UNIT–2 : Model-Based Design and Evaluation (PYQ Solution)

PART–A Answers (≈70 Words Each)

Q1. Explain limitations of GOMS. (2022)

GOMS (Goals, Operators, Methods, Selection rules) has these limitations:

- **It models only expert, error-free performance**
- **It ignores learning, fatigue, creativity and errors**
- **It cannot model problem solving or social context**
- **Building the analysis is slow and needs task detail**

Q2. What is model-based design and evaluation? (2024)

Model-based design uses models of the user, task and system (task models, GOMS, Fitts' law) to guide design decisions before building. Model-based evaluation uses the same models to predict time, effort or errors for a design, without user testing.

Q3. State Fitts' Law. (2024)

Fitts' law: the time to move quickly to a target increases with distance and decreases with target width:

$$MT = a + b \log_2 \left(\frac{D}{W} + 1 \right)$$

where D is distance, W is width, and a, b are device constants.

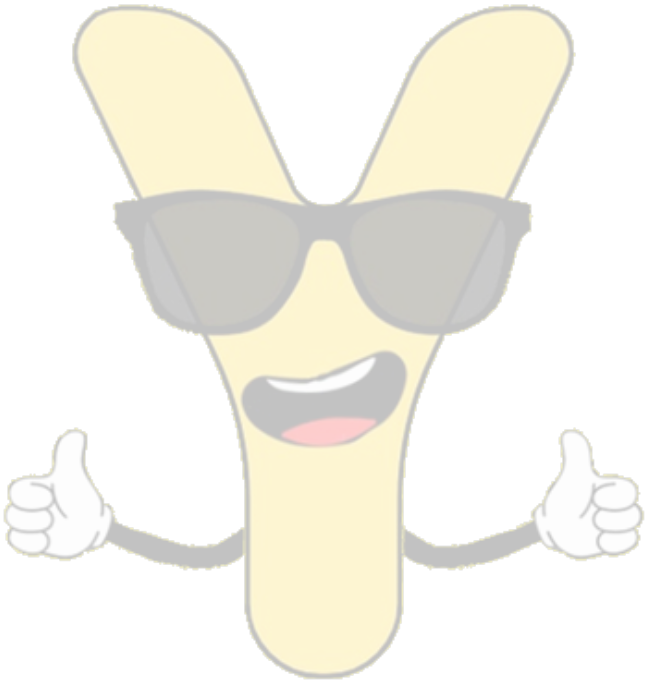
Q4. How can Fitts' Law be used to predict performance? How to measure

difficulty of a target acquisition task? (Feb 2024)

The law gives the movement time for a pointing task from D and W, so designs can be compared (larger, closer buttons are faster). Difficulty is measured by the index of difficulty

$$ID = \log_2\left(\frac{D}{W} + 1\right) \text{ bits}$$

For $D = 160, W = 20$: $ID = \log_2 9 = 3.17$ bits. With $a = 0.1, b = 0.15$: $MT = 0.58$ s.



PART–B Answers (≈250 Words Each)

Q1. What is CMN-GOMS, and how does it differ from KLM? / Discuss the key differences between KLM and CMN GOMS. (2024, Feb 2024)

CMN-GOMS (Card, Moran, Newell) is the original GOMS model. It describes a task as a **hierarchy of goals and sub-goals**, **operators** at the lowest level, **methods** to reach goals and **selection rules** (IF-THEN) to choose among methods. **KLM** (Keystroke-Level Model) is the simplest GOMS version: a flat list of keystroke-level operators (K, B, P, H, M) whose times are added.

Point	KLM	CMN-GOMS
Level	Keystroke level	Goal hierarchy
Structure	Flat sequence of operators	Goals, operators, methods, selection rules
Output	Execution time only	Time, structure and method choice
Selection rules	None	Explicit IF-THEN rules
Effort	Quick	Slower, needs full task breakdown
Use	Compare similar actions	Analyse a complete task with alternatives

---

Q2. Discuss in brief the utility of models in HCI. (Feb 2024)

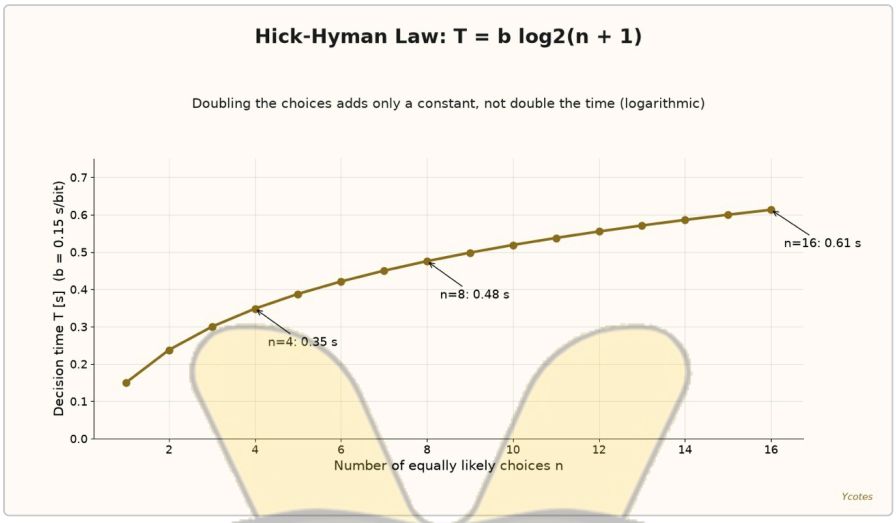
A model is a simplified representation that allows prediction without building the real system. Utility in HCI:

- **Predict** performance (time, errors) early and cheaply, for example with KLM and Fitts' law
- **Compare** design alternatives objectively
- **Explain** why an interface is slow or hard to learn
- **Communicate** and document design knowledge
- **Reduce** the number of user tests required

Models need simplifying assumptions and mostly describe expert, error-free behaviour, so they supplement, not replace, user testing.

Q3. Explain the Hick Hyman Law. Describe the predictive formulation of the law. (Feb 2024)

Hick-Hyman law: the time to choose among alternatives rises with the **information content** of the decision, and so increases **logarithmically** with the number of choices.



Predictive formulation:

- Equally likely choices: $T = b \log_2(n+1)$
- Unequal probabilities: $T = b\,H$, with $H = \sum_i p_i \log_2\left(\frac{1}{p_i} + 1\right)$

where $b \approx 0.15$ s/bit.

Worked example:

- $n = 4$: $T = 0.15 \log_2 5 = 0.35$ s
- $n = 8$: $T = 0.15 \log_2 9 = 0.48$ s
- $n = 16$: $T = 0.15 \log_2 17 = 0.61$ s

Doubling the choices adds only about 0.13 s. For probabilities 0.5, 0.25, 0.125, 0.125, $H = 2.165$ bits and $T = 0.32$ s.

Application: broad, shallow menus (16 items in one menu 0.61 s against two levels of 4 items 0.70 s), ordering items by frequency.

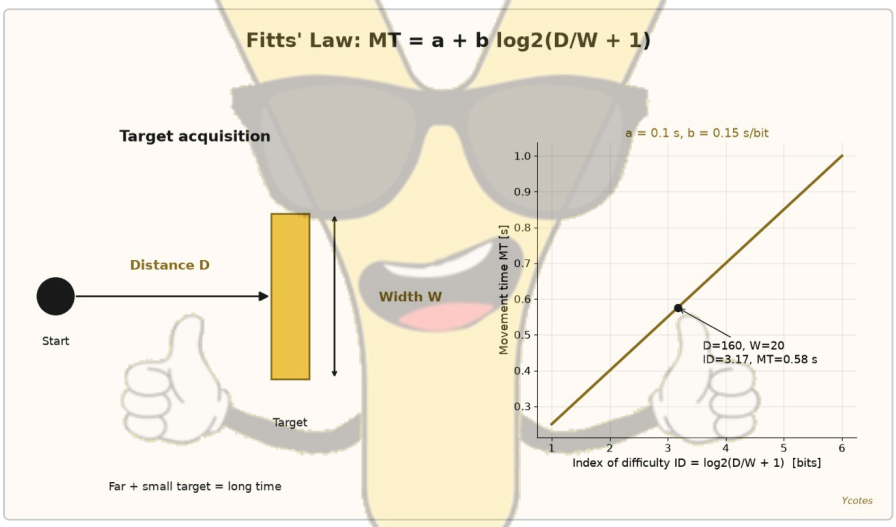
PART–C Answers (≈500 Words Each)

Q1. State Fitts' Law and explain its significance in model-based design and evaluation. Describe Hick-Hyman's Law and its application in model-based design and evaluation. (2024)

Fitts' law

$$MT = a + b \log_2 \left(\frac{D}{W} + 1 \right)$$

MT movement time, D distance to the target centre, W target width, a, b empirical constants. The term $\log_2(D/W + 1)$ is the **index of difficulty** (bits); $IP = ID/MT$ is the **index of performance** (bits/s).



Worked example ($a = 0.1, b = 0.15$):

Target	ID (bits)	MT (s)
$D = 160, W = 20$	3.17	0.58
$D = 160, W = 40$	2.32	0.45

$IP = 3.17/0.575 = 5.51$ bits/s for the first target.

Significance in model-based design:

- Predicts pointing time for buttons, menus and toolbars without testing

- Explains design rules: make frequently used targets large and near, place menus at screen edges (effectively infinite width), keep related items close
- Compares input devices through their IP values
- Supports evaluation: designers compute MT for alternatives and pick the faster

Hick-Hyman law
Decision time grows with information content:

$$T = b \log_2(n+1) \quad (\text{equal}), \qquad T = b \sum_i p_i \log_2\left(\frac{1}{p_i} + 1\right) \quad (\text{unequal})$$

With $b = 0.15$ s/bit: $n = 4$ gives 0.35 s, $n = 8$ gives 0.48 s, $n = 16$ gives 0.61 s.

Application in model-based design and evaluation:

- Limit the number of choices on one screen (7 +- 2 items)
- Prefer broad, shallow menus to deep hierarchies
- Order items by probability; frequent items are decided faster
- Predict decision time in a model such as KLM's mental operator

Together: Hick-Hyman predicts the time to **decide** and Fitts' law the time to **move**, so the two give a complete estimate of a selection task.

