

# Cours 5 - Principes, directives et patrons de conception

## Les principes de conception

Plus généraux, plus fondamentaux, plus vagues; s'appliquent dans plus de cas mais peuvent être moins évidents à appliquer

- Connaissez votre utilisateur
- Minimisez le besoin de mémoriser
- Optimisez les opérations
- Concevez en vue des erreurs possibles

## Principe de Shneiderman (principe conception) (IMPORTANT)

- Être cohérent (consistant)
- Concevoir pour différents utilisateurs (Seek universal usability)
- Donner un retour visuel informatif
- Indiquer à l'utilisateur que la tâche est complétée "Yield closure"
- Offrir une gestion d'erreur simple
- Rendre les actions réversibles
- Mettre l'utilisateur en contrôle (user initiates task it's not forced upon him)
- Réduire la dépendance à la mémoire court-terme

## Principe autre que Shneiderman

- User Engineering Principles for Interactive Systems (Hansen, 1971) (principe conception)
- Rubinstein et Hersh (1984) (principe conception)
- Les principes de Joel Spolsky (principe conception)

## Les directives de conception

Indiquent plus précisément quoi faire Plus précises, plus détaillées, s'appliquent dans moins de cas mais disent plus précisément quoi faire

Peux s'adapter selon le OS/environnement

- Permettre une navigation simple
- Rendre les cibles suffisamment grandes
- Réduire le désordre
- Afficher le texte suffisamment grand
- Utiliser des contrôles tactiles
- Utiliser des formulaires simples
- Tenir compte de la position du pouce
- Concevoir une expérience cohérente

## Avantages

- Stimules des idées de conception
- Donnent des conseils utiles
- Servent comme liste de vérification
- Peuvent être utilisé pour évaluer une interface

## Désavantages

- Parfois contradiction avec 2 directives
- Souvent trop vague ou trop spécifique
- Peuvent être trop nombreux

## Patron de conception

En GUI, un patron de conception est une solution récurrente à un problème commun de conception d'interface

## Exemple

- pull to refresh (apple mail)
- feature, search and brows (search bar in website)
- clear entry point (landing page where do i go from there for “something”)
- grid of equals (Netflix film browser)
- skeuomorphic (visual style)
- filmstrip (mobile like infinit scroll of feed)
- two-panel selector split view (spotify/discord)
- action panel (window start button)
- datatips (give info on hover google maps)
- forgiving format (input field autoformatting as you input phone number/credit card)

## Cours 6 - Esquisses et prototypage

### Prototype

Un prototype est un modèle ou un échantillon non complété servant à tester un concept ou un processus

On les utilise comme outils de conception et pour les tests avec les utilisateurs

### Types

- Prototypes statiques / dynamiques
- Prototypes de haute fidélité / basse fidélité
- Prototypes horizontaux (functionality) / verticaux (features)

### Avantages d’esquisser ou prototyper plusieurs alternatives en parallèle

- Permet d’explorer plus de possibilités et peut-être d’en combiner
- Permet d’éliminer une mauvaise idée de départ
- Séparation de notre égo des idées conçues – on accepte mieux les critiques
- Encourage la comparaison de différentes possibilités

### Dynamique

Un prototype qui offre un minimum de fonctionnalité à l’utilisateur

- Prototype animé
  - S’affiche écran par écran dans une forme de présentation par diapositives
- Prototype de type Magicien d’Oz
  - S’exécute sous forme de présentation par diapositives mais avec l’ajout d’intelligence pour afficher la diapo suivante selon l’action de l’utilisateur
- Prototype interactif
  - S’exécute sur l’ordinateur et répond à certains intrants en temps réel
- Prototype fonctionnel
  - S’exécute sur l’ordinateur, répond aux intrants en temps réel et performe certaines opérations en arrière-plan

### plusieurs moyen de prototyper statique

statique -> inverse de dynamique

- papier
- wireframe
- images
- static HTML
- .exe no features
- figma
- axure

# Cours 7 - Modélisation KLM

## KLM

se limite aux utilisateurs (experts) et aux tâches familières

- Déterminer les tâches élémentaires de l'interface.
  - e.g. ajouter un client, acheter un produit, etc.
- Établir les modèles KLM de chacune de ces tâches.
  - Modèle = séquence d'actions élémentaires
- Estimer les temps d'exécution  $T_i$  de ces tâches.
- Estimer les fréquences  $F_i$  des tâches pour un usager-type.
- Quantifier l'efficacité globale
  - cummation  $(f_i \cdot t_i)$ .

## Rules

- Homing (H) - 400ms
- Pointage (P) - 1100ms
- Keystroke (K) - 200ms
- Button press or release (B) - 100ms
- Activité mentale (M) - 1200ms
- Réponse du système (R ou W) -  $\max(t, 1200)$  ms
- Défilement (Scroll) - 3960ms
- Placer les M
  - Écrire séquence sans les M
  - Placer un M devant chaque P, B, k
  - Remplacer M consécutif par un M (MM->M)

## EG

- Écrire dlabbe au clavier -> M K M K M K M K M K M K M K =  $MK^5MK$  -> M au debut et fin
- {Click Delete} {Click Yes} {Click OK} -> MHPBB MPBB MPBB -> MHPK PBB PBB

Détruire un fichier depuis une liste

- M,H,P,BB Sélectionne l'icone du fichier
- M Activité mental
- H remet la main sur le clavier
- K Pese sur (DEL)
- R(0.500) Reponse (affiche la fenetre de dialogue)
- M Activite mental
- H Remet la mian sur le ponteur
- P Pointe sur l bouton OK
- BB Clique

On utilise FLM pour ecran tactile

# Cours 8 - Lois psychomotrices

## Loi de Fitts

$$T = a + b \cdot \log_2(D/W + 1)$$

- T ou MT = temps moyen de mouvement - D = distance - W = largeur cible (si 2D prend Min(x,y)) - a, b constantes mesurées -  $IP = 1/b$  -> indice de performance

La loi de Fitts modélise les mouvements réciproques (“aller retour”) et discrets (“un coup”) en 1D ou en 2D

### Se joue sur

- Distance à parcourir avec curseur
- Taille du curseur
- Taille du bouton

### Passer à travers un but

### Loi de Hick-Hyman (temps de réaction)

La loi de Hick-Hyman décrit le temps de réaction/décision selon le nombre d’alternatives présentées

$$\text{Temps de réaction} = a + b \cdot \log_2(N + 1)$$

- N = nb de choix - a, b des constantes mesurées de facons expérimental

## Cours 9 - Perception visuelle et graphisme

### Perception (vision)

- Éléments essentiels pour l’utilisation des GUI
- Derrière une grande partie des principes de conception des GUI
- **Vision** = processus biologique par lequel on capte la lumière et la transforme en impulsions électriques.
- **Perception** = manière dont le cerveau interprète ces impulsions.

### Vision périphérique

- On centre la fovéa (partie de l’oeil) à quelque part et notre cerveau remplit le reste avec la vision périphérique et nos attentes/connaissances.
- Implications pour le GUI
  - Si on mets un message d’erreur trop loin d’où la fovea est focus, l’utilisateur peut ne pas le voir

### Effet du “pop-out” pré-attentif

Lorsque notre attention est attiré automatiquement par quelque chose dans notre champ visuel, sans y porter attention.

Plusieurs variables: couleur, forme, taille, orientation, mouvement, espacement, lissage ou ombrage.

Conjonction de deux variables => Pas d’effet “pop-out”.

### Subitiser

- Limité jusqu’à 4 à 5 objets.
- Coûte 40 à 100ms par objet alors que compter coût 250 à 300ms par objet.

### Reconnaissance vs Rappel

- **Reconnaissance**: processus de reconnaissance d’information comme étant familière ou déjà vu précédemment. Très rapide, car ne nécessite pas de traitement comparatif.
- **Rappel**: processus de récupération active d’une information à long terme sans l’aide de stimuli extérieurs. Plus long et difficile, car on doit aller chercher dans la mémoire à long terme.
- Exemples dans les interfaces:
  - Icônes = reconnaissance des formes
  - Suggestions de recherche (Spotlight) = reconnaissance
  - Rappel de username/password = reconnaissance des indices
  - Raccourcis clavier = rappel

## Distinction des couleurs

- Trois facteurs impactent grandement notre capacité à distinguer deux couleurs:
  - Pâleur (luminance élevée, saturation basse)
  - Taille de l'échantillon
  - Espacement

## Composantes HSV

- Teinte (hue): fréquence "dominante";
- Saturation: "pureté" de la teinte (ex: rouge est plus saturé que rose, gris et blanc ne sont pas saturés du tout)
- Valeur (luminance ou lumière): mesure de clarté

## Couleurs dans les GUI

- Couleurs pures (saturées):
  - Attirent l'attention
  - Peuvent communiquer de l'excitation
  - Besoin de plus de *focus* que les couleurs moins pures (désaturées)
  - Peuvent entraîner de la fatigue oculaire
- Utiliser les couleurs saturées dans le UI pour faire ressortir un élément (avertissement, erreur, etc...)

## Perceptions

- Perception de structure: l'utilisateur cherche à percevoir la structure de l'interface, surtout pour:
  - Identifier les éléments formant un groupe logique
  - Identifier les informations importantes
  - Comprendre l'ordre logique de l'interface
- Perception de gestalt: perçoit les objets comme des ensembles cohérents plutôt que des éléments individuels.  
Quelques principes de Gestals:
  - Proximité
  - Similitude (continuité)
  - Sort commun (symétrie)
  - Figure / arrière plan

## Structurer l'information

### Patron en F

- Les premières lignes et les premiers mots à gauche sont toujours plus regardées que les autres
- On l'utilise quand:
  - une page n'a pas de mise en forme pour le Web (ex: mur de texte mais pas de gras, de puces ou de titres)
  - on essaie d'être le plus efficace sur cette page
  - on n'est pas assez engagé ou intéressé
- Créer une hiérarchie visuelle
- Ne jamais center un paragraphe de texte

## Cours 10 - Psychologie des utilisateurs

### Safe exploration

Il s'agit de permettre à l'utilisateur d'explorer une nouvelle interface de manière sécurisée en offrant des options de retour en arrière ou de réinitialisation pour éviter toute perte de données.

Exemple : Une application mobile qui permet de supprimer une tâche ou une note, mais qui demande une confirmation avant de la supprimer définitivement.

## **Instant gratification**

Les interfaces doivent être conçues pour offrir une expérience utilisateur satisfaisante et rapide, en donnant une réponse immédiate à une action de l'utilisateur.

Exemple : Un bouton "Ajouter au panier" qui affiche immédiatement une notification de confirmation.

## **Satisficing**

Les utilisateurs ne cherchent pas toujours la solution optimale, mais plutôt une solution satisfaisante. Les interfaces doivent être conçues pour offrir des solutions satisfaisantes pour les tâches courantes.

Exemple : Un site de commerce électronique qui propose des produits similaires ou liés à un produit sélectionné pour faciliter la recherche de produits connexes.

## **Changes in midstream**

Les utilisateurs peuvent changer d'avis ou d'objectifs en cours de tâche. Les interfaces doivent être conçues pour permettre aux utilisateurs de changer de direction ou de tâche en cours de route.

Exemple : Un formulaire qui permet aux utilisateurs de modifier ou de corriger les informations saisies précédemment sans avoir à recommencer depuis le début.

## **Deferred choices**

Les utilisateurs peuvent ne pas être prêts à prendre une décision immédiate. Les interfaces doivent être conçues pour permettre aux utilisateurs de différer certaines décisions.

Exemple : Un site de réservation qui permet aux utilisateurs de sauvegarder une réservation en cours pour la finaliser plus tard.

## **Streamlined repetition**

Les utilisateurs peuvent avoir besoin de répéter une tâche plusieurs fois. Les interfaces doivent être conçues pour faciliter la répétition rapide et facile des tâches courantes.

Exemple : Un formulaire d'inscription qui permet de sauvegarder les informations pour une inscription ultérieure, sans avoir à tout saisir de nouveau.

## **Loi de Miller (ou la loi magique de $7 \pm 2$ )**

Stipule que la capacité de la mémoire de travail humaine est limitée à environ 7 éléments (plus ou moins 2).

Exemple : Les numéros de téléphone sont généralement composés de 7 chiffres (et parfois plus), ce qui est facilement mémorisable pour la plupart des gens.

## **Loi de Postel (ou principe de tolérance)**

Stipule qu'un programme informatique doit être tolérant aux erreurs ou aux variations dans les entrées. Les utilisateurs ne sont pas des machines, ils peuvent faire des erreurs. Peuvent devenir irrités quand on leur demande trop d'informations.

Exemple : - Un programme qui peut décoder des fichiers multimédias même si les fichiers sont légèrement corrompus ou si certaines données sont manquantes. - Apple ID nous permet de se faire plusieurs choses (déverrouiller le téléphone, faire des achats, etc.)

## **Règle sommet-fin (peak-end rule)**

Stipule que les utilisateurs se souviennent mieux des premières et des dernières interactions avec un système, tandis que les interactions au milieu sont souvent oubliées.

Diminuer l'intensité des désagréments, les utilisateurs retiennent les mauvaises expériences plus que les bonnes.

Mettre un emphase pour rendre la fin de tâche la plus satisfaisante possible.

Exemple : - Un parcours client en ligne bien conçu qui crée une expérience utilisateur positive dès le début et qui se termine par une confirmation ou un résumé clair de l'interaction, améliorant ainsi la mémorisation de l'expérience. - Mettre de l'humour sur les pages d'erreurs (404)

## Effet usabilité-esthétique

Se réfère au fait que les interfaces visuellement attrayantes sont perçues comme plus faciles à utiliser, même si leur efficacité réelle est la même que celle des interfaces moins attrayantes.

Exemple : Un site Web avec une mise en page esthétique et des couleurs harmonieuses qui attirent l'attention des visiteurs et leur donnent une impression positive.

## Loi de Tesler (ou loi de la conservation de la complexité)

Stipule que la complexité ne peut pas être éliminée, mais seulement déplacée ou cachée. Elle doit être assumée par le système ou son utilisateur. Essayer d'assumer le maximum pour réduire le fardeau sur l'utilisateur.

Exemple : - Un logiciel qui propose des options avancées cachées dans des menus déroulants pour simplifier l'interface de base, tout en offrant des options supplémentaires pour les utilisateurs avancés. - Les suggestions de messages prédéfinis sur Gmail. - Utiliser la même adresse de shipping que celle de facturation.

## Modèle accrocheur (Hook Model)

Technique de conception d'interface utilisateur pour encourager l'utilisation continue d'un produit en créant une boucle de rétroaction (feedback loop) qui incite les utilisateurs à revenir (créer une habitude).

### Trigger (élément déclencheur)

Incite l'utilisateur à interagir avec le produit.

- **Externe:** vient de l'environnement de l'utilisateur. (ex: courriel, message texte, notification)
- **Interne:** vient de l'utilisateur lui-même, souvent une émotion ou un besoin. (ex: ennui, solitude, recherche de valorisation)

Exemple : Un réseau social qui envoie des notifications pour signaler de nouvelles activités ou messages sur la plateforme pour inciter les utilisateurs à revenir régulièrement.

### Action

L'action que l'utilisateur doit faire afin d'obtenir une récompense ou satisfaction qu'il anticipe.

Exemple: Cliquer sur une notification.

### Récompense

Élément qui procure une satisfaction à l'utilisateur après avoir accompli une action. Elle doit être **immédiate** et **variable** (imprévisible) pour maintenir l'intérêt de l'utilisateur.

- Trois natures:
  - Tribe (tribue): récompenses sociales basées sur la connexion et l'acceptation.
  - Hunt (chasse): recherche de ressources matérielles.
  - Self (soi): gratification personnelle sous forme de maîtrise ou de réalisation de soi.

Exemple : L'utilisateur lit le message et ressent une satisfaction sociale.

### Investissement

Encourage l'utilisateur à répéter l'action dans le futur. Cela peut inclure le temps, l'argent ou les données personnelles investis dans l'utilisation du produit ou service. Peut avoir l'effet d'entreposer de la valeur dans le produit.

Peut mener indirectement au prochain *trigger* externe (ou même interne).

Exemple : - L'effet IKEA, le fait que le client ai construit lui-même le meuble le rend plus précieux à ses yeux. - L'utilisateur répond au message et investit du temps et de l'énergie dans la conversation pour renforcer ses liens sociaux sur la plateforme.

## Cours 11 - Méthodologie d'évaluation

### Méthodologies d'évaluation

#### Tests d'utilisabilité

- Motivation: on dispose de connaissances précises sur l'équipement (specs, documentation, etc.). Mais nous n'avons pas ce genre d'informations pour les utilisateurs.
- Objectifs:
  - Évaluer expérimentalement l'efficacité et l'efficience
  - Détecter des problèmes ponctuels nuisant à l'utilisabilité
  - Évaluer l'utilisabilité globale du produit
- Contextes d'utilisation:
  - Durant le développement, dès qu'on a un prototype fonctionnel
  - Avant de faire une nouvelle version, complément des enquêtes et des focus groups
  - Ne pas les utiliser pour trouver des erreurs de logiciel
- Composantes de l'utilisabilité:
  - Efficacité (faire les tâches requises)
  - Efficience (faire les tâches rapidement)
  - Satisfaction (apprécié par l'utilisateur)
- Types de tests d'utilisabilité:
  - Tests d'évaluation: trouver des problèmes spécifiques nuisant à l'utilisabilité
  - Tests de comparaison: comparer deux solutions (concurrent)
  - Tests d'acceptation: valider le produit selon des niveaux d'utilisabilité prédéfinis

#### Le plan de test

##### Contenu du plan de test

1. But du test
2. Énoncé des objectifs (ou questions) du tests
3. Profil de l'utilisateur et justification des sujets
4. Protocole de test
5. Matériel et environnement de test
6. Déroulement du test
7. Analyse des résultats
8. Discussion et recommandations

#### But des tests

- Fournit à un haut niveau les raisons pour réaliser un plan de test
- Est-ce que les tests visent à résoudre des problèmes résultants des commentaires des utilisateurs?
- Est-ce le résultat d'une nouvelle politique organisationnelle qu'il faut tester?
- Est-ce à la demande des gestionnaires qui croient essentiels de faire certain types de tests?

#### Exemples

- Le test vise à déterminer si tous les principaux clients peuvent utiliser le produit correctement et s'assurer tenir compte des différences s'il y a lieu
- Le test vise à vérifier si la documentation peut compenser pour certains problèmes d'interface connus.



- Suite à plusieurs plaintes concernant le panneau de contrôle d'une application nouvellement utilisée, le test vise à connaître la nature de ce problème et identifier comment on il pourrait être réglé

## **Le test**

### **Objectifs du test**

- Les objectifs doivent être :
  - Utiles, précis et vérifiables par observation
  - Adaptés au type de test
  - Peuvent être formulés sous forme de questions auxquelles on veut répondre

### **Tests d'évaluation**

#### **Exemples**

- Est-ce qu'une zone de message de 20 caractères est suffisante ?

#### **Contre-exemples**

- Est-ce que l'aide est bien organisée ? (Trop général)
- Est-ce que le produit est prêt pour être commercialisé ? (Manque de données (sur le marché) pour répondre)
- Quel est le temps d'exécution moyen de la tâche A ? (Question quantitative (test de performance))

### **Tests de comparaison**

#### **Exemples**

- La procédure d'achat d'item est-elle plus intuitive dans A que dans B ?
- Est-ce que le bouton UNDO est utilisé plus fréquemment dans A que dans B ?
- Est-ce que les menus de A permettent un accès plus rapide aux options courantes que ceux de B ?

#### **Contre-exemples**

- Est-ce que A est plus efficace que B ? (Trop général)
- Est-ce que A est plus facile à apprendre que B ? (Trop général)
- Est-ce que A est un meilleur achat que B ? (Manque de données pour répondre)

### **Tests d'acceptation**

#### **Exemples**

- Vérifier si 70% des utilisateurs arrivent à imprimer sans aide en moins de 5 minutes.
- Vérifier si les utilisateurs novices ont un taux de succès moyen supérieur à 90% dans les tâches assignées.
- Vérifier si les utilisateurs experts peuvent transcrire une page en 5.5 minutes.
- Vérifier si les utilisateurs trouvent ce qu'ils cherchent dans l'aide dans plus de 90% des situations.

#### **Contre-exemples**

- Vérifier si le produit est rapide et fiable. (Trop général)
- Vérifier si le taux d'erreur est inférieur à 2%. (Concept de taux d'erreur mal défini (% de quoi ?))

### **Profil de l'utilisateur et justification des sujets**

Il faut décrire le profil de l'utilisateur (âge, sexe, études, etc.).

## Composition du groupe de sujets

- Beaucoup de flexibilité pour composer le groupe de sujets
- Groupe de sujets représentatifs des utilisateurs type
- Chercher la diversité plutôt que l'homogénéité.
- Enregistrer les paramètres socio-démographiques.
- Age, sexe, langue, années d'étude, de pratique avec l'ordinateur ou le logiciel testé.
- Critères d'exclusion (dépressif, médication, etc.)

## Nombre de sujets

- Pré-test
- 2 ou 3 sujets, généralement les développeurs ou des proches
- Pour mettre au point le protocole expérimental
- Test
- Tests d'évaluation: 4 à 8 sujets (recette empirique)
  - Selon J. Nielsen (1993) 5 à 7 sujets suffisent pour détecter 80% des erreurs (chiffre à prendre avec prudence)
- Tests de comparaison: 4 à 8 sujets (recette empirique)
- Tests d'acceptation: le plus possible (ex. 30 sujets)

## Considérations éthiques et légales

- Les sujets (participants) doivent être libres et consentants.
- Complètement informés.
- Pouvoir se retirer à tout moment.
- Aucune donnée nominale ne doit être informatisée.
- Tout est résumé dans un formulaire de consentement.

## Recrutement des sujets

- Personnel de l'entreprise
  - Ne pas les faire participer trop souvent (syndrome du “testeur corporatif” )
- Connaissances, conjoints, parents
  - Ne pas les faire participer trop souvent (syndrome du “testeur professionnel” )
- Babillards ou journaux
  - Étudiants (campus, collèges...), personnes âgées (Bingo, clubs...).
  - Tout venant (supermarchés).
  - Compensation financière de 10 à 50\$ environ selon le temps mis.
  - Alternative: paiement des frais de déplacement ou repas.

## Protocole de test

### Établir la liste de tâches qui seront faites par les sujets

- La liste de tâches est établie en fonction des objectifs ou questions des tests
- La tâche doit être très spécifique et doit inclure toute l'information nécessaire à la complétion de cette tâche
- La tâche ne doit pas décrire les étapes nécessaires pour effectuer la tâche

### Faire une matrice sujet x tâche.

- Les sujets peuvent tous faire les mêmes tâches ou un sous-groupe de tâches selon leur profil
- Ordre des tâches.
  - Contrôler cet ordre pour éviter (partiellement) la facilitation.
  - Ex. : Tâche 1 facilite la réalisation de Tâche 2.
- Randomiser?
  - Ordre pseudo-aléatoire.
  - Ordres différents entre les sujets?

- Cas des tests de comparaison.
  - Même tâches faites avec 2 produits?
  - 50% des sujets commencent par A, et 50% par B?
  - certains sujets travaillent exclusivement sur A, d'autres sur B?

### Protocole de test

- Groupes indépendants ( *Between subjects design* )
- Intra-participant ( *Within-subjects design* )

### Matériel et environnement de test

#### Documents nécessaires pour les tests:

- Liste de contrôle à l'attention de l'expérimentateur
- Procédure de test à l'attention de l'expérimentateur
  - Inclut les procédures d'exception (ex. quoi faire si le sujet se sent mal)
- Formulaire de consentement
- Fiche de renseignements à remplir par le sujet.
  - Renseignements personnels + données socio-démographiques.
- Consigne à lire au sujet (aussi appelée script d'orientation).
- Liste de tâches destinée au sujet
- Feuilles d'observations
  - Pour noter ce qui se passe durant le test.

#### Logiciel et équipement:

- Prototype fonctionnel (ou application finale).
  - Interface suffisamment complète pour faire les tâches.
  - Nombre minimal de bogues.
- Ordinateur correspondant au type d'équipement où l'application sera déployée. - Pour les applications web, connexion présentant les caractéristiques voulues.

### Matériel et environnement de test

#### Local et poste de travail

- Configuration simple.
- Expérimentateur
- Observateurs
- Caméra
- PC + Manuels
- Sujet

#### Local et poste de travail:

- Si possible, salle avec des conditions contrôlées (insonorisée, contrôle lumière)
- Poste de travail du sujet.
- Poste de l'expérimentateur (en arrière du sujet, interfère pas, rassure le sujet)
- Configuration complète.

#### Salle d'observation annexe

- Permet aux observateurs de converser sans interférer
- Vitre semi-transparente
  - Permet de voir directement ce qui se passe
  - Les moniteurs vidéo ne donnent pas toujours une bonne idée
- Système de son
  - Écouter les commentaires du sujet et de l'expérimentateur

- Moniteur d'ordinateur dédoublé + moniteurs de vidéo
  - Voir en même temps les actions du sujet + l'interface

### Dispositifs d'enregistrement

- Caméras vidéo + son
  - Aide à compléter et corriger les observations
  - Permet de voir les actions du sujet depuis la pièce voisine
  - Garde une trace de l'expérience pour archivage
- Programme espion
  - Aide à compléter et corriger les observations
  - Logs filtrés (calculer temps d'exécution et/ou détecter erreurs)

### Déroulement du test

1. Lire la consigne et donne la liste de tâches
2. Indiquer un temps global et un temps pour chaque tâche
3. Demander de penser à voix haute
4. Séance de débriefing

**Expérimentateur:** - Préparer le local - Accueillir, faire remplir le formulaire - Expliquer les tâches - Noter les grandes lignes de ce que fait l'utilisateur - Noter les erreurs - Encourager ou intervenir si nécessaire - Après transcrire ses notes, sauvegarder les données et archiver les documents

Il y a des phrases à ne pas dire à l'utilisateur (common sense)

### Résultats

- On fait une interprétation et on ne tire pas de conclusion
- Transcrire intégralement les feuilles d'observation
- Analyse qualitative par idées ou événements-clés
- Idées ou événements-clés
- Analyse quantitatives grossières
- Statistiques simples
- Graphiques (tartes, diagrammes de barres)
- Tableaux descriptifs

### Discussion et recommandations

- Discuter de la signification des résultats
- Faire des recommandations

### Tests d'usabilité

- Limitations générales:
  - Protocole et analyse arbitraire (voit ce qu'on veut voir)
  - Protocole peu rigoureux (résultat moins fiables)
  - Sujets peuvent fausser les résultats
  - Investissement (vidéo, salle, etc...)
  - Besoin d'expertise (conception, conduite des tests et l'analyse)

### Tests d'utilisabilité de type «guérilla»

- Forme allégée et moins formelle des tests d'usabilité
- Aller dans un café trouver des utilisateurs
- Fait sur le terrain et non dans un environnement contrôlé
- Permet de recueillir les mêmes infos qu'un tests formel, avec moins de ressources

## Conseils

- Beware the implicit bias
- Explain what's going on
- Be ethical
- Make it casual
- Be participatory
- Don't lead participants
- Keep your eyes peeled
  - Take notes for later
- Capture the feedback
- Be a timecop
  - Don't waste too much of the subject's time
- Pour tests sur ordi portable:
  - Enregistrer l'écran, caméra et micro
  - Logiciel qui permet de voir les cliques de souris

## Avantages

- Moins de planification
- Peut être fait dans le contexte de l'utilisation de l'application
- Vérifier certains aspects spécifiques avec quelques utilisateurs
- Peut être appliquée à plusieurs étapes du développement

## Inconvénients

- Moins de contrôle de l'homogénéité des participants
- Participants plus distraits
- Tests moins rigoureux
- Moins grandes généralisabilité des résultats

## Les groupes de discussion (focus groups)

- Recueillir des informations sur des thèmes pré-établis
- On donne plus d'info dans une discussion car on réagit à l'opinion des autres >
- Objectifs:
  - Évaluer la perception que les utilisateurs ont du système
  - Évaluer la satisfaction des utilisateurs
  - Identifier/comprendre certains problèmes d'usabilité >
- Contextes:
  - Marketing
  - Ingénierie
- Avantages
- Idée de la perception du produit
- Le cadre est proche des conversations de couloirs
- Résultats différents des entrevues individuelles
- Saute pas directement aux conclusions
- Limitations:
- Organisation difficile
- Risque de dérive consensuelle (groupe se plie à un leader)
- Le dit et le non-dit (disent pas leurs motivations profondes)
- Les résultats dépendent beaucoup du modérateur
- Analyse doit être rigoureuse

## Les enquêtes

- Collecter des informations en posant des questions aux users
- Le volume compense l'imprécision et l'inexactitude des réponses >
- Objectifs:
  - Mieux connaître les **utilisateurs** et les **utilisations** du système
  - Évaluer la **perception**
  - Évaluer la **satisfaction** >
- Contexte d'utilisation:
  - Faire une nouvelle version du système
  - Estimer le succès d'un produit >
- Trois façons de faire l'enquête sur des utilisateurs:
  - Questionnaire en ligne
  - Questionnaire sur papier
  - Entrevues individuelles >
- Quand ?
  - Mise à jour du système par l'utilisateur (update, product registration)
  - Après un temps de fonctionnement (bêta-testeurs, utilisateurs pilotes) >
- Recommendations:
  - Bref
  - Termes neutres
  - Réponses exhaustives
  - Libre de répondre
  - Provoquer des commentaires non dirigés en entrevue (garder le silence) >
- Avantages:
  - Bonne idée de la perception
  - Résultats utilisables pour le marketing
  - Recevoir beaucoup d'information >
- Limitations:
  - Coût et temps
  - Utilisateurs donnent une fausse image d'eux-mêmes
  - Facteurs humains fausse les statistiques
  - Facteurs techniques fausse les statistiques >
- Remarques:
  - Adapter son approche à la situation
  - Tenir compte des individus
  - Tenir compte de votre environnements

## Cours 12 - Dispositifs d'interaction

### Dispositifs de saisie

- Dvorak
  - Toutes les voyelles sont sous une main dans la rangée du milieu
- Barre tactile du Macbook
  - Avantages:
    - \* Fonctionnalités spécifiques aux applications
    - \* Rester au clavier pour des trucs fait à la souris habituellement
    - \* Façon additionnelle de faire certaines tâches
  - Désavantages:
    - \* F1, F2, F3 requièrent 2 touches
    - \* Pas de retour tactile
    - \* Pas de constance des touches
    - \* Fonctionnalités redondantes car pas tous les ordi qu'ils l'ont
    - \* Distraction
    - \* Cliquer par accident
    - \* Taille des touches (Fitts)

- Clavier virtuel du iPhone
  - Caractère affiché décalé pour éviter l'occlusion
  - Caractère saisi au relâchement pour corriger si nécessaire

### Comparaison de vitesses en mots/minute (wpm)

- Écriture: 20-30 wpm (>300 wpm avec sténographie)
- Dactylographie professionnel sur clavier normal: 50-100 wpm
- Mots parlés: ~100 wpm
- Sténotype professionnel: 100-200 wpm
- Mots lus: 200-300 wpm

## Dispositifs de pointage

Première souris en 1968 par Douglas Engelbart (Standfort Research Institute).

### Propriétés avantageuses de la souris

- Le poids stabilise et atténue les tremblements
- Direction de mouvements des boutons est perpendiculaire au plan de mouvement de la souris. Peut appuyer sur un bouton sans affecter la position de la souris.
- Lâcher et ressaisir la souris sans changer sa position

### Alternatives

- Tablette numérisante (digitizing tablet, tablette graphique)
- Souris stylet (mouse pen)
- Pavé tactile (touchpad)
- Boule de commande (track ball)
- Manette, manche à balai (joystick)
- Écran tactile (touchscreen)
- Oculomètre (eye tracking)

### Propriétés des dispositifs de pointage

*Savoir catégoriser*

- Capture absolue vs relative
  - Exemple: la souris capte des mouvements relatifs alors que les tablettes numérisantes captent une position absolu (peuvent être utilisés en mode absolu)
- Pointage direct vs indirect
  - Pointage direct: les espaces d'entrée et de sortie coïncident
  - Exemples:
    - \* Une souris ou une tablette numérisante sans écran permettent un pointage indirect.
    - \* Un écran tactile ou tablette numérisante avec écran permettent un pointage direct
  - Pointage direct est plus intuitif
- Capture discrète vs continue
  - Exemples:
    - \* Une souris capte une position continue, mais pourrait l'arrondir vers une de  $N$  positions discrètes si on voulait.
    - \* Les flèches, ou interrupteur à  $N$  positions permettent de capter des données discrètes
- Contrôle de position vs contrôle de vitesse
  - Contrôle de vitesse: la *position* du périphérique détermine la *vitesse* à laquelle une autre variable (ex: position du curseur) change.
  - Exemples:
    - \* Souris capte une position et permet un contrôle de position ou bien un contrôle de vitesse.
    - \* Manette isométrique ne permet qu'un contrôle de vitesse.

## Performance de ces périphériques

- Souris
- Écran tactile
- Souris stylet
- Boule de commande
- Pavé tactile
- Manette

## Modèle à trois états de Buxton (1990)

### Dispositifs avancés

- Capture de mouvement (interfaces gestuelles) (Kinect, Leap Motion, Hololens, Oculus Quest 2, Omni)
- Activation musculaire (Myo)
- Interfaces cerveau-machine (EEG, implants)

## Révision pour l'examen final

- 6 concepts clés de Norman (modèle conceptuel)
  - Affordances
  - Contraintes
  - Visibilité
  - Mapping
  - Retour (feedback)
  - Constance
- Ces concepts s'appliquent autant aux objets physiques qu'aux interfaces virtuelles

*Pouvoir donner des exemples des concepts*

- Analyse de tâche
  - Liste des tâches à faire sur l'interface
  - Description du domaine et contexte d'utilisation
  - Profil des utilisateurs-type
  - Glossaire du domaine
  - Description des tâches réalisées actuellement
  - Matrice intervenants-tâches

*Faire une analyse et la matrice intervenants-tâches (pas juste qui fait quoi)*

- Principes de conception
  - Plus généraux, plus fondamentaux, plus vagues: s'appliquent dans plus de cas mais peuvent être moins évident à appliquer.
  - Exemples: 8 règles de Shneiderman, réduire la dépendance à la mémoire, etc...
- Les directives de conception
  - Plus précises/détaillées, s'appliquent dans moins de cas mais disent plus précisément quoi faire.
  - Exemples: changer la couleur d'un hyperlien quand il est visité.
- Esquissage et prototypage

*Savoir les différences entre les différents types, qu'est-ce qui les distingue.*

- Éléments de KML



- Exercice KLM
  - Trouver les erreurs

*Capable de faire des scripts KLM, revoir la question sur la saisie de texte.*

- Loi de Fitts:  $T = a + b \cdot \log_2(D/W + 1)$

*Pour a et b on a besoin des temps de pointages ou les difficultés.*

*Comme dans le quiz. Reconnaître quel disposition est plus performante. Comment obtenir a et b.*

*Placer en ordre croissant de difficulté.*

- Loi de Hick-Hyman: Temps de réaction =  $a + b \cdot \log_2(N + 1)$

*Capable de l'appliquer, comparer deux profondeurs de menu.*

*Qu'est-ce qu'on a besoin pour qu'elle s'applique: l'utilisateur sait ce qu'il cherche, ordre logique.*

*Créer une structure visuel avec le principe de proximité, similitude, etc...*

- Vision et perception

*Note: Voir les questions du quiz dans le dernier PDF. C'est vrai que noir sur blanc et blanc sur noir c'est aussi difficile. Faux pour les des yeux.*

- Modèle accrocheur
- Questions du quiz
  - Recommandations du système: Récompense
  - La recherche de validation en est un exemple: Trigger interne
  - Peut être une émotion: Trigger interne
  - L'association avec un endroit physique en est un exemple: Trigger interne
  - Doit être variable: Récompense
  - Peut charger (load) le prochain élément déclencheur: Investissement

## Informations sur l'examen final

- Droit aux notes mais l'examen est fait en conséquence
- Choix de réponses
- Pas de définitions
- Voir si on a compris avec identification des exemples
- Les exercices peuvent dire qu'est-ce qui est important
- Hybride
  - Moodle (comme les quiz, associations, etc.)
  - Papier (plus développement, mettre en application dans la matière)
    - \* concevoir une interface selon un use-case
    - \* analyser une interface
    - \* réorganiser une interface
    - \* proposer un protocole de tests