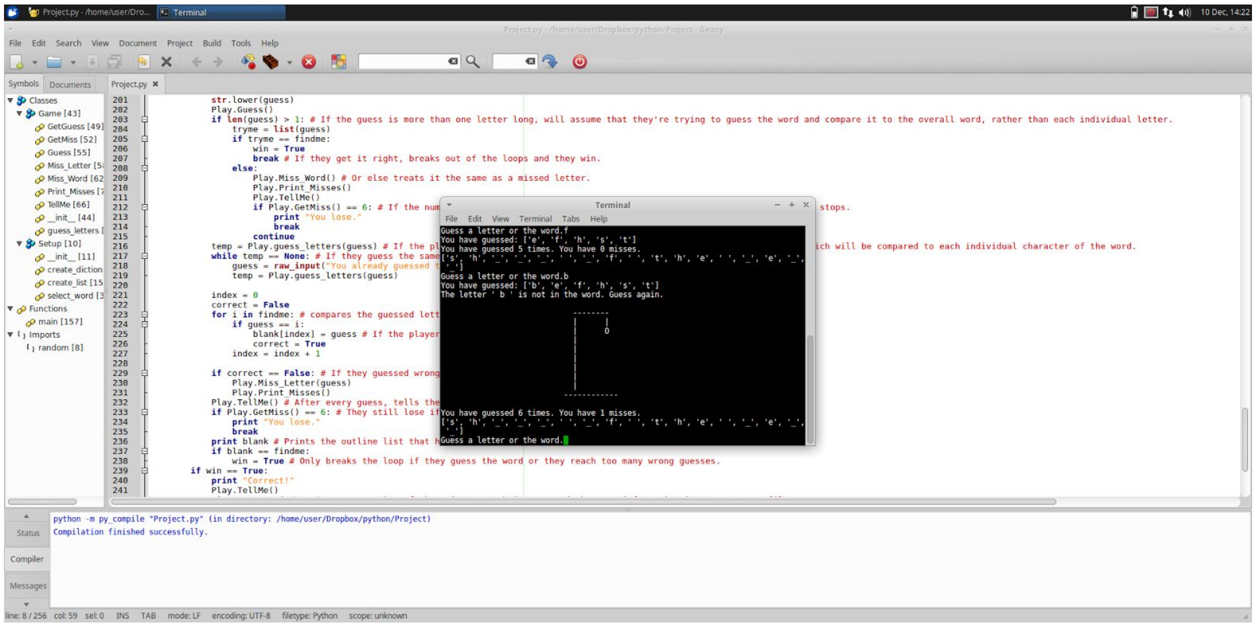
Adam Ten Hoeve
106105239
Amber Womack
Section 114

Python Project Writeup

For my project, I made hangman. Mostly because I couldn't think of another project that had a complex dictionary, but that's unimportant. The program starts by making a class that contains data members and methods for making the dictionary and the word that gets pulled out of it. The dictionary is created by reading a bunch of words from text files and putting those words into a list. Each list is like a "category," like animals and movies, for the player to select which type of word they would want to guess. Each of these lists is put into a dictionary, which is where the complex dictionary goes. The rest of the game is preformed in a while loops so the player can keep playing the game and selecting new categories as long as they want. The player is then prompted to select which category they would like to use. Using that, the class creates a random number (from importing the random library) between zero and the length of the category that they chose and pulls the word at that random location. The creating of the category lists, the addition of each of those to the dictionary, and the pulling of a random word are all done in methods of the class. The word is then broken into a list of its individual characters and a "blank" list is created that is the same length that is shown to the player to see how long the word is and that gets filled in with characters as the player guesses.

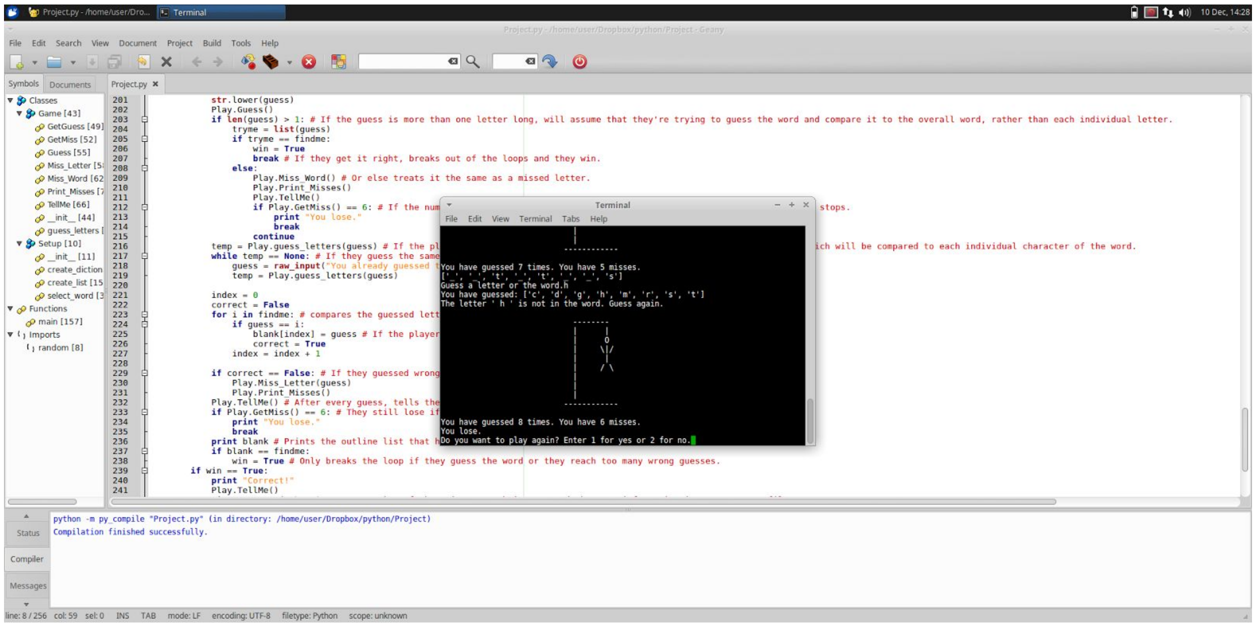
Once the blank list is created, the actual game part starts. A new class is created that tracks data like the amount of guesses and wrong guesses that the player has made, as well as methods for iterating through those variables and printing them to the player. The user is shown the blank list and prompted to guess either a letter or the word. If the length of the guess is longer than 1, then the game compares the guess to the word. If the guess and the word are the same, then they are correct and the loop exits, prompting the player if they want to play again. If they guess wrong, then the number of misses is increased by one and the hangman visual is portrayed. If the player gets to six guesses, then they lose and the loop exits. If the player's guess is only one letter long, then it get compared to each character in the word list with a for loop that compares every letter to the guessed letter, rather than entire word. This is one of the nested loops in the game, that the comparing each letter for loop is in the guessing while loop. If the guessed character is the same as any character in the word, then that character is added to the blank list so, when printed, shows that letter in the correct location. If the letter is not in the word, then they are incorrect, the misses counter increases, and the hangman is printed again. The game loop ends when the player either guesses the word, gets all of the characters correct, or makes six wrong guesses. The player is prompted to play again and the loop ends if they say no. If the player wins, then a new file is written out with the word that they guessed and how many guesses it took them.



```
        str.lower(guess)
        Play.Guess()
        if len(guess) > 1: # If the guess is more than one letter long, will assume that they're trying to guess the word and compare it to the overall word, rather than each individual letter.
            tryme = list(guess)
            if tryme == findme:
                win = True
                break # If they get it right, breaks out of the loops and they win.
            else:
                Play.Miss_Word() # Or else treats it the same as a missed letter.
                Play.Print_Misses()
                Play.TellMe()
                if Play.GetMiss() == 6: # If the num                                    stops.
                    print "You lose."
                    break
                continue
        temp = Play.guess_letters(guess) # If the pl                                    ich will be compared to each individual character of the word.
        while temp == None: # If they guess the same
            guess = raw_input("You already guessed t
            temp = Play.guess_letters(guess)

        index = 0
        correct = False
        for i in findme: # compares the guessed lett
            if guess == i:
                blank[index] = guess # If the player
                correct = True
            index = index + 1

        if correct == False: # If they guessed wrong
            Play.Miss_Letter(guess)
            Play.Print_Misses()
        Play.TellMe() # After every guess, tells the
        if Play.GetMiss() == 6: # They still lose if
            print "You lose."
            break
        print blank # Prints the outline list that h
        if blank == findme:
            win = True # Only breaks the loop if they guess the word or they reach too many wrong guesses.
    if win == True:
        print "Correct!"
        Play.TellMe()
```

```
Guess a letter or the word.f
You have guessed: ['e', 'f', 'h', 's', 't']
You have guessed 5 times. You have 0 misses.
['s', 'h', '_', '_', '_', ' ', '_', 'f', ' ', 't', 'h', 'e', ' ', '_', 'e', '_',
 '_']
Guess a letter or the word.b
You have guessed: ['b', 'e', 'f', 'h', 's', 't']
The letter ' b ' is not in the word. Guess again.

                             --------
                             |      |
                             |      O
                             |
                             |
                             |
                             |
                             |
                             |
                           -----------

You have guessed 6 times. You have 1 misses.
['s', 'h', '_', '_', '_', ' ', '_', 'f', ' ', 't', 'h', 'e', ' ', '_', 'e', '_',
 '_']
Guess a letter or the word.
```

```
python -m py_compile "Project.py" (in directory: /home/user/Dropbox/python/Project)
Compilation finished successfully.
```



```
        str.lower(guess)
        Play.Guess()
        if len(guess) > 1: # If the guess is more than one letter long, will assume that they're trying to guess the word and compare it to the overall word, rather than each individual letter.
            tryme = list(guess)
            if tryme == findme:
                win = True
                break # If they get it right, breaks out of the loops and they win.
            else:
                Play.Miss_Word() # Or else treats it the same as a missed letter.
                Play.Print_Misses()
                Play.TellMe()
                if Play.GetMiss() == 6: # If the num                                    stops.
                    print "You lose."
                    break
                continue
        temp = Play.guess_letters(guess) # If the pl                                    ich will be compared to each individual character of the word.
        while temp == None: # If they guess the same
            guess = raw_input("You already guessed t
            temp = Play.guess_letters(guess)

        index = 0
        correct = False
        for i in findme: # compares the guessed lett
            if guess == i:
                blank[index] = guess # If the player
                correct = True
            index = index + 1

        if correct == False: # If they guessed wrong
            Play.Miss_Letter(guess)
            Play.Print_Misses()
        Play.TellMe() # After every guess, tells the
        if Play.GetMiss() == 6: # They still lose if
            print "You lose."
            break
        print blank # Prints the outline list that h
        if blank == findme:
            win = True # Only breaks the loop if they guess the word or they reach too many wrong guesses.
    if win == True:
        print "Correct!"
        Play.TellMe()
```

```
                             |
                           -----------

You have guessed 7 times. You have 5 misses.
['_', '_', 't', '_', 't', '_', '_', 's']
Guess a letter or the word.h
You have guessed: ['c', 'd', 'g', 'h', 'm', 'r', 's', 't']
The letter ' h ' is not in the word. Guess again.

                             --------
                             |      |
                             |      O
                             |     \|/
                             |      |
                             |     / \
                             |
                             |
                             |
                           -----------

You have guessed 8 times. You have 6 misses.
You lose.
Do you want to play again? Enter 1 for yes or 2 for no.
```

```
python -m py_compile "Project.py" (in directory: /home/user/Dropbox/python/Project)
Compilation finished successfully.
```