



IS 2013 - Assignment 2

Creating the perfect interface is impossible. We often find through user testing problems with our interface. Other times our users request extra features we did not anticipate. This assignment is about redesign and extensions of Smart Home 1.

Requirements

1. You can now work in **pairs**.
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.
3. Pick **one** of the two topics. If you want to do a bit of both we will mark you 50% on each.

Proposed topics (select 1)

i. Redesign

Our assumption here is that during user testing you found problems within your control interface. Thus you decided to redesign it with the help of another colleague. You are requested to provide an interface with exactly the same 4 functionality points + extensions from assignment 1, but new interactions.

Note: In this option we request that you are as creative as possible and provide a different interface from your assignment(s) 1. For example moving a menu from left to right or just combining interactions from your previous assignments is not enough of a redesign. But changing from menu to direct manipulation or crossing is.

If in doubt, talk to Anastasia and your TAs

ii. Extend

Our assumption here is that after deployment in the market, the users of your system requested additional functionality from your interface. Given the urgency to get the new features out to your users, you decided to work with another colleague on the extensions. You are requested to provide an interface with exactly the same 4 functionality points from assignment 1 + 3 extensions. AND the following additions:

1. A way to **add** new rooms into your visual representation of the house.
2. A way to **edit two** visual aspects of rooms (e.g. position, color).
3. Option to **Save/Reload** the user's parameter values and house/room layout.
4. A way to program **multiple houses** together (a building block).

You can use any interactions you can think of for doing the above actions (the only requirements 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 recommended functionalities. If your 3 extensions from assignment 1 are part of the requirements now, you need to add new ones.

Possible extensions include: more visual aspects to edit (e.g. size, orientation), allowing users to add furniture images, adding note-taking, complex programmable behavior (e.g. different rules based on conditions outside the house), etc.

Note: In this option we will not you based on how complex your proposed interactions are **and** how they fit overall with your prototype.

iii. Both (50-50%)

In real life, you may also need to do both redesign and extensions of your interface. If you chose this option, you need to provide an interface with exactly the same 4 functionality points from assignment 1 + 3 extensions. AND redesign 2 of them and provide 2 new extensions (from option ii).

Advice (same)

- 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 to hard, try to simplify things.
- Try to **structure well** and **comment** your code (e.g. with Javadoc for Java).
-

Submission 2-A (due 15 Nov)

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.

Give your storyboard to Anastasia in class, or scan it and email it **before class**. Make sure both your full names are clearly visible on the storyboard.

Submission 2-B (due THURSDAY 12 Dec)

Create an archive (zip or tar.gz) named *lastname1-lastname2.zip* or *lastname1-lastname2.tar.gz*

It should include:

1. A text file containing
 - a. which **option** you chose (or 50-50%).
 - b. **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 **THURSDAY 12 December @ 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 2" in your email.

Project Demo 2-C (13 Dec)

Give a **demo** to Anastasia and your TA. Practice up-front what you will say at the demo. You will have **10 min**.

It is up to you how you present your demo, you can include slides or just talk, BUT

1. **Show us your project running** and the interactions being used
2. Stay within your 10 min limit
3. All members of your team must talk