CS209 Project - Spring 2018

You are asked in this project to create with JavaFX a subset of an "in-flight entertainment system" - those media systems that allow passengers to watch films and TV programs, listen to music, play games and check on a map the current location of the aircraft.

You will only work on the film part. Additionally, the videos to play will be trailers instead of full-featured films. You will find on Sakai a number of trailers you can use (you are free to use anything else, as long as you have enough films to please a large array of tastes). You should demo your system with at least 20 trailers.

Perhaps you have already used such a system; if not, you have on the small screen on the back of the seat before you a list of films in different categories. These categories are sometimes based on the genre of the films, sometimes based on the region of origin, sometimes based on the age of the film, e.g. "new releases" against "old classics", sometimes based on something completely different, such as "Our selection". A film can often be found in several categories. You can navigate the list, get some detail (director, main actors, year, plot summary, language, subtitles), and usually see either a trailer or launch the film (in the context of this project, the only choice will be to run the film, replaced by the trailer). Note that a film can sometimes be dubbed in two or three languages.

You are encouraged to search the web to get an idea about existing systems.

The Context

You'll assume that you are working for a software company that creates such systems for airlines. Airlines are your customers.

Requirements:

User Interface

- **KEYBOARD ONLY and MOUSE ONLY interfaces**

  Some airlines have touch-screens, others are using simple remote-control commands. That means that your system must be navigated either by clicking on buttons (NOTHING MORE SOPHISTICATED THAN A BUTTON), for the touch screen interface, or by using just a few keys on your keyboard, which will represent the remote control buttons: up, down, left and right arrows, enter, and + and - to adjust sound volume. You are allowed also * if you have any special need.

- **SIMPLE**

  Airlines don't provide a 100-page user manual to every passenger. The system must be intuitive, and usable by passengers of any age, origin, level of education and technical ability.

- **MULTILINGUAL**

  The system must be multilingual by construct. It means that all messages should be read from external files

Customer Requirements

What you'll ship to tour customer (the airline) will be a .jar that contains your application.

- **LOOKS MUST BE EASY TO CHANGE THROUGH EXTERNAL FILES**

  The customer must be able to customize easily the application so that the application looks like it has been custom-written for them. They must be able to add their logo and set colors to their corporate colors. This will be performed by the airline IT personnel, there is no need for a user-friendly interface (they can edit a configuration file, drop a file into a folder or rename their logo file to logo.png...), but it has to be possible.

- **EASY TO TRANSLATE**

  Airlines must be able to translate the interface easily, so as to have the national language of the airline, English, and some optional languages on some routes (for instance the language in the country of departure and in the country of arrival, as they can expect passengers from both countries). Hint: think of Properties.

- C**ONTENT MUST BE EASY TO CHANGE BY THE AIRLINE**

  The choice of films and the periodical update of the list of films (you have to change at least some films every month) will be the responsibility of the communication people at the airline. They must be able:

  - To store the videos files into a pre-defined folder
  - To enter ALL the information about films (director, main actors, duration, language, year, topic). As you may not have much time to create a decent interface, a reasonable option is to assume that they enter the information into a spreadsheet that is saved as a CSV file (comma separated values) that is stored in the system.
  - You will write some small utility that will retrieve film information automatically from Wikipedia.