

Heuristic Evaluation of Robotic Arm

The following is a retrospective analysis of the robotic arm, using the heuristics from Nielsen (1994).

Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

The system interface does not give feedback on the options selected, but simply moves to the next stage of selection. The only feedback comes when the arm actually starts to move, and then the user can see whether it is moving as desired or not. *This corresponds to issue 11 in the existing set of issues: "Lack of feedback about selection"*.

Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

The commands are built up in a logical order ("move *joint* in *direction* with *speed*, now *go*"). The names of the individual joints may not be immediately apparent to a novice user, and will have to be learnt (*issue 33*). There might be some initial confusion between "move" and "movearm" (*issue 3*). The names of the directions and speeds are easy to understand.

User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

In the original design, it was not possible to undo commands once issued (*issue 2*). This was corrected for the later version. It is always possible to return to the main menu by using the "end" option.

Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

In the original design, there was a possible confusion between "continue" and "go" (*issue 5*). This was corrected in a redesign. The rest of the design is consistent.

Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

The interface design is simple, so there should be minimal errors when making selections with a mouse. Gesture and voice input may be more error-prone (*issues 27, 28*).

Recognition rather than recall

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

The system presents all options clearly. The only challenge may be matching the names of the joints to the physical joints being controlled (*issue 33 again*).

Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Due to the nature of the device, users cannot tailor frequent actions themselves, but it is possible for a carer to program pre-taught positions to speed up interactions.

Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

The design conforms to this heuristic.

Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

There are no possible errors in this sense: it's simply that users might have chosen a different option from that intended. Users can only recover from those errors if they either spot the error immediately (as the system moves on to the next option), which is easily addressed through 'undo', or when they see the arm moving in the wrong direction. In the latter case, it takes a long sequence of actions to correct the direction (*new issue, related to but different from 32*).

Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

There is no help facility. It is assumed that users will be individually trained to use the system.