

# 西安电子科技大学《Python 程序设计》课程实验

## 实验名称 hangman 游戏实现

### 一、实验目的

- (1) 熟练掌握 Python 语言中的循环语句和条件语句。
- (2) 熟悉字符串的 `split()` 方法。
- (3) 掌握成员运算符的使用。
- (4) 熟悉模块 `string` 中常用的常量。

### 二、实验内容

编写 4 个函数，实现电脑和玩家交互的文字游戏 hangman。

### 三、实验环境

【填写自己所用环境】

### 四、实验原理和步骤

本次实验，我们编程实现一个经典的文字游戏 Hangman。第二个玩家是电脑，它将随机生成一个单词。首先我们会实现一些简单的帮助函数，接着实现 `hangman` 函数，然后通过电脑和玩家的交互来进行整个游戏。

在这个游戏中，我们需要一个输入文件 `words.txt`。相关的代码已经给出来了。如果一切正常，运行之后你将会看到屏幕上有以下输出：

```
Loading word list from file...
```

```
55909 words loaded.
```

如果有错误发生，比如此文件找不到，那么你应该检查此文件是否和

你的脚本文件在同一个文件夹，或者文件名是否写正确。比如，你可以将文件的相对路径改为绝对路径：如果你的文件放在目录 "C:/Users/Ana/"

```
WORDLIST_FILENAME = "words.txt"
```

修改为：

```
WORDLIST_FILENAME = "C:/Users/Ana/words.txt"
```

目录依赖于你放置文件的位置。

电脑将从文件 words.txt 中随机选择一个单词。这个游戏是交互式的，流程如下：

1. 游戏开始之前，让玩家知道电脑选择的单词的长度，也就是单词里面有多少个字符。
2. 每一轮，让用户去猜一个字符。
3. 用户猜完之后，立马给用户反馈所猜的字符是否出现在单词中。
4. 每一轮结束之后，显示用户当前所猜中的字符，以及还没有猜过的所有字符。

**游戏额外的规则如下：**

1. 每个用户最多有 8 次猜测机会，可以设置为一个常量。每一轮之后提醒用户还剩下几次。每次假设用户输入的是 26 个字母中的一个。
2. 只有当用户猜错字符的时候才失去一次机会。
3. 当用户所猜字符与之前所猜字符相同时，不要减少用户的次数，而是打印消息提醒用户此字符已经猜过了，然后让用户重新猜一次。
4. 当用户猜出单词或者机会用完时，游戏结束。如果游戏结束时，用户没有猜出来，则最后将单词显示给用户。

输入输出示例：（输入输出示例中，除了输入，其他字符串都是你的程序要输出的内容。功能都要实现，但是具体的输出内容大家可以随

意设计。)

玩家赢了时的程序输出类似这样：

从文件中加载单词列表...

加载了 55909 个单词。

欢迎来到 Hangman 游戏！

我选择了一个包含 7 个字母的单词。

\*\*\*\*\*

你还剩 8 次机会。

可用字母如下：abcdefghijklmnopqrstuvwxyz

请猜一个字母：e

哎呀！那个字母不在所猜单词中： \_ \_ \_ \_ \_ \_ \_

\*\*\*\*\*

你还剩 7 次机会。

可用字母如下：abcdefghijklmnopqrstuvwxyz

请猜一个字母：t

哎呀！那个字母不在所猜单词中： \_ \_ \_ \_ \_ \_ \_

\*\*\*\*\*

你还剩 6 次机会。

可用字母如下：abcdefghijklmnopqrstuvwxyz

请猜一个字母：i

哎呀!那个字母不在所猜单词中: \_ \_ \_ \_ \_

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: abcdefghijklmnopqrstuvwxyz

请猜一个字母: o

猜对了: \_ o \_ \_ \_ o

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: abcdefghijklmnopqrstuvwxyz

请猜一个字母: a

猜对了: \_ o \_ a \_ o

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: bcddefghijklmnopqrstuvwxyz

请猜一个字母: t

哎呀! 你已经猜过那个字母了: \_ o \_ a \_ o

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: bcddefghijklmnopqrstuvwxyz

请猜一个字母: p

猜对了: po \_ pa \_ o

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: bcd fghjklmnqrsuv wxyz

请猜一个字母: t

哎呀! 你已经猜过那个字母了: po\_ pa\_ o

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: bcd fghjklmnqrsuv wxyz

请猜一个字母: m

猜对了: pompa\_ o

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: bcd fghjkl nqrsuv wxyz

请猜一个字母: u

哎呀! 那个字母不在所猜单词中: pompa\_ o

\*\*\*\*\*

你还剩 4 次机会。

可用字母如下: bcd fghjkl nqrsv wxyz

请猜一个字母: x

哎呀! 那个字母不在所猜单词中: pompa\_ o

\*\*\*\*\*

你还剩 3 次机会。

可用字母如下: bcd fghjkl nqrsv wyz

请猜一个字母: s

哎呀!那个字母不在所猜单词中: pompa\_ o

\*\*\*\*\*

你还剩 2 次机会。

可用字母如下: bcd fghjkl nqrvwy z

请猜一个字母: n

猜对了: pompano

\*\*\*\*\*

恭喜, 你赢了!

玩家输了时的程序输出类似这样:

欢迎来到 Hangman 游戏!

我选择了一个包含 5 个字母的单词。

\*\*\*\*\*

你还剩 8 次机会。

可用字母如下: abcdefghijklmnopqrstuvwxyz

请猜一个字母: a

猜对了: \_ a \_ \_ \_

\*\*\*\*\*

你还剩 8 次机会。

可用字母如下: bcdefghijklmnopqrstuvwxyz

请猜一个字母: e

哎呀!那个字母不在所猜单词中: \_ a\_ \_ \_

\*\*\*\*\*

你还剩 7 次机会。

可用字母如下: bcd fghi jklmnopqrstuvwxyz

请猜一个字母: t

哎呀!那个字母不在所猜单词中: \_ a\_ \_ \_

\*\*\*\*\*

你还剩 6 次机会。

可用字母如下: bcd fghi jklmnopqrsuvwxyz

请猜一个字母: o

哎呀!那个字母不在所猜单词中: \_ a\_ \_ \_

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: bcd fghi jklmnpqrsuvwxyz

请猜一个字母: i

猜对了: \_ ai\_ \_

\*\*\*\*\*

你还剩 5 次机会。

可用字母如下: bcd fghjklmnpqrsuvwxyz

请猜一个字母: f

哎呀!那个字母不在所猜单词中: \_ ai\_ \_

\*\*\*\*\*

你还剩 4 次机会。

可用字母如下: bcdghjklmnpqrsuvxyz

请猜一个字母: h

哎呀!那个字母不在所猜单词中: \_ ai\_ \_

\*\*\*\*\*

你还剩 3 次机会。

可用字母如下: bcdgijklmnpqrsuvxyz

请猜一个字母: p

哎呀!那个字母不在所猜单词中: \_ ai\_ \_

\*\*\*\*\*

你还剩 2 次机会。

可用字母如下: bcdgijklmnpqrsuvxyz

请猜一个字母: m

猜对了: mai\_ \_

\*\*\*\*\*

你还剩 2 次机会。

可用字母如下: bcdgijklnqrsuvxyz

请猜一个字母: l

猜对了: mail\_

\*\*\*\*\*

你还剩 2 次机会。

可用字母如下: bcdgjklnqrsuvxyz



请猜一个字母: 1

哎呀! 你已经猜过那个字母了: mail\_

\*\*\*\*\*

你还剩 2 次机会。

可用字母如下: bcdgjk nqrsuv wxyz

请猜一个字母: u

哎呀!那个字母不在所猜单词中: mail\_

\*\*\*\*\*

你还剩 1 次机会。

可用字母如下: bcdgjk nqrs v wxyz

请猜一个字母: v

哎呀!那个字母不在所猜单词中: mail\_

\*\*\*\*\*

非常抱歉, 猜测次数已经用完了, 所猜单词是: mails

时间和精力允许的情况下, 我们还可以在每次猜测之后输出图像, 类似如下画面:

初始状态

```
+---+
|   |
|   |
|   |
|   |
|   |
|   |
|   |
|   |
=====
```

猜错 5 次时的状态	<pre> +---+        0     / \    /             ===== </pre>
猜测次数用完的状态	<pre> +---+       [0]    / \    / \           ===== </pre>

我们可以将整个程序分解为不同的函数，比如

### 1) 单词是否被猜中

```

1 secretWord = 'apple'
2 lettersGuessed = ['e', 'i', 'k', 'p', 'r', 's']
3
4 print(isWordGuessed(secretWord, lettersGuessed))

```

False

如果 secretWord 被正确猜到（也就是说，在 secretWord 中的所有字母都在 lettersGuessed 中）函数输出 TRUE，其他情况输出 False。

### 2) 打印猜测的单词

```

1 secretWord = 'apple'
2 lettersGuessed = ['e', 'i', 'k', 'p', 'r', 's']
3 print(getGuessedWord(secretWord, lettersGuessed))

```

\_ pp\_ e

当输出字符串时，在每个下划线之后加入至少一个空格会是一个好主意。它会使你的程序输出更清晰，让玩家很明确地知道还有几个字母

没有猜出来。（比较这两种方式的可读性 \_\_\_\_\_ 和 \_ \_ \_ \_ ）.

### 3) 打印可用（还未猜过的）字母

```
1 lettersGuessed = ['e', 'i', 'k', 'p', 'r', 's']
2 print(getAvailableLetters(lettersGuessed))
```

abcdfghjlmnoqtuvwxyz

### 4) 导入单词文件 words.txt

实现一个函数，返回单词列表，每个单词是由小写字母构成的字符串。

### 5) 随机选择一个单词

借助于 random 模块中的 choice 函数，随机返回一个单词作为 secretWord.

### 6) 实现 hangman 函数

- \* 第一步：输出单词 secretWord 的长度；
- \* 第二步：每一轮让用户猜测一个字母，并且立即给用户反馈，所猜字母是否在 secretWord 中；
- \* 第三步：在每一轮结束之后，输出目前为止用户猜对的字母，以及还未猜过的字母

调用 hangman 函数进行测试。

## 五、实验要求

按照试验内容编写程序，实验结束之后整理结果并提交电子版实验报告。