Control

The level of control provided by a system should be related to the proficiency and experience levels of the people using the system.

People should be able to exercise control over what a system does, but the level of control should be related to their proficiency and experience using the system. Beginners do best with a reduced amount of control, while experts do best with greater control. A simple example is when children learn to ride a bicycle. Initially, training wheels are helpful in reducing the difficulty of riding by reducing the level of control (e.g., eliminating the need to balance while riding). This allows the child to safely develop basic riding skills with minimal risk of accident or injury. Once the basic skills are mastered, the training wheels get in the way, and hinder performance. As expertise increases, so too does the need for greater control.[1]

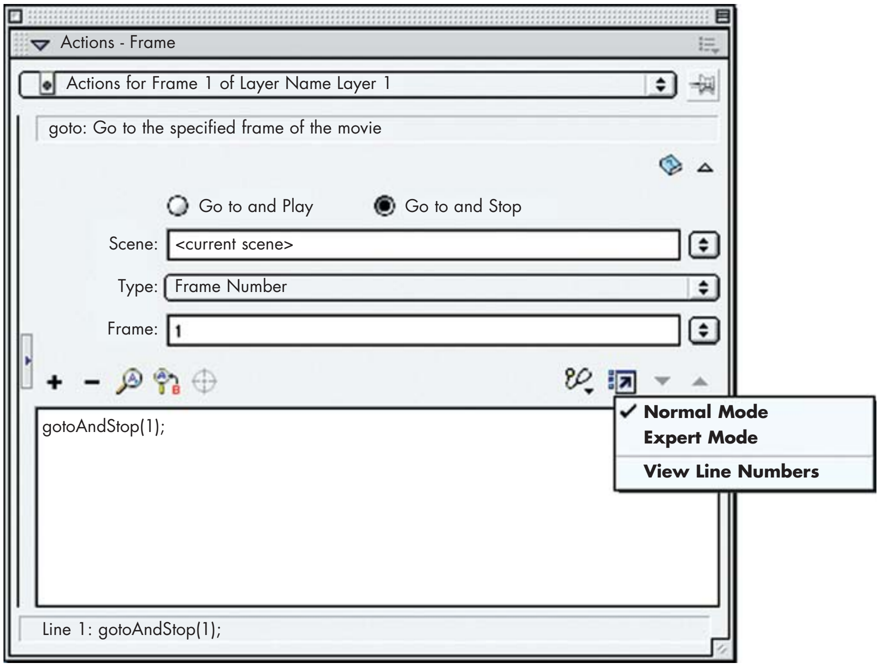
A system can accommodate these varying needs by offering multiple ways to perform a task. For example, novice users of word processors typically save their documents by accessing the *File* menu and selecting *Save*, whereas more proficient users typically save their documents using a keyboard shortcut. Both methods achieve the same outcome, but one favors simplicity and structure, while the other favors efficiency and flexibility. This tradeoff is standard when allocating system control. Beginners benefit from structured interactions with minimal choices, typically supported by prompts, constraints, and ready access to help. Experts benefit from less structured interactions that provide more direct access to functions, bypassing the support devices of beginners. Since accommodating multiple methods increases the complexity of the system, the number of methods for any given task should be limited to two—one for *beginners*, and one for *experts*.

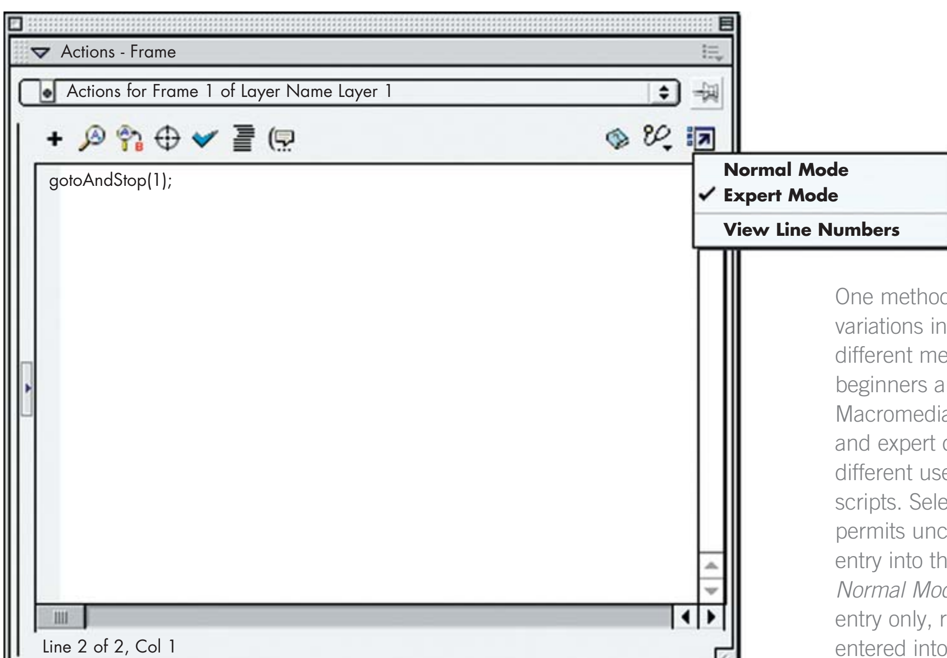
The need to provide expert shortcuts is limited to systems that are used frequently enough for people to develop expertise. For example, the design of museum kiosks and ATMs should assume that all users are first-time users, and not try to accommodate varying levels of expertise. When systems are used frequently enough for people to develop expertise, it is often useful to provide simple ways to customize the system design. This represents the highest level of control a design can provide. It enables the appearance and configuration of a system to be aligned with personal preferences and level of expertise, and enables the efficiency of use to be fine-tuned according to individual needs over time.

Consider the allocation of control in the design of complex systems. When possible, use a method that is equally simple and efficient for beginners and experts. Otherwise, provide methods specialized for beginners and experts. Conceal expert methods to the extent possible to minimize complexity for beginners. When systems are complex and frequently used, consider designs that can be customized to conform to individual preference and levels of expertise.

See also Constraint, Flexibility-Usability Tradeoff, and Hierarchy of Needs.

[1] See, for example, *The Psychology of Human-Computer Interaction* by Stuart K. Card, Thomas P. Moran, and Allen Newell, Lawrence Erlbaum Associates, 1983; and *The Humane Interface: New Directions for Designing Interactive Systems* by Jef Raskin, Addison-Wesley 2000.





One method of accommodating variations in proficiency is to provide different methods of interaction for beginners and experts. For example, Macromedia Flash supports novice and expert developers by providing different user modes when writing scripts. Selecting the *Expert Mode* permits unconstrained command entry into the editor field. Selecting *Normal Mode* permits constrained entry only, requiring commands to be entered into specialized fields so that they can be immediately checked for correctness.