**西安电子科技大学**

**安全前沿讨论班（I） 课程实验报告**

**实验名称 实现hangman游戏**

学 院 网络与信息安全学院 班 级1618019

姓 名 曹寅峰 学 号16020610025

实验日期 2019 年 4 月 17 日

|  |
| --- |
| 指导教师评语：  指导教师：  年 月 日 |
| **实验报告内容基本要求及参考格式**   1. 实验目的   实现一个hangman交互式猜字游戏，第二个玩家总是计算机，它会随机选择一个单词，根据用户猜测的字母返回相应的结果，用户一共有8次机会。   1. 实验环境   Python3 jupyternotebook   1. 实验基本原理及步骤（或方案设计及理论计算）     首先根据范例弄懂游戏的具体细节，每次输入一个字母，如果存在则返回欠缺的部分（重复的时候把字母都填上）  根据需求，我们把游戏判断部分分为三个函数：  1. isWordGuessed(secretWord, lettersGuessed)  这个函数用作胜利的判断条件，如果输入过的猜测字母所组成的列表包含所有的秘密单词中的字母，则胜利。    列举secretword中的每个单词，如果在lettersGuessed内，则继续列举下一个，直到所有的字母都在lettersGuessed内；如果有一个不在，则返回失败。  2. getGuessedWord(secretWord, lettersGuessed):  这个函数用于返回用户的猜测状态，如果没有出现，则用 \_ 代替字母  思路同样是对比secretWord中的字母，如果在lettersGuessed中，则在res中加上该字母；如果不在，则补上 \_  3. getAvailableLetters(lettersGuessed):  这个函数用于更新用户猜测后的字母列表  思路同样类似，每次对比字母表与当前列表，如果存在说明猜测过，则跳过；不存在说明还没有猜测过，该字母尚可以使用  4.hangman(): 最终我们需要将上面3个函数结合起来，完成整个游戏。内置的文件读取部分已经写好，完成其余部分即可。    首先要调用getAvailableLetters打印出当前的单词，其次调用isWordGuessed判断是否已经可以获得胜利，如果没有的话则判断这个字母是否正确，通过调用getGuessedWord返回尝试结果。将整个过程循环八次（最后加上字符画）   1. 实验结果分析及回答问题（或测试环境及测试结果）   1.    2.  3.    4.      可以看到，结果均与示例相同 |