Chapter 2

Design

This section takes a look at the cross-cutting topic of design, from different perspectives. It is based on a number of sources, principally Don Norman's classic "The Design of Everyday Things" [Nor02] and Terry Winograd's "Bringing Design to Software" [Win96]. The section on game design takes from [FS04] and Rouse's "Game Design - Theory and Practice" [Rou01].

Handouts printout of this chapter

2.1 What is design?

([Win96] Introduction)

Whenever objects are created for people to use, design is pervasive.

Problems attributed to "human errors" can most often be traced to bad design.

Winograd observes that design is really more an activity than a thing (although "design" is used as a noun), and lists the following characteristics:

- Design is conscious
- Design keeps human concerns in the center (Extends to human-like performance)
- Design is a conversation with materials
- Design is creative (criteria are underconstrained)
- Design is communication (design languages)
- Design has social consequences
- Design is a social activity

Design entails:

- Intention
- Purpose
- Target (e.g. design **for** a user).

2.1.1 Software design

definition from the Association for Software Design:

> Software design sits at the crossroads of all the computer disciplines: hardware and software engineering, programming, human factors research, ergonomics. It is the study of the intersection of human, machine, and the various interfaces–physical, sensory, psychological–that connect them.

Winograd defines software as a medium for the creation of *virtualities*–the world in which a user of this software perceives, acts and responds to experiences.

- Software Engineering: the discipline concerned with the construction of software that is efficient, reliable, robust and easy to maintain.
- Interface design: the possibilities for software are both created and constrained by the physical interfaces.
- Human-computer interaction: whenever someone designs software that interacts with people, the effects of the design extend beyond the software itself to include the experiences that people will have in encountering and using that software.
- Art: because computing evolved initially for use in the laboratory and the office, noncognitive aspects have been largely ignored, except by creators of computer games. Yet, whenever people experience a piece of software, they have natural human responses. They experience beauty, satisfaction and fun, or the corresponding opposites.

Game design: gameplay, interaction/interface design, assets... Target: ?

Architecture design: conceptual (components, organization), logical (components, interactions, algorithms, data structures), physical (physical components, communications). Target: ?

Code design: classes, functions, etc. (not yet actual code...). Target: ?

2.1.2 Principles of Design for Usability and Understandability

- Provide a good conceptual model: make it easy to determine what actions are possible at any moment (make use of constraints)
- Visibility: Make things visible, including the conceptual model of the system, the alternative actions, and the result of actions.
- Mapping: Follow natural mappings between intentions and the required actions; between actions and the resulting effect; and between the information that is visible and the interpretation of the system state.
- Feedback: make it easy to evaluate the current state of the system, give each action an immediate and obvious effect

2.2 Designing for People

Elements of psychology...

From [Nor02] p, 115:

Many scientists who study artificial intelligence use the mathematics of formal logics–the predicate calculus–as their major tool to simulate thought.
But human thought–and its close relatives, problem solving and planning–seem more rooted in past experience than in logical deduction. Mental life is not neat and orderly. It does not proceed smoothly and gracefully in neat, logical form. Instead it hops, skips and jumps its way from idea to idea, tying together things that have no business being put together; forming new creative leaps, new insights and concepts. Human thought is not like logic; it is fundamentally different in kind and in spirit. The difference is neither worse nor better. But it is the difference that leads to creative discovery and to great robustness of behavior.

2.2.1 The seven stages of action

([Nor02] fig. 2.6 p. 47)
action = goal (1) + **evaluation** (3) + **execution** (3)

- Forming the goal
- Forming the intention
- Specifying an action
- Executing the action
- Perceiving the state of the world
- Interpreting the state of the world
- Evaluating the outcome

Corresponding design challenges:

- **Gulf of Execution**: Difference between intentions and allowable actions (affordances). Example: movie projector vs. VCR (cartridge hides threading complexity).
- **Gulf of Evaluation**: Amount of effort that the person must exert to interpret the physical state of the system and to determine how well the expectations and intentions have been met. Example: movie projector vs. VCR (trap door hides state).

The seven stages as design aid ([Nor02] fig. 2.7, p. 53)

- *Visibility*: by looking, the user can tell the state of the device and the alternatives for action.
- *Good conceptual model*: consistency in the presentation and results; coherent system image.
- *Good mappings*: It is possible to determine the relationships between actions and results, between the controls and their effects, and between the system state and what is visible.
- *Feedback*: The user receives full and continuous feedback about the results of actions.

2.2.2 Knowledge in the Head (memory) and in the World

([Nor02] p. 54)
Precise behavior can emerge from imprecise knowledge (ex: coins):

- *Information is in the world*
- *Great precision is not required*
- *Natural constraints are present*
- *Cultural constraints are present*

Behavior is determined by the combination of internal knowledge (in the head) and external information and constraints.

Always trade-off between knowledge in the world and in the head ([Nor02] p.79).

Memory as knowledge in the head

Three important categories of memorization:

- Arbitrary things
- Meaningful relationships
- Memory through explanation

Limits on how much a person can hold in memory at one time, and on how many active thoughts can be pursued at once.

Short Term Memory (STM): a person should not be required to remember more than 5 or so unrelated items at one time.

Long Term Memory (LTM): information is better and more easily acquired if it makes sense, if it can be integrated into some conceptual framework. Retrieval from LTM is apt to be slow and contain errors.

In "Magical Number Seven (plus or minus two)" [Mil56], cognitive psychologist George A. Miller showed a number of remarkable coincidences between the channel capacity of a number of human cognitive and perceptual tasks. In each case, the effective channel capacity is equivalent to between 5 and 9 equally-weighted error-less choices: on average, about 2.5 bits of information. Miller hypothesized that these may all be due to some common but unknown underlying mechanism.

Memory as knowledge in the world

Very useful but requires physical situation (out of sight, out of mind)

- Reminding (ex: calendars, knotted handkerchief, etc.)
- Natural mappings (ex: stove controls layout)

2.2.3 Designing for error

To err is human! Always design with that in mind...

- Understand the causes of error and design to minimize those causes.
- Make it possible to reverse actions (undo) or make it harder to do what cannot be reversed.
- Make it easier to discover the errors that do occur, and make them easier to correct.
- Change the attitude towards errors: Don't think of the user as making errors, think of the actions as approximations of what is desired.

2.2.4 The design challenge

Design is an evolutionary, dynamic, iterative process.

Design must balance aesthetics, usability, cost, schedule, client requirements, etc.

Some reasons why designs might fail:

- Putting aesthetics first
- Designers are not typical users
- The designer's clients may not be users

2.3 Seven Principles for Transforming Difficult Tasks in Simple Ones

([Nor02] p. 188)

2.3.1 Use both knowledge in the world and knowledge in the head

3 conceptual models:

- Design model: conceptualization that the designer has in mind.
- User's model: what the user develops to explain the operations of the system.
- System image: its physical appearance, its operation, the way it responds, and the manual and instructions that accompany it.

Ideally, the user's model and the design model are equivalent. However, the user and designer communicate only through the system itself. Thus the system image is critical: the designer must ensure that everything about the product is consistent with and exemplifies the operation of the proper conceptual model.

Note that the system image includes instruction manuals and documentation. Two problems: (1) often documentation is not developed with necessary care and concern; (2) human nature is such that most people do not read the documentation.

2.3.2 Simplify the structure of tasks

Take into account psychology, STM and LTM limitations.

- Provide mental aids (keep task the same)
- Use technology to make visible what would otherwise be invisible, thus providing feedback and the ability to keep control.
- Automate (but keep task the same). Don't take away control!
- Change the nature of the task. Example: Hook-and-Loop fastener

2.3.3 Make things visible: bridge the gulfs of execution and evaluation

Make things visible on the execution side of an action so people know what is possible and how actions should be done; make things visible on the evaluation side so that people can tell the effects of their actions.

The system should provide actions that match intentions. It should provide indications of system state that are readily peceivable and interpretable and that match intentions and expectations.

The system state should be visible and readily interpretable.

Make the outcome of an action obvious.

2.3.4 Get the mappings right

Exploit natural mappings. Make sure the user can determine the relationships:

- Between intentions and possible actions
- Between actions and their effect on the system
- Between actual system state and what is perceivable by sight, sound or feel
- Between the perceived system state and the needs, intentions, and expectations of the user

2.3.5 Exploit the power of constraints, both natural and artificial

Use constraints so that the user feels there is only one possible thing to do (the right thing!).

2.3.6 Design for errors

Assume any "error" that can be made will be made.

2.3.7 When all else fails, standardize

Sometimes an arbitrary choice is necessary.

Example: aspects of car driving had to be standardized (side of the road to drive on, side of the car the driver sits on, positions of essential components such as steering wheel, clutch, brakes, accelerator, etc.)

2.4 Game design

([Rou01] p.xxi)

The Game Design is what determines the gameplay. The game design determines what choices the players will be able to make in the game-world and what ramifications those choices will have on the rest of the game. The game design determines what win or loss criteria the game may include, how the user will be able to control the game, and what information the game will communicate to her, and it establishes how hard the game will be. Is short, the game design determines every detail of how the gameplay will function.

([FS04] p.13)

Game (creative) design: analogous to a production plan for a film; provides the artistic vision behind the game; implies that software will be developed.

Game software design: moves forward from the mandate that the production plan offers; reduces the features of the game to a set of software requirements and says how the requirements will be implemented.

2.4.1 Know your customer

([Rou01] ch.1)

- Players want a challenge
- Players want to socialize
- Players want a dynamic solitary experience
- Players want bragging rights
- Players want an emotional experience
- Players want to explore
- Players want to fantasize
- Players want to interact
- Players expect a consistent world
- Players expect to understand the game-world's bounds
- Players expect reasonable solutions to work
- Players expect direction
- Players expect to accomplish a task incrementally
- Players expect to be immersed
- Players expect some setbacks
- Players expect a fair chance
- Players expect not to need to repeat themselves
- Players expect not to get hopelessly stuck
- Players expect to do, not to watch
- Players do not know what they want, but they know when it is missing

2.4.2 The game design document

([Rou01] ch.19)
Sections:

- table of contents
- Introduction/overview or executive summary
- Game mechanics
- Artificial intelligence
- Game elements: characters, items and objects/mechanisms
- Story overview
- Game progression
- system menus

Inauspicious design documents ([Rou01] ch.19):

- The waffer-thin or ellipsis special document
- The back-story tome
- The overkill document
- The pie-in-the-sky document
- The fossilized document