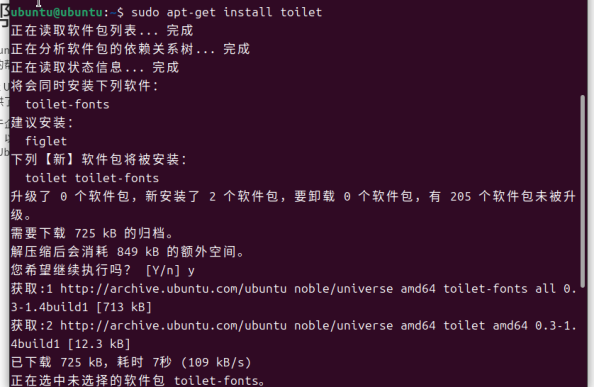


1. 使用默认已经安装的一个命令printerbanner：printerbanner -w 50 A

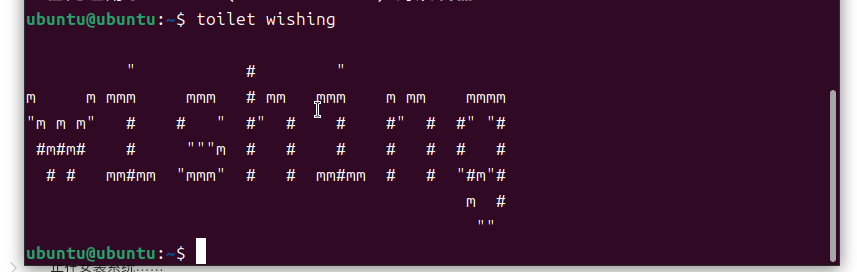


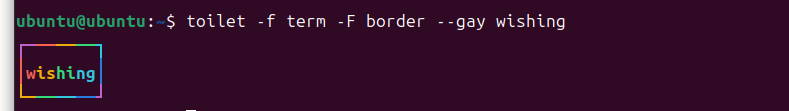
**toilet & figlet**

1. 使用sudo apt-get install toilet和sudo apt-get install figlet安装



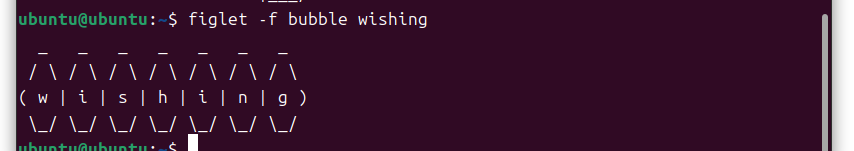
1. 使用toilet

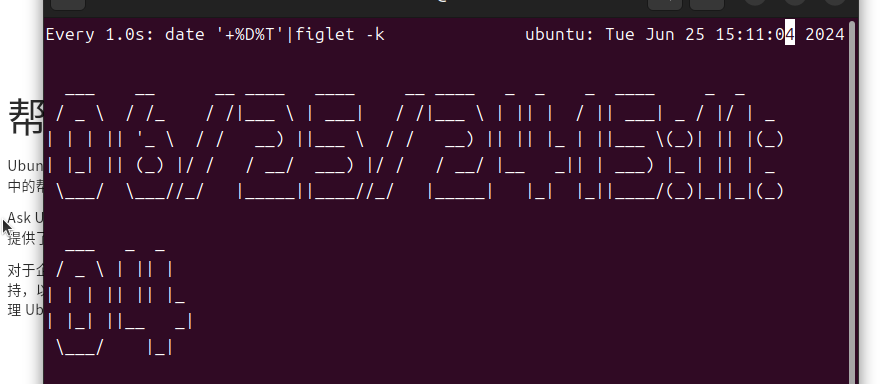




1. 使用figlet







**3.其他**

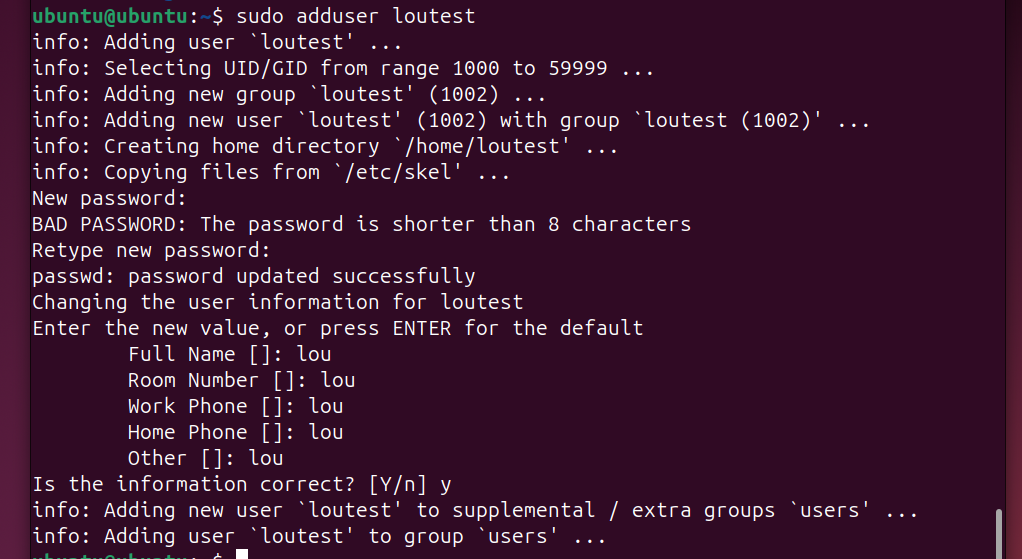
学习使用了banner，toilet，figlet三中命令显示，三种命令有不同的效果，让我感受到了终端显示文字的趣味性

**实验3**

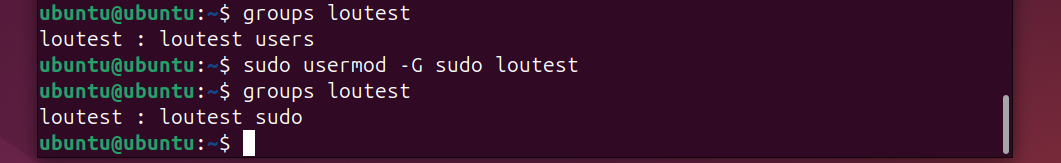
**1.作业要求概述**

添加一个用户 loutest，使用 sudo 创建文件 /opt/forloutest，设置成用 户 loutest 可以读写。

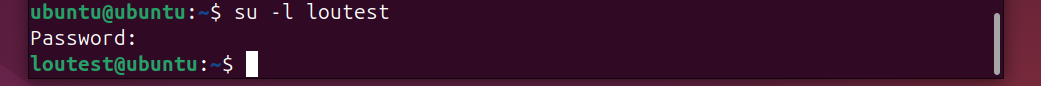
1. **操作截图和简要说明**
2. 使用sudo创建loutest用户



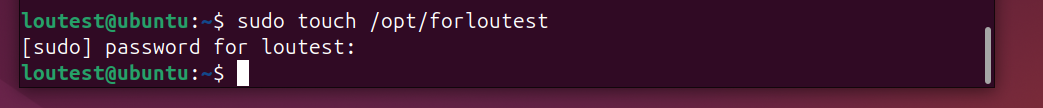
1. 给loutest赋予sudo权限



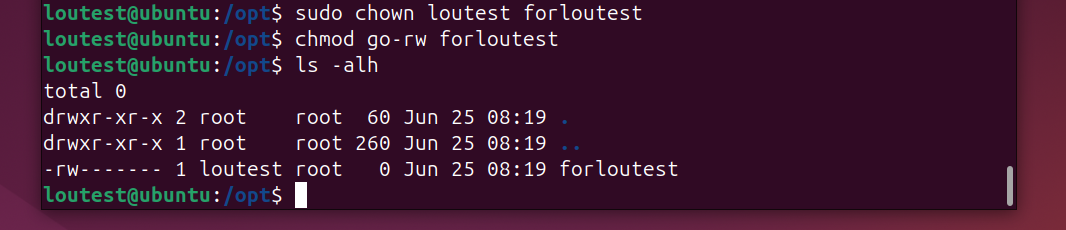
1. 切换到 loutest用户



1. loutest使用sudo创建文件



1. 将文件所有者设置为loutest并设置成loutest可读写



**3.其他**

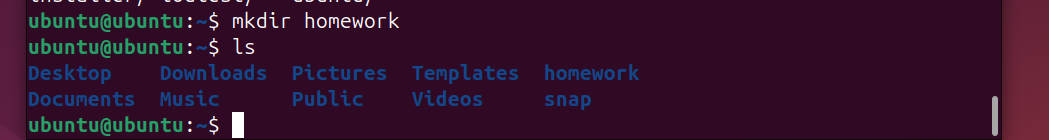
**实验4**

**1.作业要求概述**

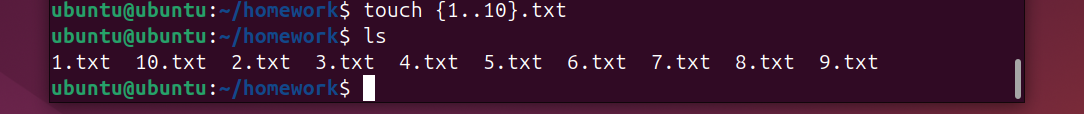
1. 创建一个 homework 目录，建立名为 1.txt ～ 10.txt 文件，并删除 1.txt ～ 5.txt
2. Linux 的日志文件在哪个目录？

**2.操作截图和简要说明**

1. 创建一个homewordk目录



2.创建名为1.txt ~ 10.txt 文件



1. 删除1.txt ~ 5.txt



Linux的日志文件存放在/var/log/的目录下

**3.其他**

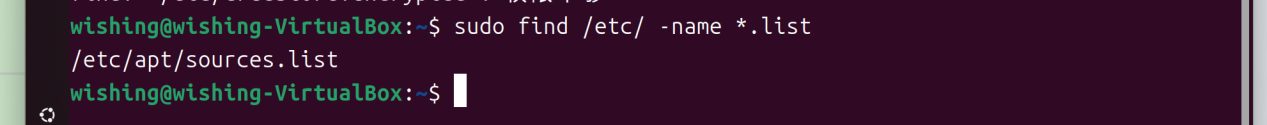
**实验5**

**1.作业要求概述**

找出 /etc/ 目录下的所有以 .list 结尾的文件

**2.操作截图和简要说明**

1. 使用sudo find来查找



**3.其他**

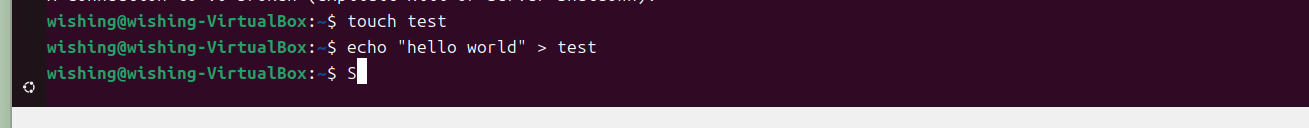
**实验6**

**1.作业要求概述**

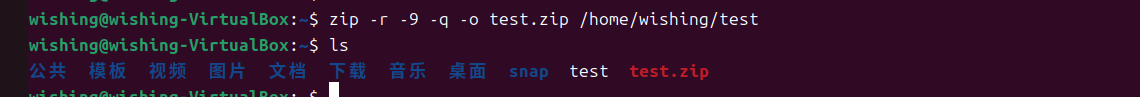
创建一个名为 test 的文件，分别用 zip 和 tar 打包成压缩包，再解压 到 /home/shiyanlou 目录

**2.操作截图和简要说明**

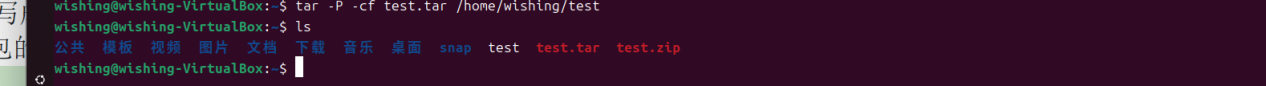
1. 创建test文件，并往里面随便放一点内容



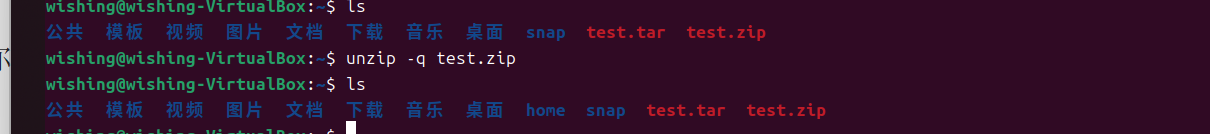
1. 使用zip压缩



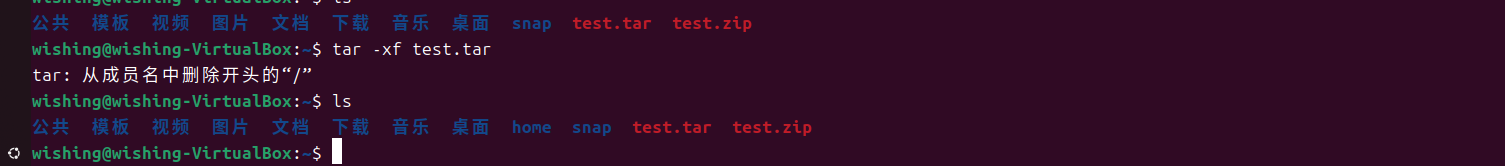
1. 使用tar打包



1. 使用unzip解压缩（可以看到解压后多了一个home的文件夹）



1. 使用tar解包（可以看到解包后多了一个home的文件夹）



**3.其他**

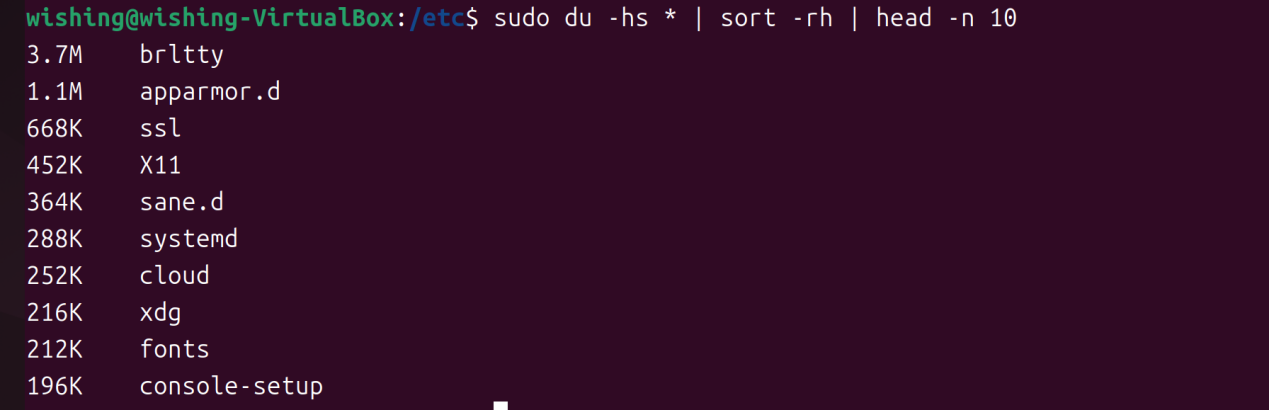
**实验7**

**1.作业要求概述**

找出当前目录下面占用最大的前十个文件。

**2.操作截图和简要说明**

1. 使用du来查看当前目录下的文件大小并排序



**3.其他**

**实验10**

**1.作业要求概述**

熟悉 cut，sort，uniq 命令以及参数

**2.操作截图和简要说明**

1. cut

cut命令用于从每一行中提取指定的部分，例如提取特定的列或字符。

* -b：指定要显示的字节。
* -c：指定要显示的字符。
* -d：指定字段分隔符，默认是Tab。
* -f：指定要显示的字段，需配合-d使用。

2. sort

sort命令用于对文件或标准输入的数据进行排序。

* -n：按数值排序。
* -r：逆序排序。
* -k：指定排序字段。
* -t：指定字段分隔符。
* -u：去除重复行。

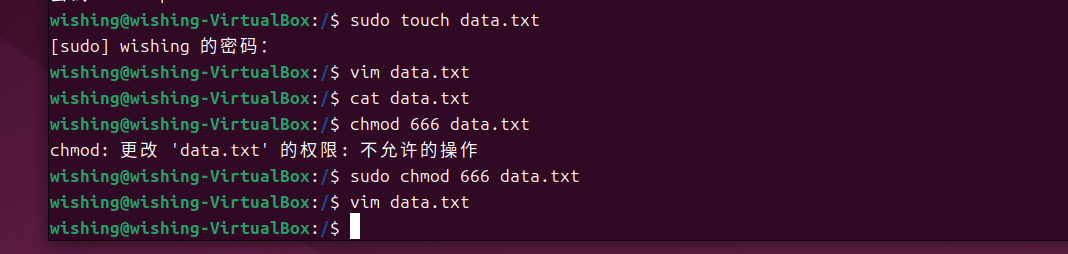
3. uniq

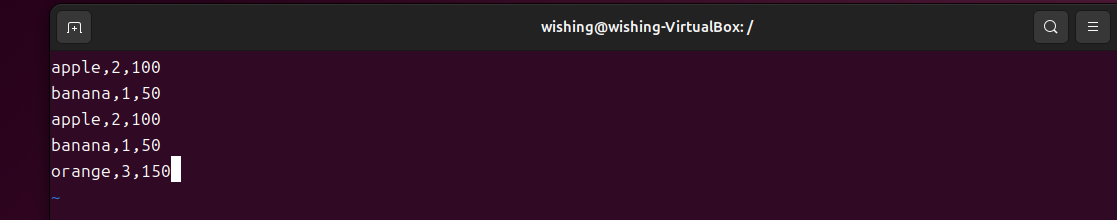
uniq命令用于删除或报告重复行，通常需要与sort命令一起使用，因 为uniq只能处理连续的重复行。

* -c：统计每行出现的次数。
* -d：仅显示重复的行。
* -u：仅显示不重复的行。

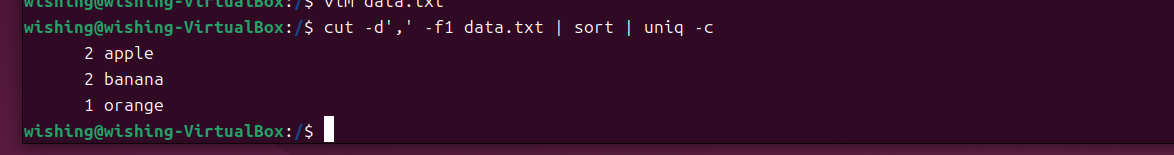
使用

先创建一个data.txt并写入一下内容





使用cut -d',' -f1 data.txt | sort | uniq -c来查看结果



**3.其他**

**实验11**

**1.作业要求概述**

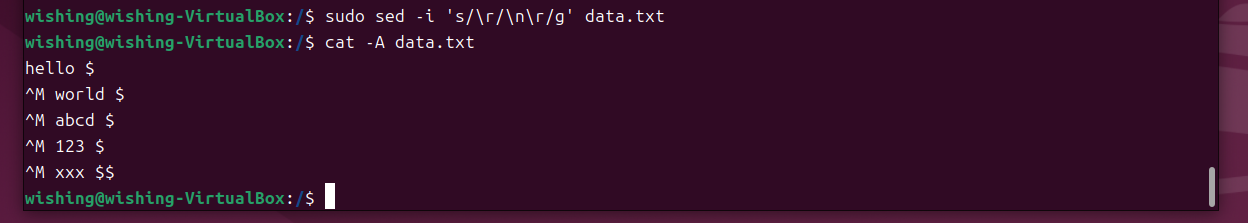
在不使用上述两个转换工具的情况下，使用前面学过的命令手动完成 dos 文本格式到 UNIX 文本格式的转换。

**2.操作截图和简要说明**

1. 使用cat -A查看所有字符



1. 使用sed -i 's/\r/\n\r/g' data.txt将\r替换成\n\r



**3.其他**

**实验12**

**1.作业要求概述**

理解下面这段代码的作用，实际这段代码不会正常工作，请结合这一小节的知识分析这段代码没有正确工作的原因，并设法解决这个问题。

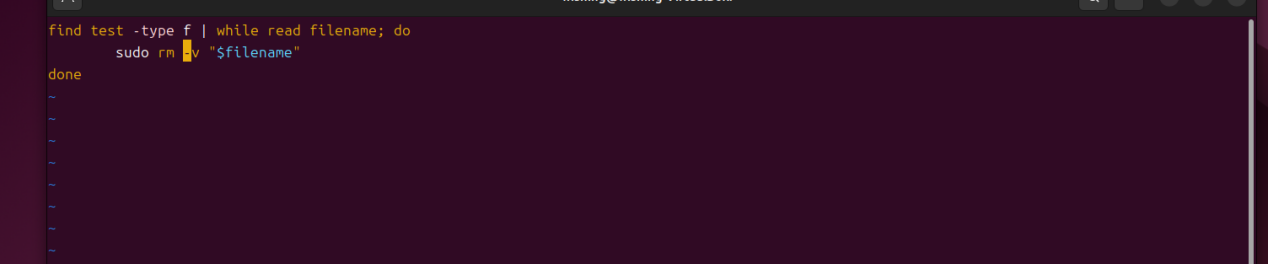
while read filename; do

rm -iv $filename

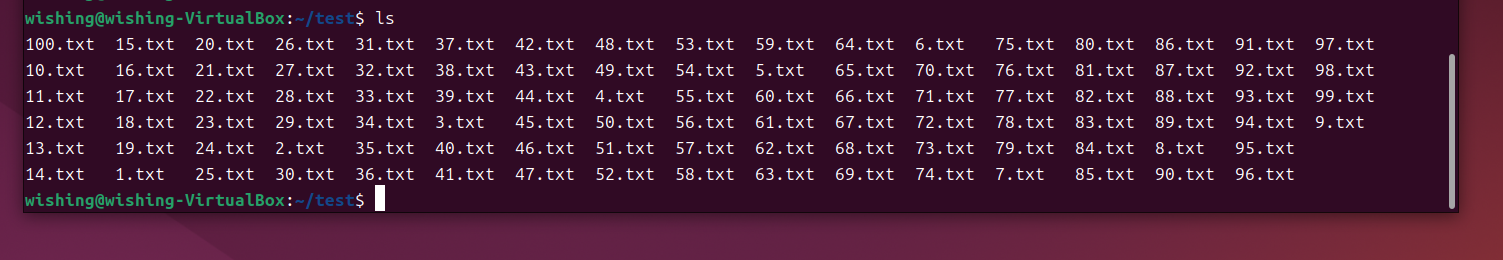
done << (ls)

**2.操作截图和简要说明**

这段代码主要用于遍历目录下的所有文件，然后删除文件。该代码错在试图将ls的结果通过重定向的方式重定向到while，且在删除文件时文件名没有加””。以下是改进后的代码，使用管道将find的结果重定向给while



删除前的文件夹



删除后，可以看到已经没有任何文件了



**3.其他**

**实验13**

**1.作业要求概述**

练习 1: 结合正则表达式做更多练习。

练习 2: 参考下面的链接，掌握 sed 处理文本的基本原理，理解 pattern space 和 hold space 概念。

练习 3: 基于 pattern space 和 hold space 实现将一个文本倒序输出和交换奇数行和偶数行。

**2、一个在线游戏，当然我们主要目的是学习，这个游戏也是有寓教于乐的性质，让你快速学会 vim 的基础操作：**

**2.操作截图和简要说明**

**匹配邮箱地址**:

grep -E '[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+' filename.txt

**匹配电话号码**:

grep -E '\b\d{3}[-.]?\d{3}[-.]?\d{4}\b' filename.txt

**匹配日期格式**:

grep -E '\b\d{4}-\d{2}-\d{2}\b|\b\d{2}/\d{2}/\d{4}\b' filename.txt

### ****基本原理****

sed 是一种流编辑器，它逐行读取输入文件，对每一行应用指定的命令，然后输出 结果。sed 处理文本的主要步骤包括：

1. 读取输入文件的一行到 pattern space。
2. 对 pattern space 应用命令。
3. 将 pattern space 的内容输出（除非使用了 -n 选项）。
4. 重复上述步骤，直到处理完所有行。

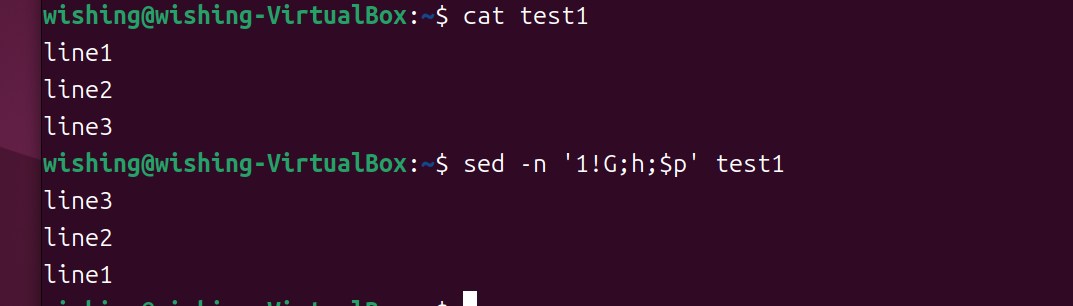
### ****pattern space 和 hold space 的概念****

### Pattern Space：pattern space 是当前处理的文本行的临时存储区。sed 命令主 要在 pattern space 上操作。

### Hold Space：hold space 是一个辅助存储区，用于暂时存储文本，以便在后续命 令中使用。hold space 与 pattern space 之间可以进行数据交换。

**倒序输出：**

sed -n '1!G;h;$p' input.txt

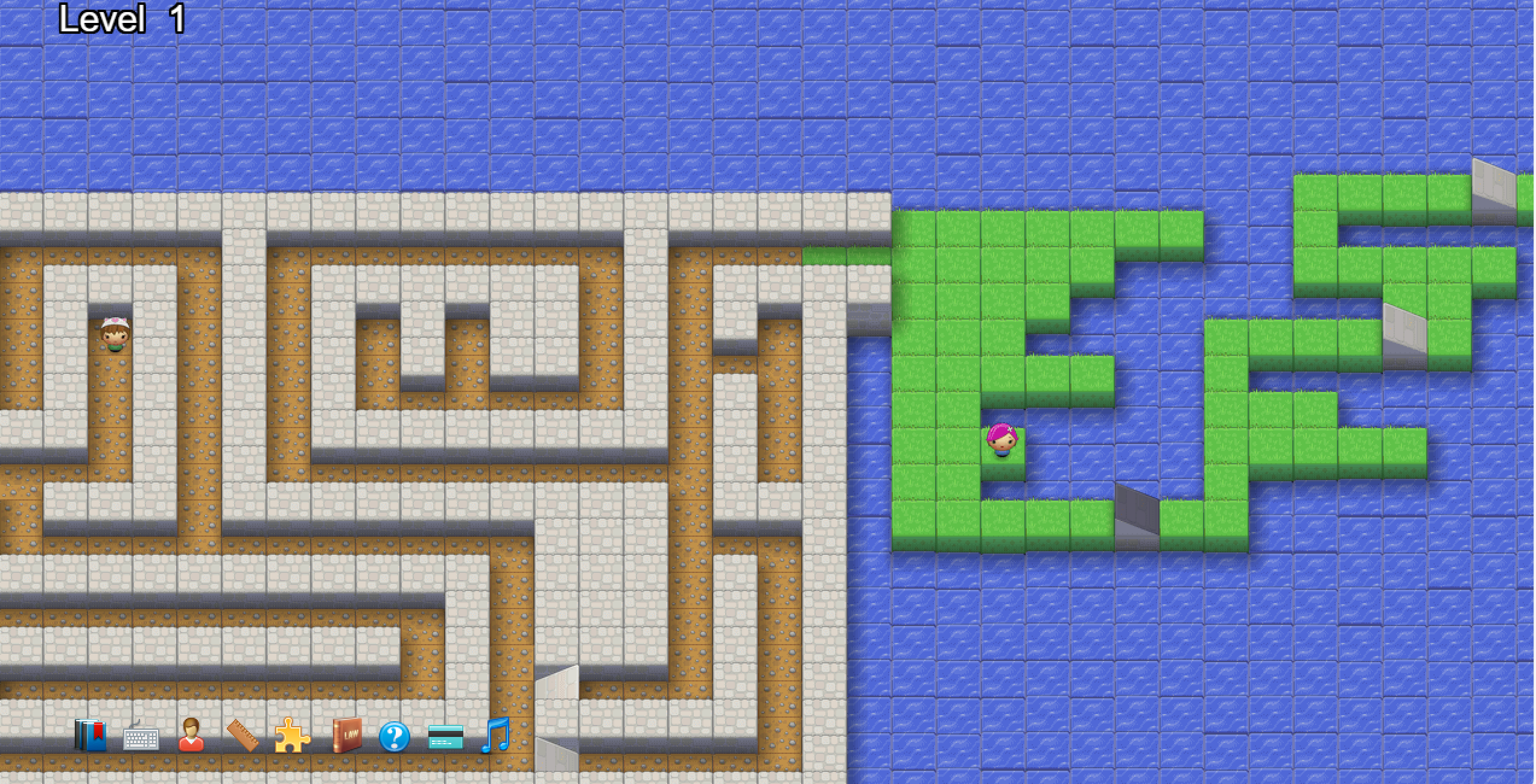


**交换奇数行和偶数行：**

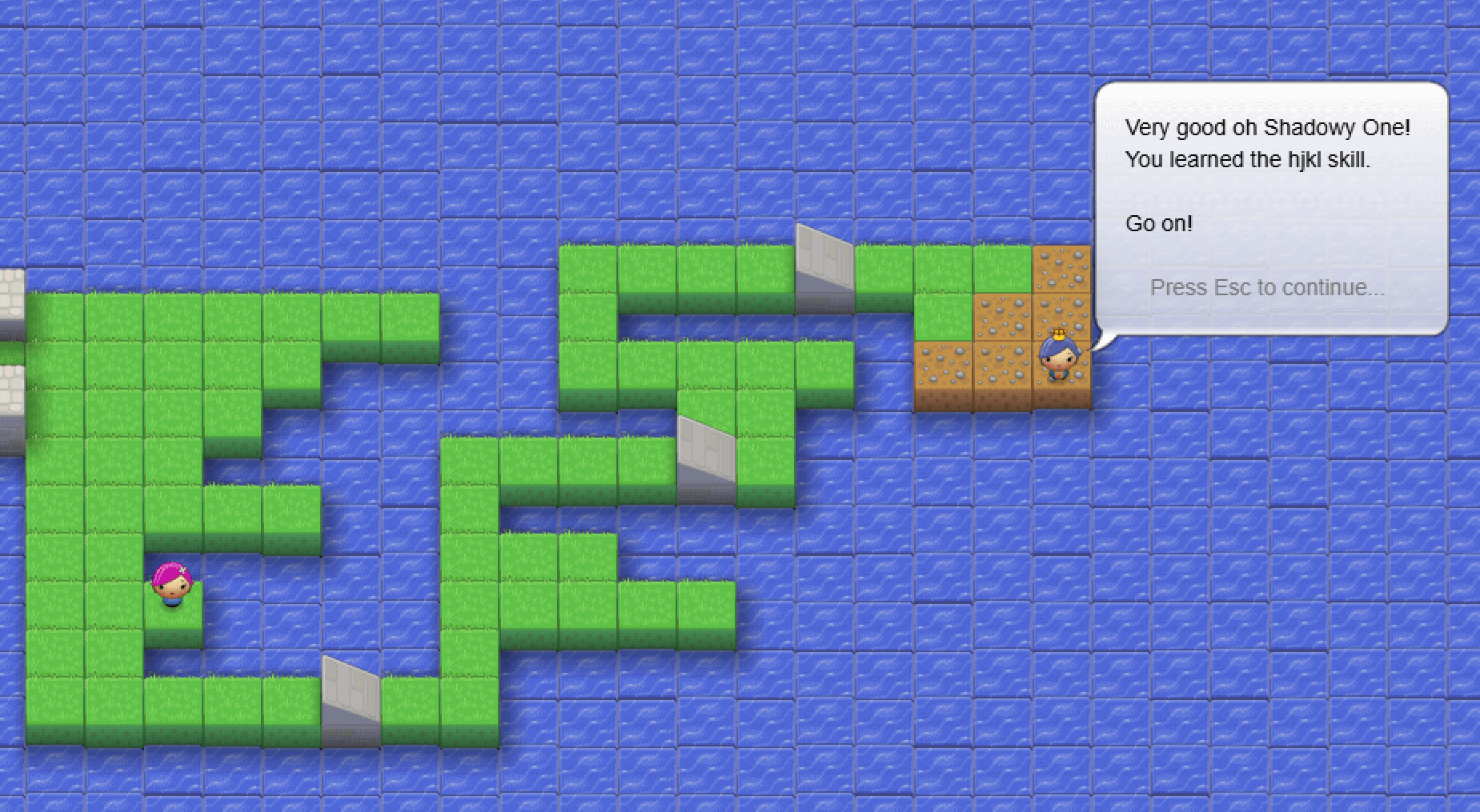
sed 'h;n;G' file.txt

这个命令首先将当前行存储在 hold space 中，然后跳到下一行，再将下一行与 hold space 进行交换，实现了奇数行和偶数行的交换操作。

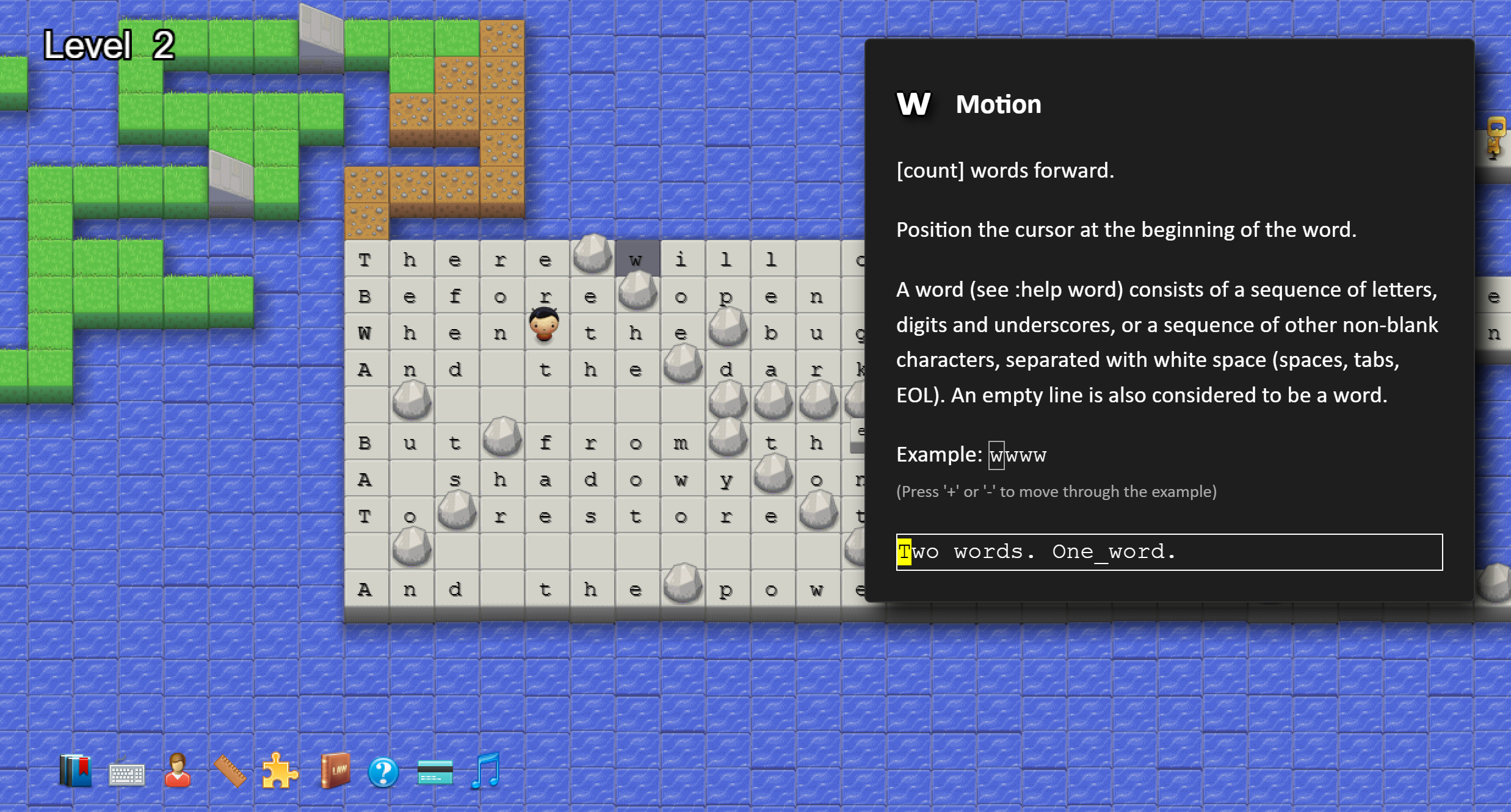
使用hjkl移动



走到末尾使用k到达另一行

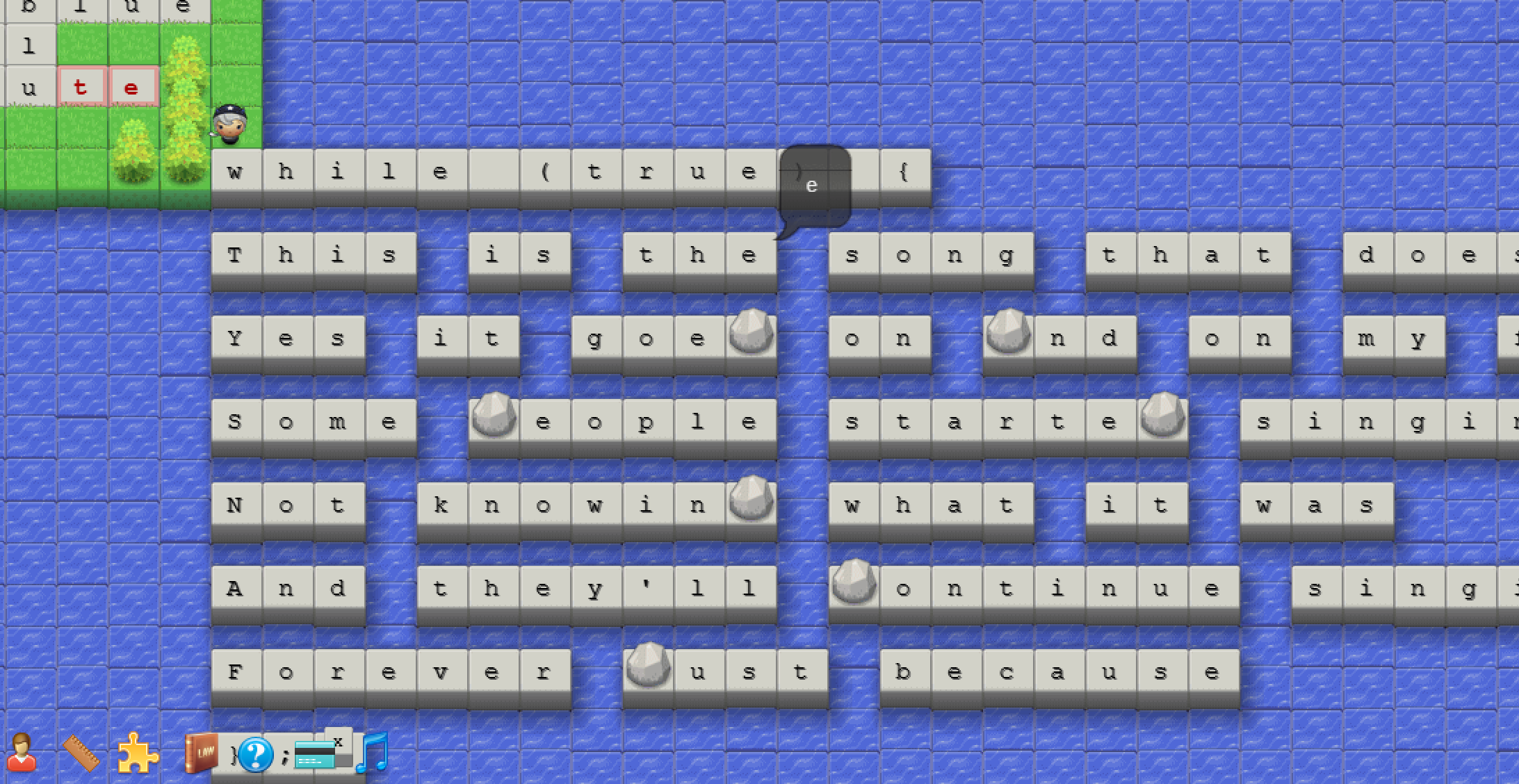


获得w和e，b，使用w和e和b可以越过障碍

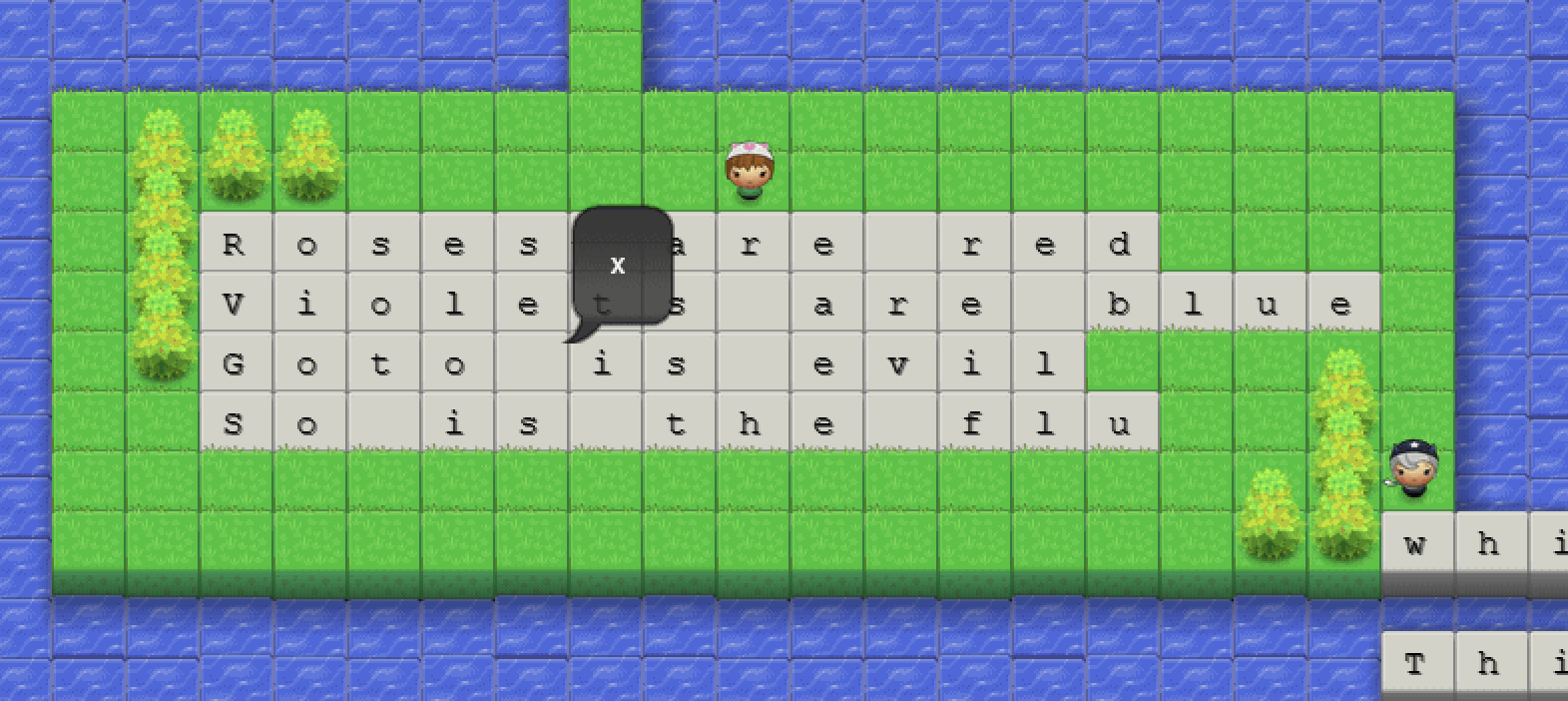




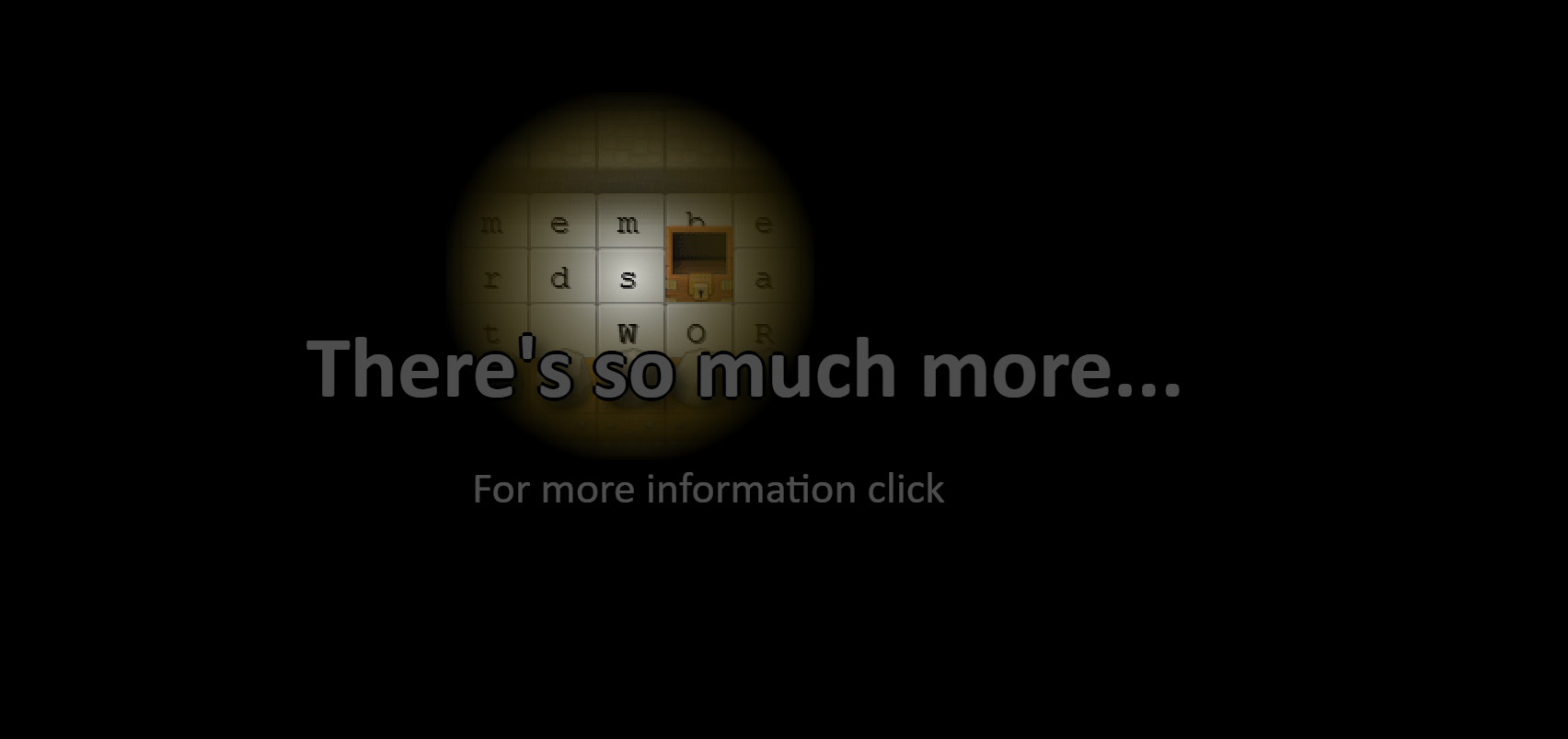
这里交替正确使用w和e可以到达最下面



获得x并删除某些字符



在！的格子中使用B大写可以向前越过障碍



**3.其他**