Celebrities

Introduction

(From Wikipedia…) Celebrity (also known as Celebrities, The Hat Game, Lunchbox, Salad Bowl or The Naked Game) is a party game similar to Charades, where teams play against each other to guess as many celebrity names as possible before time runs out.
This first assignment will give you a non-trivial introduction to C++. You'll be required to integrate several concepts we've discussed already: string manipulation, reading from files, creating classes, using vectors, and randomness.

Version 0.0 = Get the timing working

We'll do this together in class. The top-design pattern of this game is that a player is given a clue and then guesses a celebrity game. This pattern continues until one minute expires. We haven't yet discussed how to do a timer in C++, so we'll do this together.

Version 0.1 = Write the Celebrity class

You'll need to write a class called `Celebrity`. Recall that this will be split between two files: `Celebrity.h` and `Celebrity.cpp`. This class should have two member functions: `getName` and `getClue`. The constructor should take two parameters: the celebrity's name and a string of clues. The string of clues might contain multiple clues separated by commas, like `"gives presents to kids,rides a sled pulled by reindeer,lives at the North Pole,runs a workshop of elves making toys"`. You'll want to store these individual clues in a `vector`. The `getClue` member function should return a randomly chosen clue and then remove that clue. If a there are no more clues (probably because the player can't guess the celebrity), this member function should return "no more clues".

Version 0.2 = Make a "one celebrity" game

Now that you have a working `Celebrity` class, you can play a simple version of the game involving just one celebrity. Modify your `main.cpp` to create a celebrity and then inside the loop, provide a clue, and ask the user to guess a name. You'll want to provide feedback on whether the guess is correct or not. (This game won't be very fun, but it is helpful to get this working before you move on to later versions.)

Version 0.3 = Manually make a vector with 3 or 4 celebrities

Now, modify your `main.cpp` to create three or four celebrities, and manually add them to a `vector` of celebrities. Then, modify the game to pick a random celebrity from the `vector`. If the player's guess is correct, the player earns a point and that celebrity is removed from the vector and a new celebrity is picked. If the player is incorrect, the game stays on the current celebrity, so the next clue will be for the same celebrity.

Version 0.4 = Load the celebrity file

The starter project you've downloaded contains a file called "celebrities.txt". If you open up this file, you'll see that it is formatted like this.

```
Oprah Winfrey:famous talk show host,gave everyone a car
```

This particular celebrity has two clues, but there could be many more, separated by commas. In this task, you should use an ifstream to load the entire list of celebrities into your vector. It will feel very similar to what we did loading the rooms.txt file in the CampbellRooms example, although you'll have to split up the line using the string methods.

Possible extensions

Like many of the larger assignments in this class, there are some possible extensions. Here are a few I thought of:

- The current version of the game doesn't let you "pass" if you don't know the answer. You could somehow incorporate that into the game.
- You could certainly develop a better user interface, perhaps even learning about graphics or how to develop a GUI in C++.
- When you actually play "Celebrities", there are using two teams competing against each other. You could somehow incorporate that into the game.