-20171660 이건민
-13주차 과제 -hangman 게임 만들기
-4분반

Guess.py

미널 파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

```
1  class Guess:
2
3      def __init__(self, word):
4          ## 단어를 secretword에,
5          self.secretWord = word
6          self.guessedChars = []
7          self.numTries = 0
8          self.currentStatus = '_' * len(self.secretWord)
9
10     def display(self):
11         print('Current : ' + self.currentStatus)
12         print('Tries : ' + str(self.numTries))
13
14
15     def guess(self, character):
16         ## 문자를 이미 입력했었던 경우
17         if character in self.guessedChars:
18             pass
19         ## 문자를 새로 입력하는 경우
20         else:
21             ## 문자가 단어에 있는경우
22             if self.secretWord.find(character) != -1:
23                 self.guessedChars.append(character)
24                 ## 문자가 있는 부분의 인덱스만 따로 초기화
25                 for i in range(len(self.secretWord)):
26                     if character == self.secretWord[i]:
27                         self.currentStatus = self.currentStatus[:i] + character + self.currentStatus[i+1:]
28                 if self.currentStatus == self.secretWord:
29                     print('Answer is ' + self.secretWord)
30                     return True
31             ##문자가 단어에 없는 경우
32             else:
33                 self.guessedChars.append(character)
34                 self.numTries += 1
35
```

입력받은 단어를 secretWord 안에 넣는다.
입력받은 문자들을 저장하기 위해 guessedChars라는 리스트를 만들음.
입력받은 경우를 세는 numTries라는 변수를 만듦.
문제 상태를 나타내기 위해 처음 currentStatus를 받은 단어의 길이의 _만큼 지정.

17. if character in self.guessedChars:
입력받은 문자가 guessedChars라는 리스트에 있다면 그냥 passg나다

22. if self.secretWord.find(character) != -1:
문자가 단어 안에 포함된 경우
23.문자를 우선 guessedChars에 포함시키고
문자가 단어안에 위치한 인덱스를 알아내어
그 인덱스의 전후부분은 그대로 이전 currentStatus의 문자를 따오고 그 인덱스만 character로 교체한다.

문자를 다 찾아내면 True를 리턴한다

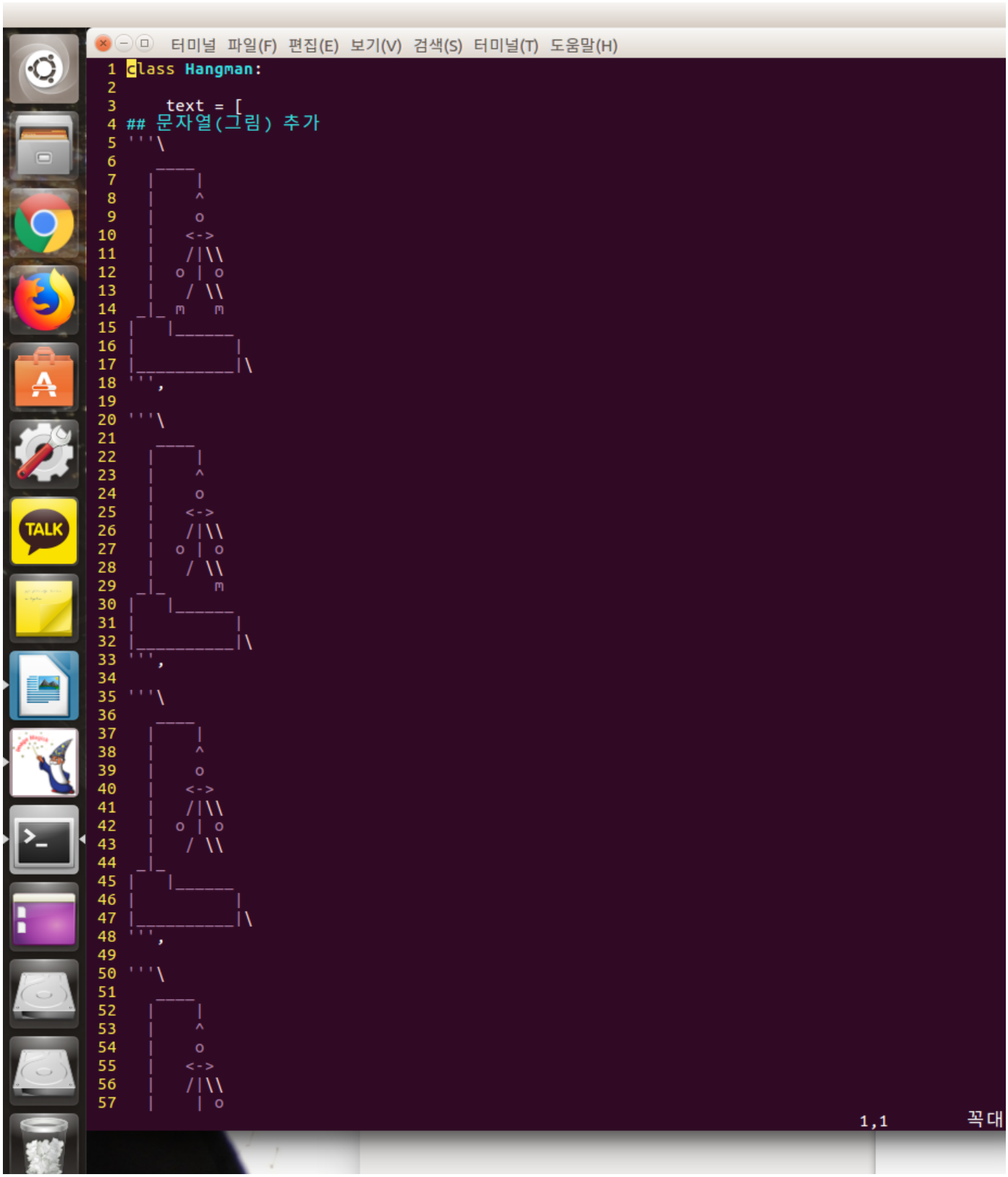
32.
문자가 단어 안에 없는경우
guessedChars에 문자를 포함시키고 numTriesf를 1추가한다

Game.py

```
터미널 파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
 1 from hangman import Hangman
 2 from guess import Guess
 3 from word import Word
 4
 5
 6 def gameMain():
 7     word = Word('words.txt')
 8     guess = Guess(word.randFromDB())
 9
10     finished = False
11     hangman = Hangman()
12     maxTries = hangman.getLife()
13
14     while guess.numTries < maxTries:
15
16         display = hangman.get(maxTries - guess.numTries)
17         print(display)
18         guess.display()
19
20         guessedChar = input('Select a letter: ')
21         if len(guessedChar) != 1:
22             ## 만약 사용자가 문자대신 단어를 입력해서 맞으면 success
23             if guessedChar == guess.secretWord:
24                 print('genius godgod smart!')
25                 finished = True
26                 break
27             else:
28                 print('One character at a time!')
29                 continue
30         if guessedChar in guess.guessedChars:
31             print('You already guessed \"' + guessedChar + '\"')
32             continue
33
34         finished = guess.guess(guessedChar)
35         if finished == True:
36             break
37     if finished == True:
38         print('Success')
39     else:
40         print(hangman.get(0))
41         print('word [' + guess.secretWord + ']')
42         print('guess [' + guess.currentStatus + ']')
43         print('Fail')
44
45
46 if __name__ == '__main__':
47     gameMain()
~
~
~
~
~
~
~
~
~
~
                                                          1,1           모두
```

21. 만약 사용자가 문자열이 아닌
답을 입력하면 finished에 True값을 넣어주고 while문을 빠져나온다.

Hangman.py



영어 단어가 어려워서 목숨을 늘렸다