



IS 2013 - Assignment 1

The goal is to implement a simple version of an interface to manage the behavior of a smart home. This interface will be extended in the second assignment.

Requirements

1. You must work **individually**.
2. We suggest implementing the interface in **Java**, using **Swing** or **SwingStates**. You can use another toolkit or language, but you should ask the instructors first and keep in mind that you will work in pairs for the second assignment.

About smart homes

"A smart home is [a home] equipped with lighting, heating, and electronic devices that can be controlled remotely by smartphone or computer: you can contact your smart home on the Internet to make sure the dinner is cooked, the central heating is on, the curtains are drawn, and a gas fire is roaring in the grate when you get home. "

-- Oxford dictionary

To control a smart home you need an interface to program its behavior, running on a mobile, the web or on a panel at home.

Proposed topic

Our assumption is that the control interface will run on a panel inside your smart home. As it is difficult to build a complete system in one go, here are the recommended functionalities for this assignment:

1. A **graphical representation** of the house and its rooms.
2. The ability to program in detail **at least 2 aspects** of the house (e.g. temperature and light). Users should be able to do this for a specific room or for the entire house.
3. At least one **drag-and-drop interaction** (it is up to you what/where).
4. **Undo/redo functionality** for all actions (your "history" can be limited to the last 20 actions)

You can use any interactions you can think of for doing the above actions (the only requirement is the existence of one drag-and-drop interaction). Everything else is up to you. We are interested in how everything fits together in your system, and how creative you are in your interactions.

To get a very good mark we expect you to go **beyond** the above recommended functionalities, with at least 3 extensions: they will be judged by their complexity **and** how they fit overall with your prototype).

Possible extension examples include: more parameters to control, editing the graphical representation, adding note-taking, complex programmable behavior (e.g. different rules based on conditions outside the house), saving your parameters, etc.

Advice

- Define the **limits** of your project before starting. Developing more functionalities will not necessarily give you a better mark (or a better product).
- What is important is the **realization** of the techniques, that need to be **well thought out, easy to use, and working well together** in the interface.
- Your implementation should be **usable**. You should prevent users from reaching unexpected states in the system and provide feedback and feedforward mechanisms to avoid user errors.
- You are encouraged to be **creative**, but make sure what you decide to do is inside your competence level and abilities. If some aspects seem too hard, try to simplify things.
- Try to **structure well** and **comment** your code (e.g. with Javadoc for Java).

Submission 1-A (due latest 18 Oct)

Create a **storyboard** of the tool you plan to create as one or more images:

- Write first a **scenario** that illustrates the usage of your system (10 lines of text max).
- Then show this scenario on a **storyboard**. Show how to use the techniques you plan to implement (even if you never implemented them). Make sure the storyboard is done **before implementation**.
- The storyboard can be done by sketching on paper, with pictures of paper prototypes, on power point, etc.

Examples of interaction storyboards as sketches can be found here:

<http://msdn.microsoft.com/en-us/library/windows/desktop/ff800706.aspx>

If you want to learn more about storyboards than what we cover in class you can have a look here:

<http://grouplab.cpsc.ucalgary.ca/grouplab/uploads/Publications/Publications/2012-NarrativeStoryboard.Interactions.pdf>

Give your storyboard to your tutorial instructor in class, or scan it and email it **before class**. Make sure your full name is clearly visible on the storyboard.

Submission 1-B (due latest 8 Nov)

Create an archive (zip or tar.gz) named *first-last-name.zip* or *first-last-name.tar.gz*

It should include:

1. A text file with a **brief description of your result**: what functionalities you implemented and what simplifications you made (bullet points are ok).
2. Your **code**: both the *source code* and an *executable version* (in java a .jar), or your eclipse project. Add any necessary instructions for running your project.

Send your compressed archive by **Friday 8 November @ 23:00 the latest** to your tutorial instructor, David (David.Bonnet@lri.fr) or Cedric (Cedric.Fleury@lri.fr) and put Anastasia in cc (Anastasia.Bezerianos@lri.fr).

Use the title "[IS] Assignment 1" in your email.