Challenge Twitterbots, topic 3
Foundations of Interaction Technology

Challenge:
This week's challenge is about *designing* and *implementing* a Twitterbot. The bot should be interactive and interesting; it does not need to be useful. Your bot should implement a persona, and you have to describe in a report the persona, how it is implemented in the bot, and how the general interaction with users works.
You will be working in groups of 4. On Friday afternoon (13:45-15:30) there is a "lab session" for groups that struggle with implementation details.

Prototype:

You should implement an *interactive* Twitter bot. The bot should either

- tweet something first, and interact with users who reply to the initial tweet (e.g. @bot tweets "Knock knock", someone replies "who's there?", and @bot replies something else)
- react to user-initiated tweets (e.g. respond when a tweet mentions the @bot. For example, if a user says "@bot knock knock", the bot might answer "who's there?"), and continue from there.

The bot should be based on a *persona*, that is, it should show a consistent "personality" (for example like https://twitter.com/bottledbrian that behaves like a kid, or https://twitter.com/RedScareBot that impersonates an American politician of the '50s). Please notice how these examples are not necessarily linked to an existing person, but more to a stereotypical character. Also note that in this challenge by a "persona" we do NOT mean a user persona as is commonly used in UX design.
The last main requirement is that the bot should be *interesting*. There is no clear-cut definition for this; it means that you should aim at something better than https://twitter.com/bottledbrian, https://twitter.com/neniambulance or https://filiph.github.io/markov/ - these bots are interactive in a minimal sense, if at all: they don't really communicate with the user, and they don't really need to understand anything of what you tweet to them. They are more like "deaf content-producers", which is what makes them uninteresting.

Your bot does not need to be always active (e.g., if you are hosting it on your own computer instead of a dedicated server), but *make sure it is running the day you are presenting it* (Tuesday 8 October, 13:45-15:30). In the challenge page on Canvas there is a link to a Python sample script that you can use as a starting point for your bot (notice that the script has no API rate limiting, if you tweet too often you can get temporarily blocked!).

If you cannot produce a working implementation of the bot, as a fallback strategy you can do a very precise specification (e.g. writing in "pseudocode" instead of Python); during presentation day, however, you *will have to act out the specification* (you'll have to impersonate the bot - not on Twitter, in the classroom! - and you'll have to answer the messages of your users as the bot would do).

Report: (length max. 2 pages A4, 11pt font size)

1. Describe your bot, what makes it interesting, its persona, the behaviors that the user can see and how they are implemented. (3 points)
2. Explain what challenges you encountered and how you tried to overcome them (2 pt)
3. How did you make sure that the bot is well-behaved? (2 pt)
4. Discuss how you could further improve your bot. What challenges would those new features pose? (1 pt)
5. The final 2 points will be given based on the quality of your interactive Twitterbot. How interactive is it, how interesting (according to us), how clearly recognizable is the persona, ... ? (2pt)

Make sure the report contains the @botusername or Twitter url (http://twitter.com/@botusername).