CIS4350_Lab02_PicoFermiBagel

Due: **03:00 PM** on **Friday, Oct 20**
Points: **100** points

Objective:
This assignment will demonstrate your knowledge of loops, classes, methods, and Swift data types.

About:
You will write a Swift program to play the game of Pico, Fermi, Bagel.

Rules:
Here are the rules of the game:

1. The computer will generate a "secret" three digit number at random.
2. The first number will not be 0, and all the digits will be different.
3. The user tries to guess the number.
    - If the user guesses correctly, then the game is over.
    - If not, the computer gives a hint and the player tries again.

Hints:

- For each digit that matches the secret number in the proper place, the computer prints "Fermi"
- For each digit that matches, but not in the proper place, the computer prints "Pico"
- If none of the digits match, the computer prints "Bagels"

Examples (If the program generated the secret number `482`):

```
Hi, I have a number on mind, guess that number?

guess = 637, Bagels
guess = 381, Fermi
guess = 382, Fermi Fermi
guess = 832, Fermi Pico
guess = 328, Pico Pico
guess = 428, Fermi Pico Pico
guess = 482, Winner!
```

(Game is over)

When the game is over, the results are printed: whether the user won or quit, and the number of guesses made

- To avoid making it too easy for the player, you should print all Fermis first, and then the Picos, for each guess.
- To avoid making it too difficult, print all guesses and hints to monitor so the user can scroll through them.

The Bagels Class (`Bagels.swift`)

Your Bagels class will have only one public method, `playGame()`. This method will call three other methods to:

1. Generate the secret number
2. Determine whether the current guess is a winner
3. Evaluate the current guess and print hints

Since these methods are called only by a method of the same class – `playGame()` - and not by the test class, it is customary to declare them private, instead of public. Such methods are sometimes called “utility” methods or “housekeeping” methods since they do “behind the scenes” chores.

OPTIONAL: You may have additional utility methods, also called from `playGame()`, to get the user’s guess and print the game results.

High-Level Algorithm for the playGame() Method

```
Generate the Secret Number

do
      Get User's Guess (including option to quit)
      If (! userQuits)
      Evaluate User's Guess
while ( not (winningGuess or userQuits) )

Print results – did user win or quit, and number of guesses made
```

Your Main class (`main.swift`)

Your main class will create a Bagels object and call the method that plays the game.
After each game, give the user the option of playing another.
After the last game, thank the user. That's all, nothing else.

Additional Specifications

1. The skeleton of the Bagels class you are to use is below. Write the methods bodies and declare instance and local variables as necessary.
2. DO NOT CHANGE THE METHOD DECLARATIONS IN ANY WAY!
3. The user guesses must be entered as a single, 3-digit string. Do not ask the user to enter three separate strings.

Instructions:

- Create a project called **`PicoFermiBagel`** that has `Bagels.swift` and `main.swift`.
- Be sure to document your code (add comments on top of your swift and main files).
- In the comments add your name, date, course, homework number, and statement of problem.

Submission

- To prep your project for submission, you need to zip up your entire project directory. To do this right click (or control click) on the project folder **PicoFermiBagel** and select "Compress **PicoFermiBagel**".
- This will create a **PicoFermiBagel.zip** file that contains your project.
- Simply upload **PicoFermiBagel.zip** through Lab02 link on Blackboard.