CS 1410: Word Zap!

Game rules: each player begins the game with seven letters. Players take turns eliminating their letters by forming words that use their letters (but only their letters), and the first player to eliminate all of their letters wins. If a player cannot form a valid word using their letters, they may choose to pass their turn and receive an additional letter. Letters are selected randomly by the game.

Assignment

Your assignment is to create a Python program that uses a class to play the game described above for two or more players.

Your program will start by asking how many players will play the game, and then ask for the name of each of the players. Seven random letters should be assigned to each player, and then gameplay should begin.

For each player, in turn, your program will print the player’s name and their letters, and then ask the player for a word. Your program should check to ensure that the word entered by the player consists only of the player’s letters (and also ensure that the player does not use a letter more times than that letter occurs in the player’s letters).

If the word contains a letter that the player does not have, then this word should be discarded without any effect to the game, and the player should be asked for another word. Otherwise, all letters used by the word should be eliminated from the player’s letters.

If the player does not enter a word, then their turn should be considered a pass, and an additional random letter should be added to the player’s letters.

After the last player completes their turn, gameplay should continue with the first player and proceed until a player eliminates all of their letters. At this point, the name of the winner should be printed and your program should terminate.

Extra Challenges

- Add a bit of friendliness to your program! When your program first starts, display a brief message explaining the rules of the game.

- Make your user input case-insensitive, so that the letter ‘A’ is considered equal to the letter ‘a’.

- When the player enters a word, also check to make sure that the word is valid according to an English dictionary.

Hints

- The class that you create should represent a player and its data (name, letters, etc.), and should include methods to add a random letter to a player’s letters, verify a word against a player’s letters, eliminate a player’s letters used by a word, etc.

Sample

Program execution:

```
Welcome! Time to play! Try to use all of your letters. The first player that uses all of their letters wins!

How many players will be playing? 2
Enter the name for player #1: Luke
Enter the name for player #2: Leia

Great! Now we can play!

Luke, it is your turn!
Your letters are: e r c q b m j
Enter a word to play (or press enter to pass):
```
You get another letter!

Leia, it is your turn!
Your letters are: m k v c m p g
Enter a word to play (or press enter to pass): move
Check your letters and try again!

Leia, it is your turn!
Your letters are: m k v c m p g
Enter a word to play (or press enter to pass): mpg
Great job!

Okay! Next round!

Luke, it is your turn!
Your letters are: e r c q b m j o
Enter a word to play (or press enter to pass): comb
Great job!

Leia, it is your turn!
Your letters are: k v c m
Enter a word to play (or press enter to pass): mvc
Great job!

Okay! Next round!

Luke, it is your turn!
Your letters are: e r q j
Enter a word to play (or press enter to pass): jeer
Check your letters and try again!

Luke, it is your turn!
Your letters are: e r q j
Enter a word to play (or press enter to pass): jerq
Great job!

Luke wins!!