



Conception d'Interfaces

+ Bases de Kotlin, Android Studio et Compose

8INF865

UQAC

Université du Québec
à Chicoutimi



Ordre du jour

- Interface terrible (activité)
- Principes de conception d'interfaces (de Norman)
- Intuitivité et modèle mental
- Persona
- Conception centrée sur la personne utilisatrice
- Concepts spéciaux
- Kotlin
- Android Studio (démonstration)
- Jetpack Compose (exercices - test d'autonomie)

Activité

Interface terrible



Activité interface terrible

10 minutes (maximum)

Instructions



Compléter le formulaire sur
votre téléphone

Moodle (lien web)

Chaque fois que vous faites une
«erreur» ou que le site ne répond
pas comme vous vous attendiez :

1. Arrêtez-vous, prenez quelques secondes
2. Réfléchissez, prenez des notes si vous le voulez

Activité

Interface terrible

Rétroaction/Discussion

00:00:55

1 2 3 4

1 / 4

Choose Password

Your email @ Domain other

☒ I do not accept the [Terms & Conditions](#)

Next Cancel

How can we help?

[Help](#) [Send to bottom](#)

VERHA

Your p
Your pas
Your p
Your pas
Your pas

Principes de conception d'interfaces (de Donald A. Norman)

<https://www.nngroup.com>



Principes de conception d'interface de Norman

 Visibilité

 Rétroaction

 Contraintes

 Correspondance

 Consistance

 Affordance



Visibilité



Principe de conception d'interface de Norman

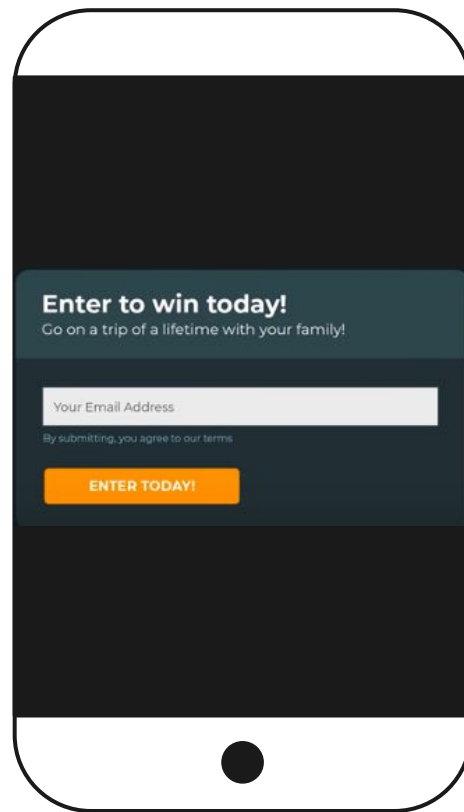
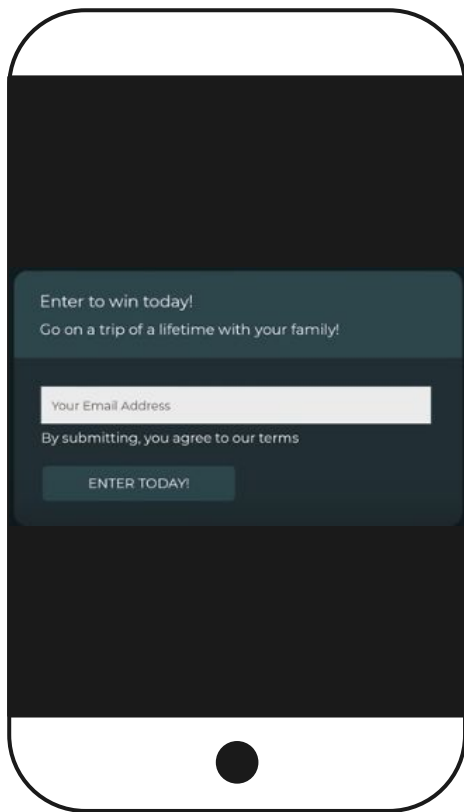
Plus les fonctionnalités sont visibles, meilleures seront les chances que la personne utilisatrice sache quoi faire pour atteindre son objectif



Exemple
Compromis entre les menus
latéraux et les barres d'onglets
dans les applications mobiles

<https://www.sachinrekhi.com/don-norman-principles-of-interaction-design>

Visibilité



Visibilité (parfois juste dans le détail)



Rétroaction



Principe de conception d'interface de Norman

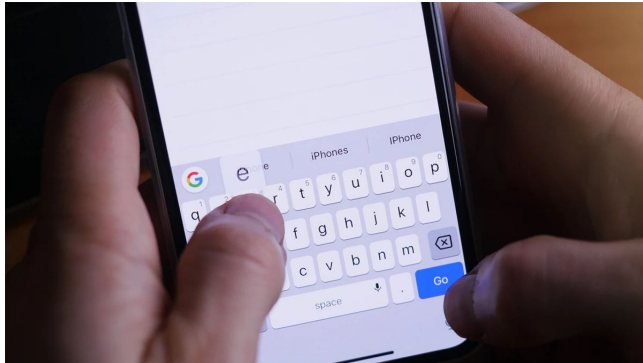
Retour d'information pour
confirmer l'action de la personne
utilisatrice

- Visuel
- Audio
- Tactile





<https://support.apple.com/fr-ca/HT213404>

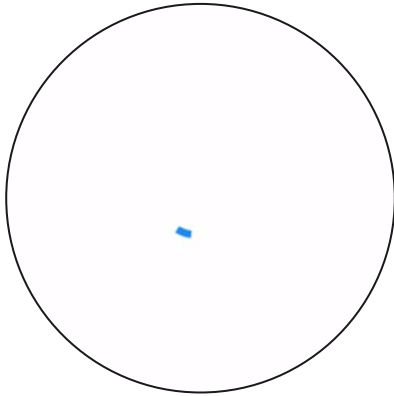


https://www.phonearena.com/news/Finally-Haptic-feedback-arrives-to-iPhone-thanks-to-Gboard_id113694

Exemple
Clavier virtuel offrant trois
modalités de rétroaction

Rétroaction

Exemple Chargement



<https://m2.material.io/components/progress-indicators>

Rétroaction

Attention

- Perception de lenteur
- Signe de progression manquant
 - Période d'incertitude
 - Effet de surprise

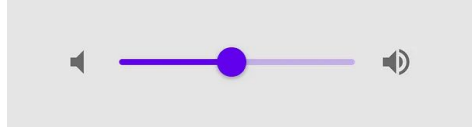
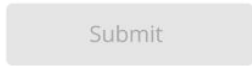
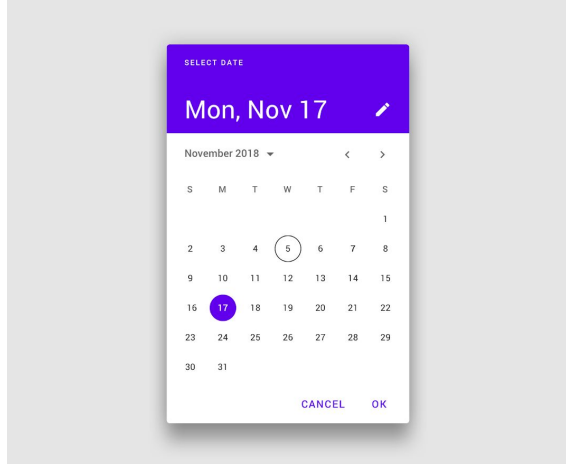
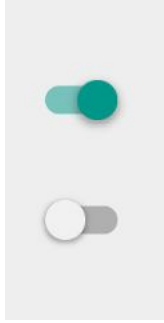


Contraintes



Principe de conception d'interface de Norman

Limite les actions possibles de la personne utilisatrice, réduisant les risques d'erreur et la charge cognitive



<https://m2.material.io/components>

Contraintes



Correspondance

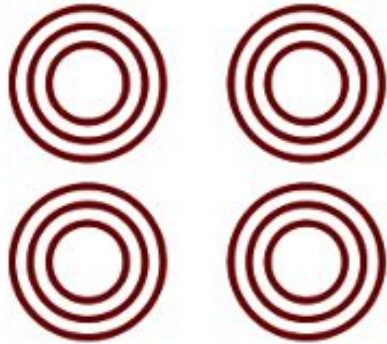
(mapping)



Principe de conception d'interface de Norman

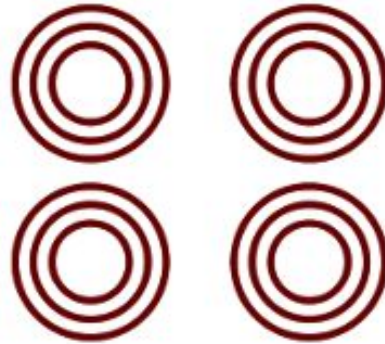
La relation entre les contrôles
et leurs effets réels

Ambigu



Back Left Back Right
Front Left Front Right

Correspondance
naturelle



Back Left Back Right
Front Left Front Right

<https://bradorego.com/ditw-stovetop>

Correspondance

Défilement

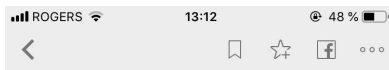


Photo by Vjieran Pavic / The Verge

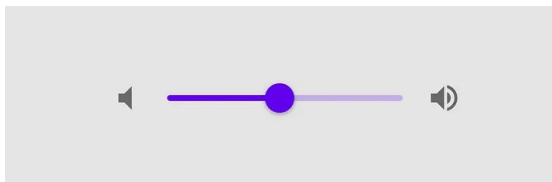
We know for a fact that Valve's hardware ambitions didn't end with [the Steam Deck](#). Over the past two years, Valve has suggested it would like to see [a standalone VR headset](#), [a new Steam Controller](#), and a Steam Deck revision [with better battery life and screen](#).

Now, Valve may be actually getting ready to ship at least one of its hardware ideas. South Korea's National Radio Research Agency [has certified](#) a "low power wireless device" from Valve with the designation "RC-V1V-1030," as spotted [by @dxpl at Arca.live](#) (via [Brad Lynch](#)).

A new Valve hardware device just

Correspondance

Glissière



Icône





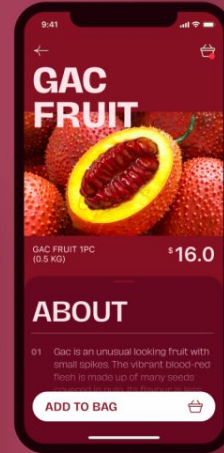
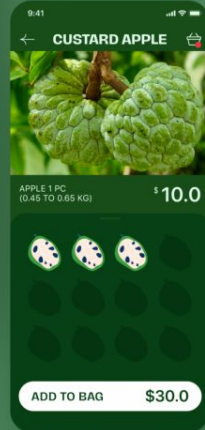
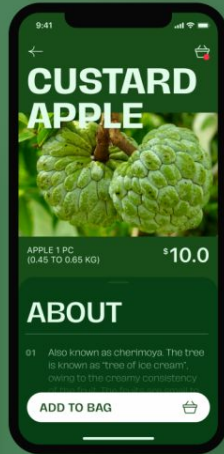
Consistance

(cohérence)

=

Principe de conception d'interface de Norman

Les éléments similaires devraient avoir des fonctions similaires à l'intérieur de l'application mobile, **mais également entre les applications mobiles**



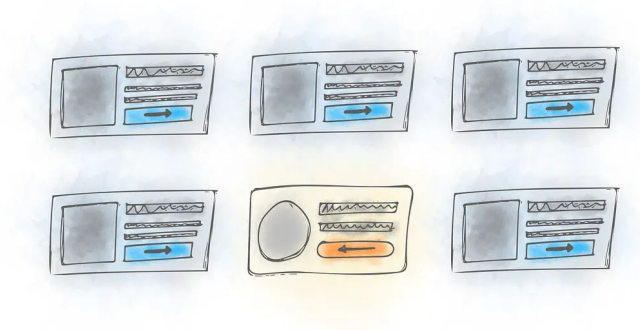
<https://blog.tubikstudio.com/design-consistency>

Consistance

Consistance *(cohérence)*

Loi de Jakob (Nielsen)

Les personnes utilisatrices de votre ~~site~~ application mobile passent la majorité de leur temps sur d'autres ~~sites~~ applications mobiles



<https://uxdesign.cc/design-principle-consistency-6b0cf7e7339f>



Consistance *(cohérence)*

Suivre les lignes directrices « officielles »

- Google Material Design
<https://material.io/design>
- Apple Human Interface Guideline
<https://developer.apple.com/design/human-interface-guidelines/platforms>



Affordance

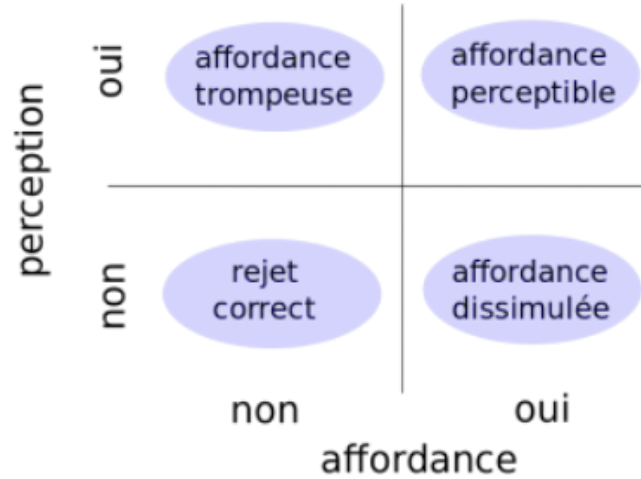
(potentialité)



Principe de conception d'interface de Norman

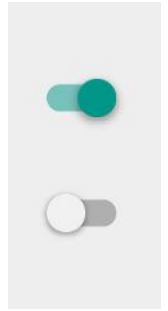
L'affordance désigne une interaction suggérée et réalisable pour laquelle l'élément a été conçu

Cliquez sur le [lien](#) pour plus d'information



<https://fr.wikipedia.org/wiki/Affordance>

Affordance



Cliquez sur le lien pour plus d'information

Intuitivité et modèle mental



Mythe de l'interface « **intuitive** »

Intuitif

« Qui procède par intuition ; que l'on a par intuition »

Intuition

« Connaissance directe, immédiate de la vérité, **sans recours au raisonnement, à l'expérience.** »

- LAROUSSE

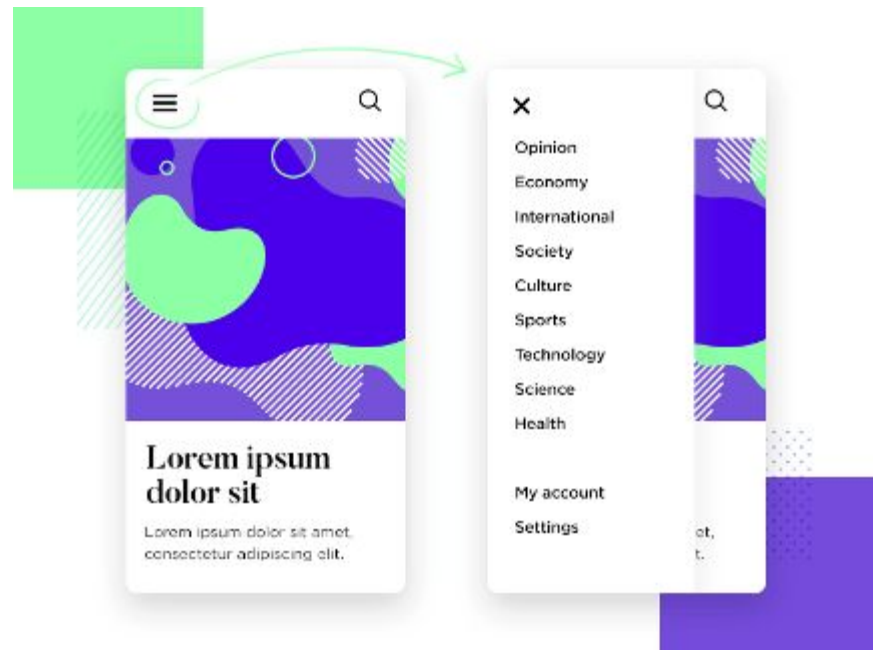
Intuitif → Familier

"Intuitive = uses readily transferred, existing skills"

Intuitive Equals Familiar. By Raskin, J. 1994



Mythe de l'interface « intuitive »





Modèle Mental



L'ensemble des croyances de la personne utilisatrice par rapport au fonctionnement et comportement d'un système ou logiciel (d'une application mobile)

Modèle Mental



- Basé sur des croyances, **pas des faits**
- Chaque personne utilisatrice a son propre modèle mental (MM)
- Constamment mise à jour (ex., lien hypertexte souligné)

Pourquoi est-ce important ?

- Utilisé pour choisir comment interagir avec l'interface

Persona



Persona

Qu'est-ce que c'est ?

Un personnage imaginaire représentant un groupe de personnes utilisatrices partageant des caractéristiques expérientielles et/ou comportementales d'intérêt pour notre application mobile

- <https://xd.adobe.com/ideas/process/user-research/putting-personas-to-work-in-ux-design>
- <https://www.smashingmagazine.com/2014/08/a-closer-look-at-personas-part-1>
- <https://www.imagescreations.fr/qu-est-ce-qu-un-persona>



Persona

Quelle information y retrouve-t-on ?

Minimum

- Information personnelle et démographique
- Objectifs
- Frustration
- Comportements



Nerdy Nina

"The book is way better than the movie!"

#booklover

#bookaddict

#booknerdproblems

DEMOGRAPHICS

Age: 25
Location: Sao Paulo, Brazil
Education: Software Engineer
Job: Q/A at Indie Game Company
Family: Lives with her boyfriend

TECH

Internet
Social Networks
Messaging
Games
Online Shopping



GOALS

- Discovering new books / authors to read
- Finding unique stories
- Cataloging book collection

FRUSTRATIONS

- Keeping track of different series
- Forgetting a book launch date
- Finding space for more books

READING HABITS

- Fast pace reader
- Never lends books
- Likes hardcovers and boxed collections
- Pre-order books to get them first
- Reads eBooks, but prefer physical copies
- Always finishes a book
- Loves binge reading and re-reading

FAVORITE BOOKS



American Gods
Neil Gaiman



Harry Potter
J.K. Rowling



Ready Player
One

<https://dribbble.com/shots/3052941-UX-Persona-for-Book-App>

Exemple (persona pour une application de suggestion de livre)



USER PERSONA

Olivia

"I don't want my hectic work schedule to interfere with my fitness goals."

ABOUT

Olivia is a Public Relations and Communications Manager for a leading financial advisory and consulting firm. She prefers to be active in her free time, but travels a lot for work. Growing up, Olivia was active in sports and she continues to go to CrossFit and yoga classes as often as she can.

Age: 29

Work: PR Manager

Income: \$80K to \$100K

Status: Single

Location: Toronto, ON

NEEDS

To maintain an exercise routine that fits her changing work schedule.

To find drop-in fitness classes in new cities that don't require a membership.

The ability to pay for classes in advance to save time.

Recommendations to help determine the best class for her experience level

Extroverted

Leader

Independent

Athletic

FRUSTRATIONS

Travelling for work makes it difficult to keep a gym membership.

Using a web search to find classes that match her experience level is complicated and time-consuming.

Making a phone call to cancel a class if her work schedule changes.

Manually keeping track of classes she enjoys and would return to the next time she travels to the same city.



<https://clearbridgemobile.com/how-to-create-user-personas-for-app>

Exemple (persona pour une application de fitness)



Personas

À quoi ça sert ?

- Développer de l'empathie pour les personnes utilisatrices
- Soutient la prise des décisions de conception
 - Conception **POUR** les personas...
 - Est-ce que les idées sont cohérentes avec la réalité des personas ?
- Soutient la communication des concepts



Personas

Qu'est-ce qu'une bonne persona ?

1. Idéalement générée à partir de données réelles, issues d'un processus de recherche, d'observations, d'entrevues...
2. Représente de vraies tendances chez les personnes utilisatrices et pas des rôles stéréotypés
3. Se concentre sur le présent et non le futur
4. Contient surtout des informations spécifiques au contexte et domaine d'intérêt



Personas (trois sortes)

Proto-persona

Catalogue et aligne les connaissances et hypothèse de l'équipe sur les personnes utilisatrices de l'application

Persona qualitative

Issues d'un processus de recherche qualitative

5-30 personnes participantes, principalement des entrevues et observations

Persona statistique

Recherche méthodes mixtes

5-30 personnes participantes en entrevues **ET** déploiement de questionnaires (plus de 500 personnes participantes)



Personas

Problèmes fréquents

- Des personas sont créées, mais pas utilisées
- La direction ne croit pas aux bénéfices des personas sur l'alignement de l'équipe de conception, de développement, de marketing...
- Personas créées en silos, sans l'implication des différentes parties prenantes
- Les personas sont mal adaptées à la portée du projet



Proto-Persona

*Catalogue et aligne les
connaissance et hypothèse de l'
équipe sur les personnes
utilisatrices de l'application*

Comment les créer ?

Format

Atelier (2-4 heures)

Qui

Équipe de conception, personnes clientes et autres parties prenantes

Comment

1. Chaque personne participante doit créer entre 2 et 5 proto-personas en utilisant un gabarit simple, puis les partager avec le groupe
2. Le groupe discute de toutes les personas, les combine et modifie les différents attributs pour arriver entre 3 et 6 proto-personas au total, représentant divers segments des personnes utilisatrices

Conception Centrée sur la personne Utilisatrice (CCU)

“On est en ..., on doit bien avoir une application qui ...”

“Je voudrais faire une application qui utilise le capteur X pour faire Y”

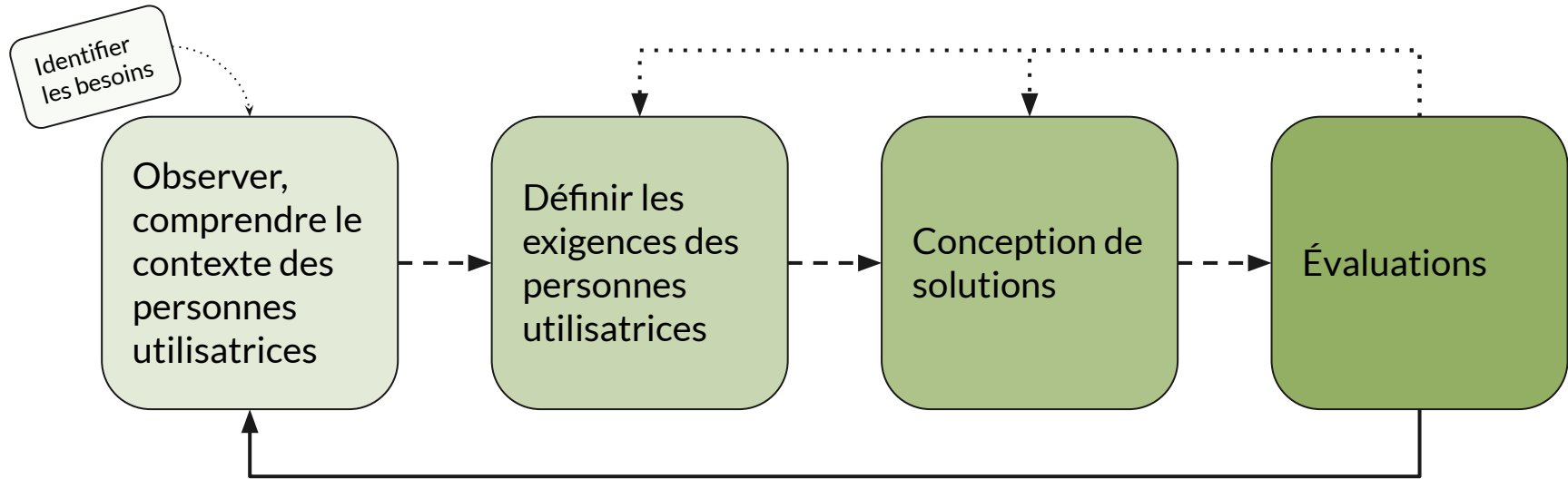
“Nous voulons faire une application qui utilise X pour soutenir les gens de la communauté Y”

Conception centrée sur la personne utilisatrice

(À l'image des personas)
Développer de l'empathie pour
la personne utilisatrice tout du
long du processus de
conception



Conception centrée sur la personne utilisatrice



Basé sur la norme de la conception centrée sur l'opérateur humain pour les systèmes interactifs ([ISO 9241-210:2019](https://www.iso.org/standard/72430.html))



Observations

- Quoi ?
(Qu'est-ce que les personnes utilisatrices font ?)
- Pourquoi ?
- Comment ?





Observations (où ? quand ?)

Contexte réel

- + Valide écologiquement
- Plus long
- Collecte plus complexe
- Saisonnalité des comportements
- Analyse plus complexe
- Plus dispendieux

Contexte artificiel

- + Rapide
- + Contrôle sur la personne utilisatrice
- + Collecte plus aisée
- + Analyse moins complexe
- + Abordable
- Validité écologique moindre

Observations (pratique - méthodes)

Directe

- + Expérience de la personne participante
- + Vision contextuelle globale
- + Adaptation
- + Richesse des données
- Long
- Dispendieux

Indirecte

- + Plus rapide
- + Moins dispendieux
- + Collecte de données sur le long terme
- Expérience de la personne participante
- Vision contextuelle globale
- Adaptation
- Richesse des données
- Risques de pertes de données
- Données inutilisables



Méthode de la pensée à voix haute

Demandez aux personnes participantes de verbaliser simplement, mais continuellement leurs pensées durant la réalisation d'une tâche

- Qu'est-ce qui attire mon attention ?
- À quoi je pense ?
- Qu'est-ce que je fais ?
- Qu'est-ce que je ressens ?

<https://www.youtube.com/watch?v=g34tOmyKaMM>



Problèmes fréquents

- Accent sur la technologie (approche technocentrée)
- Suppositions sur les besoins réels des personnes utilisatrices
- Conception égocentrique



Conception Centrée sur la personne l'Utilisatrice (CCU)

VS

Conception Centrée sur la Machine (CCM)

Point de vue	<i>Les personnes sont</i>	<i>Les machines sont</i>
Centrée sur la machine	vague	précis
	désorganisé	organisé
	distractible	imperturbable
	émotionnel	sans émotion
	illogique	logique
Centrée sur la personne	créatif	stupide
	accommodant	rigide
	attentif au changement	insensible au changement
	ingénieux	sans imagination

Concepts spéciaux

Interface truquée (dark pattern)

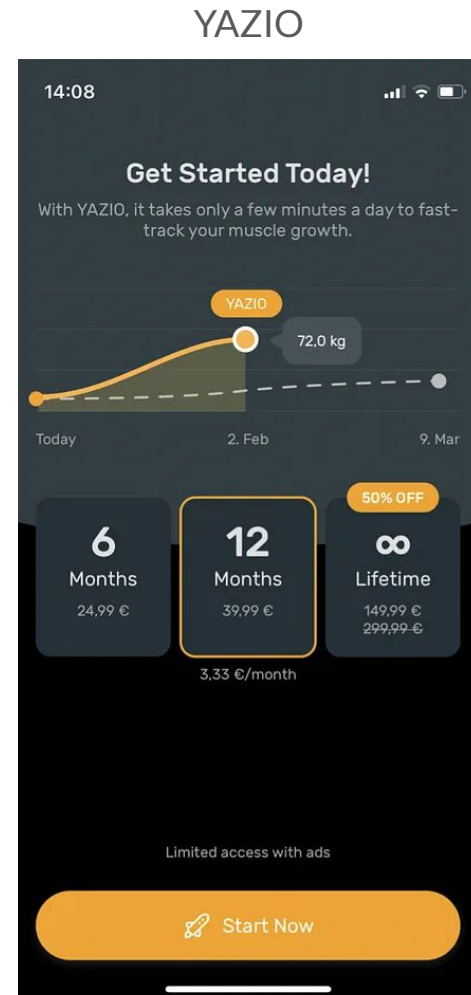
Conception universelle/accessible

Développement durable

Interface truquée (dark pattern)

Volontairement conçu pour
tromper ou manipuler une
personne utilisatrice

[UI Dark Patterns and Where to Find Them:
A Study on Mobile Applications and User Perception](https://medium.com/@marcpeter1997/dark-patterns-in-popular-mobile-apps-296011029579)





Conception universelle

La conception universelle comme gagnant-gagnant, augmente l'accessibilité tout en améliorant l'expérience d'utilisation pour toutes les personnes

L'accessibilité comme impératif juridique et **moral**

- États-Unis : Americans with Disabilities Act, article 508 du Rehabilitation Act 1973
- Diverses règles régissant le monde
- **Juste la bonne chose à faire**

[Apple Accessibility Guidelines](#)

[Material Design Accessibility](#)



Conception universelle

Une interface bien conçue pour les **personnes malvoyantes** est également bien conçue pour une utilisation :


- Dans le noir
- Quand les yeux devraient être concentrés ailleurs
- La prise en charge des lecteurs d'écran aide également les personnes à mobilité réduite grâce à la reconnaissance vocale



Conception universelle

Une interface bien conçue pour les **personnes malentendantes** est également bien conçue pour une utilisation :

- Dans les espaces bruyants
- Dans les espaces publics où le son est désactivé
- Quand les oreilles sont autrement occupées
- Locuteurs non natifs



**This interweaving of science and art
elevates the world to a place of wonder**

Sous-titre, régulièrement utilisé par les personnes sans déficience auditive



Conception universelle

Une interface bien conçue pour les **personnes ayant des déficiences cognitives** est également bien conçue pour une utilisation :

- Plus simple
- Nécessitant moins d'attention
- Plus facile à reprendre
- Plus facile d'identifier les prochaines étapes



Conception universelle

Une interface bien conçue pour les **personnes ayant un handicap moteur** est également bien conçue pour une utilisation :

- En mouvement
- Avec une panne du périphérique d'entrée
- En tant que personne utilisatrice expérimentée avec des raccourcis



Développement durable

Diminuer la consommation énergétique

- Réduire tout code inutile et redondant
- Utiliser les algorithmes et les structures de données les plus efficaces possibles
- Réduire les opérations d'entrée/sortie
- Réduire les requêtes réseau
- Optimiser les performances du code en utilisant des mécanismes de mise en cache, des opérations asynchrones et le profilage pour identifier les goulots d'étranglement

<https://www.foonkiemonkey.co.uk/sustainable-app-development-what-is-it-and-how-to-enforce-it>



Développement durable

Éviter l'OBSOLESCENCE programmée

→ Supporter les anciennes versions des systèmes d'exploitation

<https://lebondigital.com/obsolescence-programmee-un-fleau-difficile-a-eradiquer>

Kotlin

(function, variable)



Kotlin vs Java

Kotlin est le langage privilégié pour le développement Android depuis 2017, succédant à Java.

- Comme Java, il est orienté objet.
- Statiquement et fortement typé
 - Néanmoins, utilise l'inférence du type
- Plus de flexibilité que Java
 - Fonction sans classe
 - Se passer de getter et setter
 - Pas de Null pointer exception
- **Interopérabilité avec Java**



Instruction

Une **expression** est une combinaison d'opérateurs, de fonctions, de valeurs littérales, de variables ou de constantes qui produisent une valeur unique

Dans Kotlin, chaque fonction renvoie une valeur au moins Unit, donc chaque fonction est une expression

- `if` est une expression
(la plupart des structures de contrôle, sauf `for`, `do` et `do/while` sont des expressions)

En Java, une instruction se termine toujours par un point-virgule mais, en Kotlin, le point-virgule (;) est facultatif



Déclaration (statement)

- Une déclaration (statement) est l'unité syntaxique de tout langage de programmation qui exprime une action à effectuer
- Une déclaration peut contenir des expressions, mais en soi, une déclaration ne se résout pas en une valeur
- L'initialisation d'une variable est une déclaration
- L'attribution d'une valeur à une variable est une déclaration
 - `var sum = 10 + 20 //` est un statement avec `10+20` comme expression



Expression et Déclaration (statement)

Les affectations sont des expressions en **Java** mais ce sont des déclarations en **Kotlin**

Cela signifie que vous ne pouvez pas passer d'opérations d'affectation en argument à des instructions de boucle comme `while`

```
while ((rem = a % b) != 0) // ne compile pas
{
    a = b
    b = rem
}
```



Variable et constante

Mutable (modifiable)

```
var width = 12
var length: Double = 2.5
```

Immutable (inchangeable)

(assigné en cours, Runtime)

```
val name = "Jennifer"
val age: Int = 22
```

```
const val h: String = "hello"
```

(assigné à la compilation, Compile time)

Avec Kotlin, le type est inféré

```
val webView: WebView = WebView(this)
val webView = WebView(this)
```

Kotlin suit la notation Pascal, qui est, le nom suivit du type



Types de données

<https://kotlinlang.org/docs/basic-types.html>

Type	Bits	Notes
Long	64	-2^{63} to $2^{63}-1$
Int	32	-2^{31} to $2^{31}-1$
Short	16	-32768 to 32767
Byte	8	-128 to 127

Type	Bits	Notes
Double	64	16 - 17 significant digits
Float	32	6 - 7 significant digits
Char	16	16-bit Unicode character
Boolean	8	true of false



Opérateur

<https://kotlinlang.org/docs/keyword-reference.html>

Mathematical

+ - * / %

Increment/decrement

++ --

Comparison

< <= ==> >

Assignment

= +=

Equality

== != ===

Logical

&& || !

Convention (variable)

- nom commence par une lettre minuscule et camelCase
- espace après les deux-points (:)
- espace avant et après un opérateur

The diagram illustrates the convention of using spaces in variable declarations. It shows three examples of code with arrows pointing to the spaces between the variable name and the type, and between the assignment operator and the value.

```
val discount: Double = .20
```

space

```
var pet = "bird"
```

space

```
val sum = 1 + 2
```

space

Fonction

Utilisation du mot clé `fun`

Recommandation de style

- Généralement une expression verbale et camelCase
- Chaque instruction sur une ligne distincte
- Accolade ouvrante à la fin de la ligne où la fonction commence
- Un espace avant l'accolade ouvrante
- Corps de la fonction en retrait de 4 espaces (ou tabulation)
- Accolade fermante sur sa ligne et alignée avec le mot clé `fun`

```
fun name ( inputs ) {  
    body  
}  
  
fun main() {  
    println("Hello, world!")  
}
```

Fonction

```
fun printHello() {  
    println("Hello, World!")  
}
```

```
fun printHello(name: String) {  
    println("Hi $name")  
}
```

```
fun printHello(name: String): Unit {  
    println("Hi $name")  
}
```

```
fun birthdayGreeting(name: String, age: Int): String  
{  
    val nameGreeting = "Happy Birthday, $name!"  
    val ageGreeting = "You are now $age years old!"  
    return "$nameGreeting\n$ageGreeting"  
}
```

(ici, le type du paramètre est obligatoire dans la définition d'une fonction, il ne peut être inféré)

Fonction (paramètres requis et par défaut)

```
fun reformat(str: String,  
    divideByCamelHumps: Boolean,  
    wordSeparator: Char,  
    normalizeCase: Boolean = true) {
```

```
...
```

```
reformat("Today is a day like no other day", false, '_')
```

→ Arguments nommés = meilleure lisibilité, liberté de l'ordre

```
reformat(str, divideByCamelHumps = false, wordSeparator = '_')
```




Fonction

1. À l'intérieur d'une classe, comme méthodes en Java
*elles sont appelées **fonctions membres***
 2. En dehors des classes
*elles sont appelées **fonctions de niveau supérieur***
 3. Elles peuvent être écrites à l'intérieur d'autres fonctions
*elles sont appelées **fonctions locales** (peu communes avec le développement de plateforme Android, mais peuvent toujours améliorer l'encapsulation et la lisibilité du code)*
- Quel que soit l'endroit où vous placez la fonction, la déclaration est

```
fun functionName([parameters]) [type] {  
    statements  
}
```

Commentaires

```
/**
 * This program displays the number of messages
 * in the user's inbox.
 */
fun main() {
    // Create a variable for the number of unread
messages.
    var count = 10 // This is another inline comment
    println("You have $count unread messages.")

    // Decrease the number of messages by 1.
    count--
    println("You have $count unread messages.")
}
```

Deux barre oblique ou barre
oblique-étoile

```
// ...
/* ... */
```

Projets de recherche

- Keycube
 - jeu vidéo compagnon
 - surface tactile connectée
 - clavier virtuel connecté
- Véluqac
 - ludification

(Présentation)

Android Studio

Interface de l'environnement, création d'une application et émulateur

(Démonstration)

Jetpack Compose

(Classe inversée - Découverte autonome)



Préparation (obligatoire)

1. Créez un dépôt GitHub
 - **Public**
 - nommez le 8INF865A2024_InitialesPrénomNom (ex. 8INF865A2024_DB)
2. Ajouter **votre prénom et nom complet** dans un **README**, ajouter un [.gitignore pour Android](#) ET commit/push
3. Envoyer par courriel le lien de votre dépôt à dbrun@uqac.ca
4. Créer une branche **Séance2**
5. Créez votre projet Android dedans, puis réalisez les exercices, en pensant à commit/push régulièrement, **PAS uniquement au début et à la fin des exercices**



Exercices

(tutoriels officiels)

TEST d'autonomie

Mots clés

- Annotations
- Modificateurs
- Éléments de mise en page
- Recomposition

Apprentissage des base de Jetpack Compose

(à effectuer dans l'ordre, le second commence à la suite du premier)

1. <https://developer.android.com/codelabs/basic-android-kotlin-compose-text-composables?hl=fr#0>
2. <https://developer.android.com/codelabs/basic-android-kotlin-compose-add-images?hl=fr#0>

Pour la prochaine séance

- **Exercices** (tutoriels officiels) à **finir**
- **Composition des équipes de projet**
(à annoncer sur Moodle, 2 ou 3 membres)



Idées d'activités d'enrichissement personnel

- Analyser une application en identifiant des éléments pertinents aux principes de conception d'interface de Norman
- Parcourir en profondeur les lignes directrices de conception proposées par Apple et Google
- Approfondir la compréhension de la méthode de la pensée à voix haute, puis essayer de l'appliquer avec des proches en les faisant manipuler une application mobile
- Observer des « gens » qui font une tâche, ou qui sont dans un endroit particulier, puis documenter ce qu'ils font, comment ils le font, et essayer de déterminer pourquoi ils le font
- Étudier les détails de création de personas qualitatives et quantitatives