Cognition *week 2*

Computational offloading is how external tools affect the amount of cognitive effort required to complete a task, e.g. how calculators make performing calculations easier.

There are many kinds of cognition, though user experience design mainly focuses on attention, memory, and learning. Other examples are reading, speaking and listening, problem-solving, planning, reasoning, and decision-making.

Don Norman's seven stages of action are:
1. perceiving the state of the world,
2. interpreting those perceptions,
3. evaluating those interpretations,
4. forming goals,
5. intention to act,
6. sequence of actions,
7. executing the action sequence.

Perception is out ability to make sense of the world around us and respond appropriately; it is how we sense the information around us. Sensory perception includes sight, sound, smell, taste, and touch.

Our image of the world is not given but constructed. What we can perceive is different to what we do perceive. The information that we extract depends on our motivations, our arousal, individual differences, and cultural differences. The way we perceive the world around us depends on how the brain interprets and constructs its meaning. Perception is active and constructive, not just receptive.

However, perception is not perfect; it can be fooled. Designers must help users construct the correct interpretation of the world (their system/product). We receive input and assimilate (take in and understand) it to construct our understanding, based on previous experiences.

Attention is the cognitive process of selectively concentrating on one aspect of the environment while ignoring other aspects; it is the allocation of the brain's processing resources.

Selective (a.k.a. focused) attention: focusing on one source of information while ignoring others (e.g. listening to one person speak in a crowd of others talking).

Divided attention: monitoring two or more tasks simultaneously, with attention being paid to both (e.g. talking while driving).

Sustained attention: focusing on a task over prolonged periods of time (e.g. reading a book).

We search for meaning, scanning input very quickly and looking for meaningful patterns, ignoring things that don't make sense or can't be decoded easily.

Our brains make assumptions and fill in missing details. Ambiguity causes us to 'see' different things. We also have built-in predispositions and expectations.

Designers must consider the relationship between their design and the user's attention, regarding both appropriate and inappropriate 'drawing' of attention. They must make sure that they do not bombard the user with every function at the same time, with the user's attention being drawn to appropriate functions at appropriate times. Fine grain details should only be accessible when they are needed.

Memory is how we store, manipulate, and retrieve information. Long-term memory is very large and associative; it needs time to retrieve information.

Short-term, or working, memory is where problem-solving and current processing take place. Here is where Miller's law (we can hold 7±2 things in our short-term memory), Gestalt psychology (the laws of proximity, closure, symmetry, and similarity), and chunking apply. Information is easily lost before it is transferred to long-term memory, due to disruption/interruption or distraction. Anxiety, frustration, and distraction can impede information processing, while familiarity aids processing and chunking.

Recognition is easy, while recall is difficult as recognition tasks provide memory cues that facilitate searching through memory, e.g. command lines vs GUI. When using a computer, people are already using a lot of short-term memory, so shouldn't need to think about the tool they are using.

To reduce memory load, information should be on-screen when it is needed, recognition should be used over recall, menus and paths should be shown, and screen components, menu structures, and commands should be consistent. In some cases, however, recall is more efficient than recognition. Whether it makes sense for a user to learn how to use an interface depends on the context of use (e.g. a cashier using recall on a till interface vs a customer using recognition on a self-checkout interface).

Learning is the act of acquiring new knowledge, behaviours, skills, values, preferences, or understanding. Feedback should be timely and specific – the system should be responsive – as people learn from experiences and consequences. Problems occur when feedback is not specific enough to allow us to infer cause-effect relationships, e.g. vague error messages.

It is easy to learn from structure and patterns as we need to order, categorise, and make sense of things. Learning is fastest when we can identify cause and effect, use prior knowledge to interpret, make connections, make things obvious.

Affordances are the perceived properties of an object that suggest how it can be used, e.g. pushing a button, flipping a switch, rotating a knob, labels (informative), metaphors (leveraging real-world functions), and patterns (leveraging previously learned behaviours).

When using a computer system, the user should have as little learning to do as possible. Time spent learning the system, rather than doing the task, is perceived as a waste of time. Designers should make use of existing knowledge and affordances. The system should be transparent, rather than constructive, and cause and effect should be obvious. Help should be available but well-timed. Don't obstruct the task with help messages and make it optional.