

# UE projet de développement

Licence 3 SPI, parcours Informatique

## IHM - GUI

2024-2025

Intervenants : Marie-Laure Nivet – Evelyne Vittori



# Objectifs généraux de l'UE Ingénierie logicielle (Modules Qualité - Projet - IHM)

- Développer en groupe une application logicielle de type desktop
  - Pratiquer le travail en groupe – outils et bonnes pratiques
  - Respecter les échéances
- Mettre en œuvre des principes de qualité logicielle
  - Développement itératif et incrémental
  - Séparation logique applicative/interface (architecture MVC)
  - Concevoir et implémenter des tests automatisés
- Acquérir la compréhension et la maîtrise des concepts propres à la réalisation d'IHM
  - Programmation évènementielle
- Travailler l'expression écrite et orale
  - Présentation des rendus intermédiaires
  - Reviews de code
  - Présentation/démonstration finales



# Présentation des principes du projet de développement

Ce qu'on attend de vous !

# Principes/Impératifs du projet

- Travail en binômes ou trinômes
- Application Desktop possédant une interface graphique et intéragissant avec une base de données
- Conception itérative et incrémentale de la réalisation
  - Version 1 – Interface textuelle en ligne de commandes (Command Line Interface)
  - Version 2 – Interface graphique minimalisté
  - Version 3 – Interface graphique finalisée
- Pratique de la programmation événementielle
- Utilisation **collaborative** d'un dépôt Git partagé
- Choix libre du langage :
  - Python avec Tkinter ou PyQt/PySide ou autres
  - Java avec Swing ou JavaFX
  - C# avec Windows Presentation Foundation (WPF) ou Windows Forms
  - Autre langage, tout sauf web (Javascript, HTML5)

# Déroulé

- Points cours
- Réunions/points hebdomadaires
- Soutenances intermédiaires
- Soutenance finale en amont du départ en stage

# Attendus - Evaluations

Note Suivi Projet, travail au long court (30%)

- Sérieux et implication dans le projet pendant tout son déroulé
- Régularité du travail / Nombreuses itérations
- Ponctualité des rendus
- Qualité et soin des présentations et soutenances intermédiaires
- Vie du dépôt Git
- Qualité – vie du travail collaboratif
- Prise en compte des retours

Oral et démo de présentation finales, rapport de présentation finale : 20%

Qualité – structuration du code – lisibilité du code : 20%

Qualité – IHM finale : 30%

# Objectifs – déroulé des séances

- Présentation des projets
- Enquête IHM
- Cours d'introduction
  - Rapide histoire des IHMs
  - Les différents modes d'interaction avec la machine
  - Présentation des principes de la programmation événementielle
- Choix du langage et de la librairie graphique
- Activité de découverte de la programmation événementielle
  - Programmation d'une calculatrice simplifiée
  - Présentation orale synthétique



# Avant d'aller plus loin...

Un petit questionnaire : canal “Projet programmation IHM”

Teams L3SINF

<https://forms.office.com/e/Puv06rcXY4>



IHM ? GUI ? HCI ? Vers des RHM ?



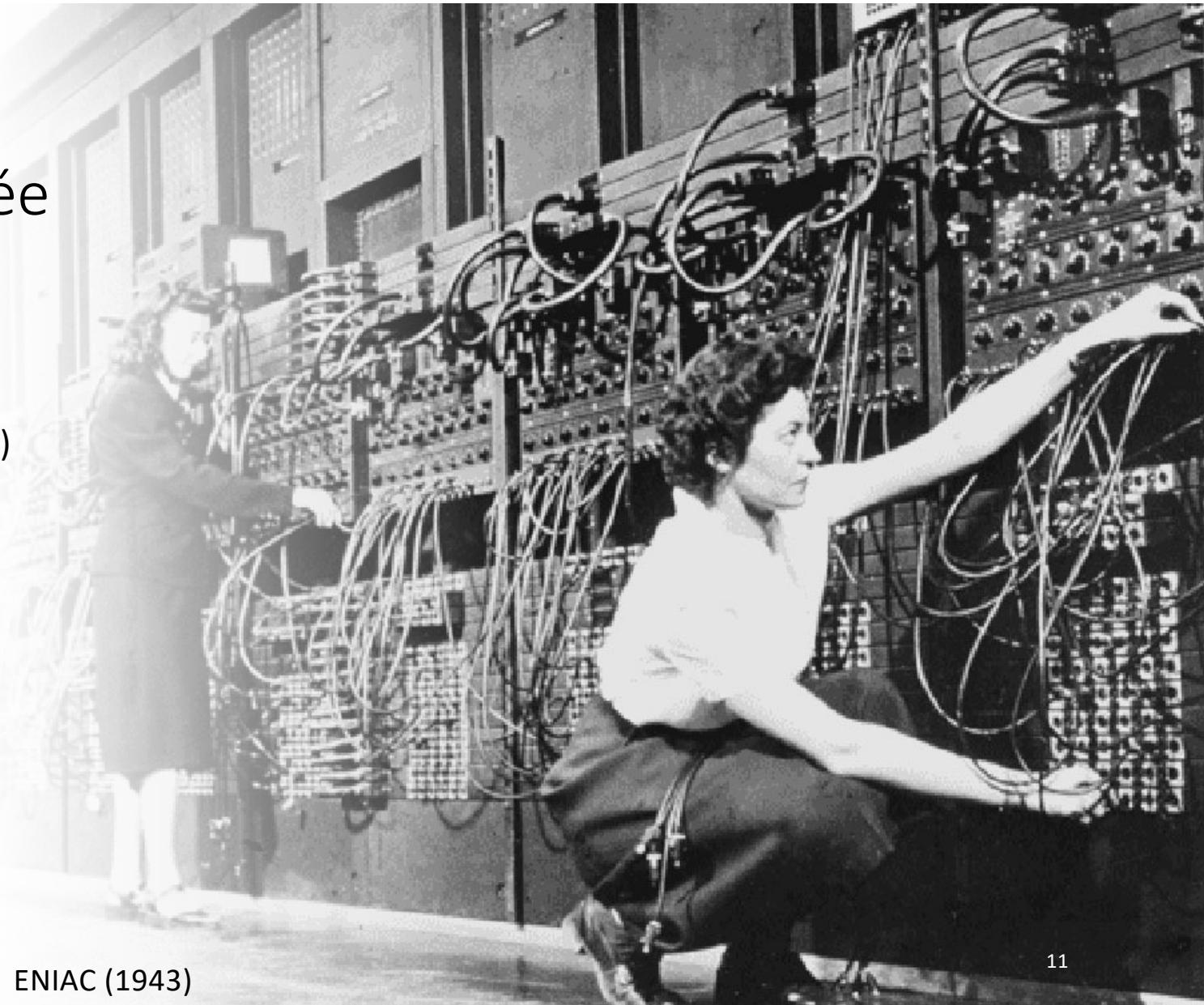
# Historique des Interfaces Homme Machine

... Plus un petit regard vers le futur

Pour le passé : Computer History Archives Project (CHAP)  
<https://www.youtube.com/@ComputerHistoryArchivesProject/videos>

# Les débuts, une interaction limitée 1945 - 1965

- Fils/Jacks/interrupteurs
- Cartes perforées (INA 1960)
- Voyants, cadrant
- Imprimantes



# L'arrivée des écrans/ claviers et les expériences de dispositifs de pointage



Sketchpad Demo 1963

[https://www.youtube.com/watch?v=6orsmFndx\\_o](https://www.youtube.com/watch?v=6orsmFndx_o)

Ivan Sutherland's (1938 - ) and Tim Johnson's (à l'image) Sketchpad (1963) 2 of 3 - 2D Graphics (9m56)

<https://www.youtube.com/watch?v=hB3jQKGrJo0>

Presentation du Sketchpad dans le cadre de la thèse de Sutherland MIT 1963 (16mn41)

<https://www.youtube.com/watch?v=57wj8diYpgY>

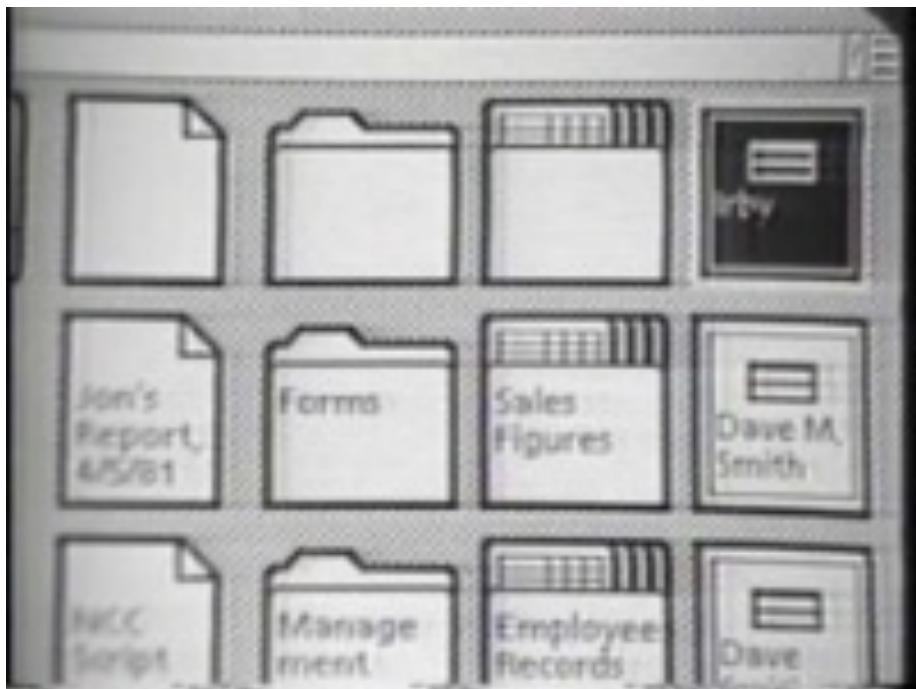
# Les prémisses : Douglas Engelbart 1968

## La première souris



The Mother of All Demos, presented by Douglas Engelbart (1968)  
<https://www.youtube.com/watch?v=yJDv-zdhzMY>

# Xerox et la première GUI (années 1980)



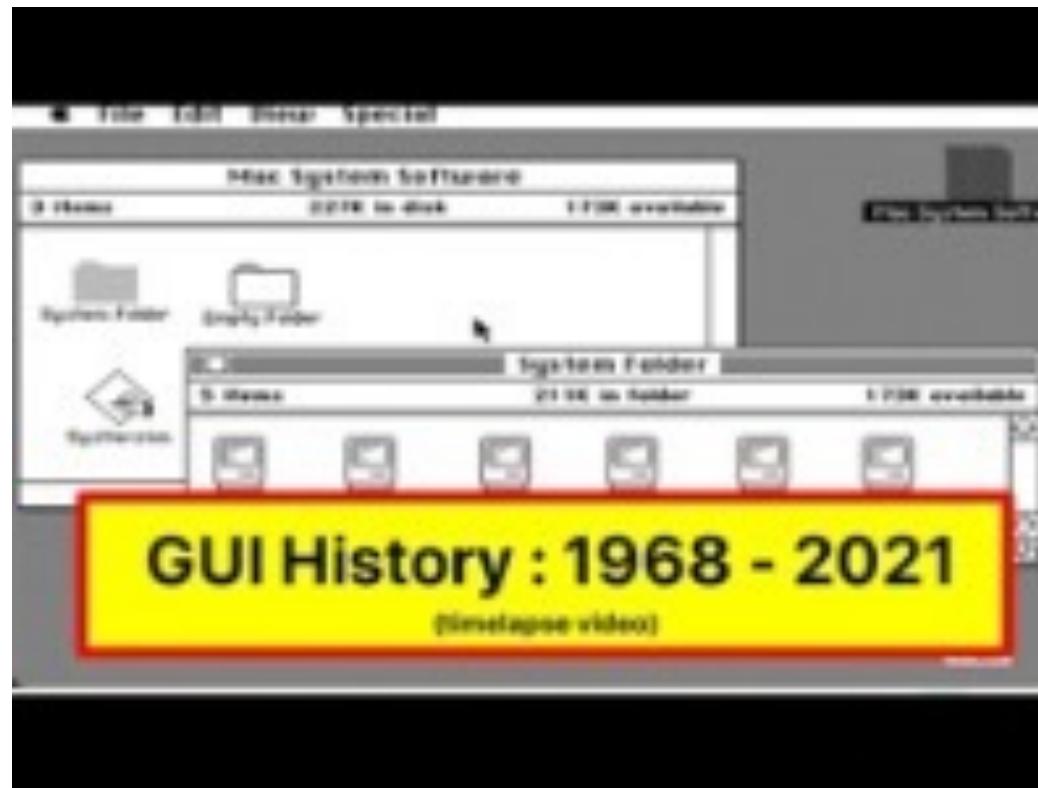
**Xerox Star User Interface (1982) 1 of 2**

<https://www.youtube.com/watch?v=Cn4vC80Pv6Q>

**How Steve Jobs got the ideas of GUI from XEROX**

<https://www.youtube.com/watch?v=J33pVRdxWbw> (7mn00)

# History of GUI (1968-2020) | Apple | Microsoft | Xerox & More



Aman Jat, <https://www.youtube.com/watch?v=nW5v5q9Nm6A>, 4m24

# Les différents types d'interfaces hommes machines

De l'interface textuelle, aux interfaces cerveau ordinateur en passant  
par les ...

# Interfaces textuelles, *Command Line-oriented Interfaces (CLI)*

- Interfaces à ligne de commandes, *Command Line Interface*
  - Dialogue questions/réponses fortement structuré
  - Construit sur des langages de commandes
  - Effort de mémorisation important
  - Saisie par le clavier (*character-oriented interface*)

Version 0 de votre application,  
conception de la logique applicative

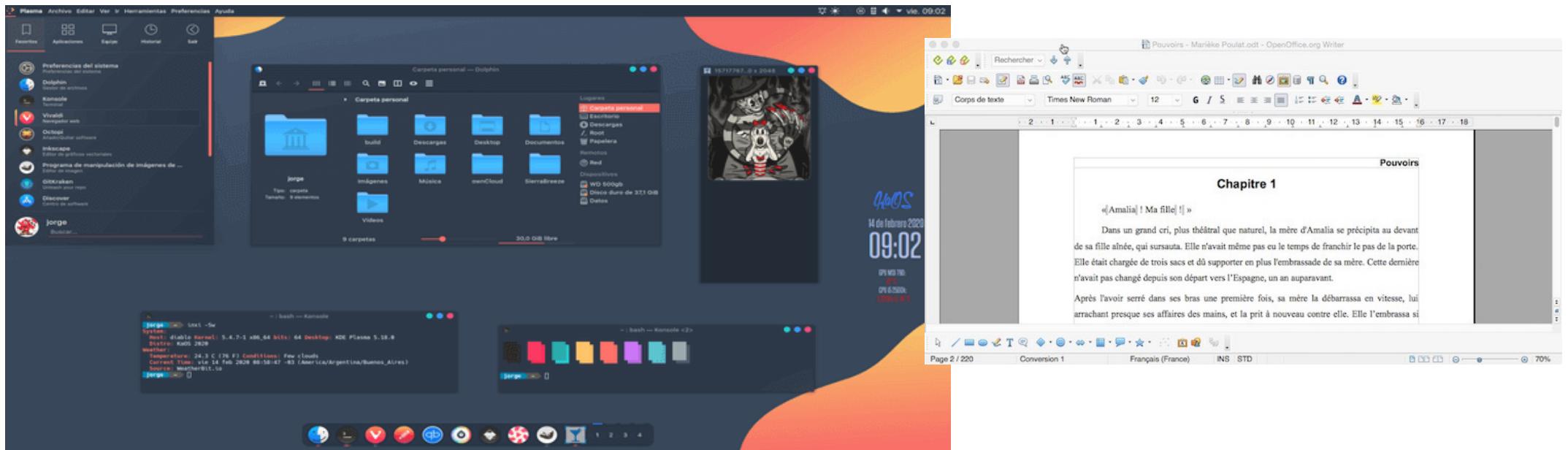
```
[root@localhost ~]# ping -q fo.wikipedia.org
PING text.pmpfa.wikimedia.org (200.80.152.2) 56(84) bytes of data.
...  
--- text.pmpfa.wikimedia.org ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms  
[root@localhost ~]# pwd  
/root  
[root@localhost var]# cd /var  
[root@localhost var]# ls -la  
total 72  
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .  
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..  
drwxr-xr-x. 2 root root 4096 May 14 00:15 account  
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache  
drwxr-xr-x. 3 root root 4096 May 18 16:03 db  
drwxr-xr-x. 3 root root 4096 May 18 16:03 empty  
drwxr-xr-x. 2 root root 4096 May 18 16:03 games  
drwxrwx--T. 2 root gdm 4096 Jun 2 18:39 gdm  
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib  
drwxr-xr-x. 2 root root 4096 May 18 16:03 local  
lrwxrwxrwx. 1 root root 11 May 14 00:12 lock -> ../run/lock  
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log  
lrwxrwxrwx. 1 root root 10 Jul 30 22:43 mail -> spool/mail  
drwxr-xr-x. 2 root root 4096 May 18 16:03 nis  
drwxr-xr-x. 2 root root 4096 May 18 16:03 opt  
drwxr-xr-x. 2 root root 4096 May 18 16:03 preserve  
drwxr-xr-x. 2 root root 4096 Jul 1 22:11 report  
lrwxrwxrwx. 1 root root 6 May 14 00:12 run -> ../run  
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool  
drwxrwxrwt. 4 root root 4096 Sep 12 23:58 tmp  
drwxr-xr-x. 2 root root 4096 May 18 16:03 yp  
[root@localhost var]# yum search wiki  
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves  
rpmlfusion-free-updates 2.7 kB 00:00  
rpmlfusion-free-updates/primary_db 206 kB 00:04  
rpmlfusion-nonfree-updates 2.7 kB 00:00  
updates/metalink 5.9 kB 00:00  
updates 4.7 kB 00:00  
updates/primary_db 73% [=====] 62 kB/s 2.6 MB 00:15 ETA
```

- Toujours très efficaces dans le cadre d'un usage intensif (rapidité)
- Autocad Raccourcis/langage de commande

# Les interfaces graphiques, Graphical User Interfaces (GUI)

Version 1 et 2 de votre application,  
interfaçage de la logique applicative  
démontrée en version 0

- Actions directes sur les objets apparaissant à l'écran
- Concept « What You See Is What You Get » (WYSIWYG)



# Les interfaces intuitives, Natural User Interfaces (NUI)

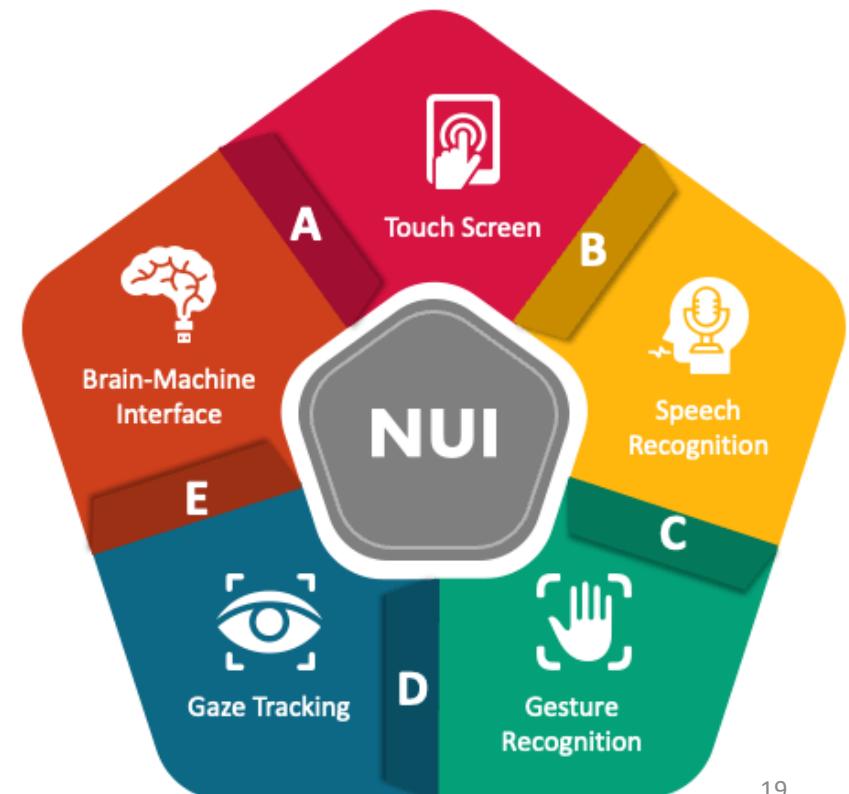
- Interaction naturelle, prise en mains directe.
- Commandes vocales, gestuelles, multimodales, pensée... !.
- Generative AI



Voice User Interfaces

## NATURAL USER INTERFACE (NUI)

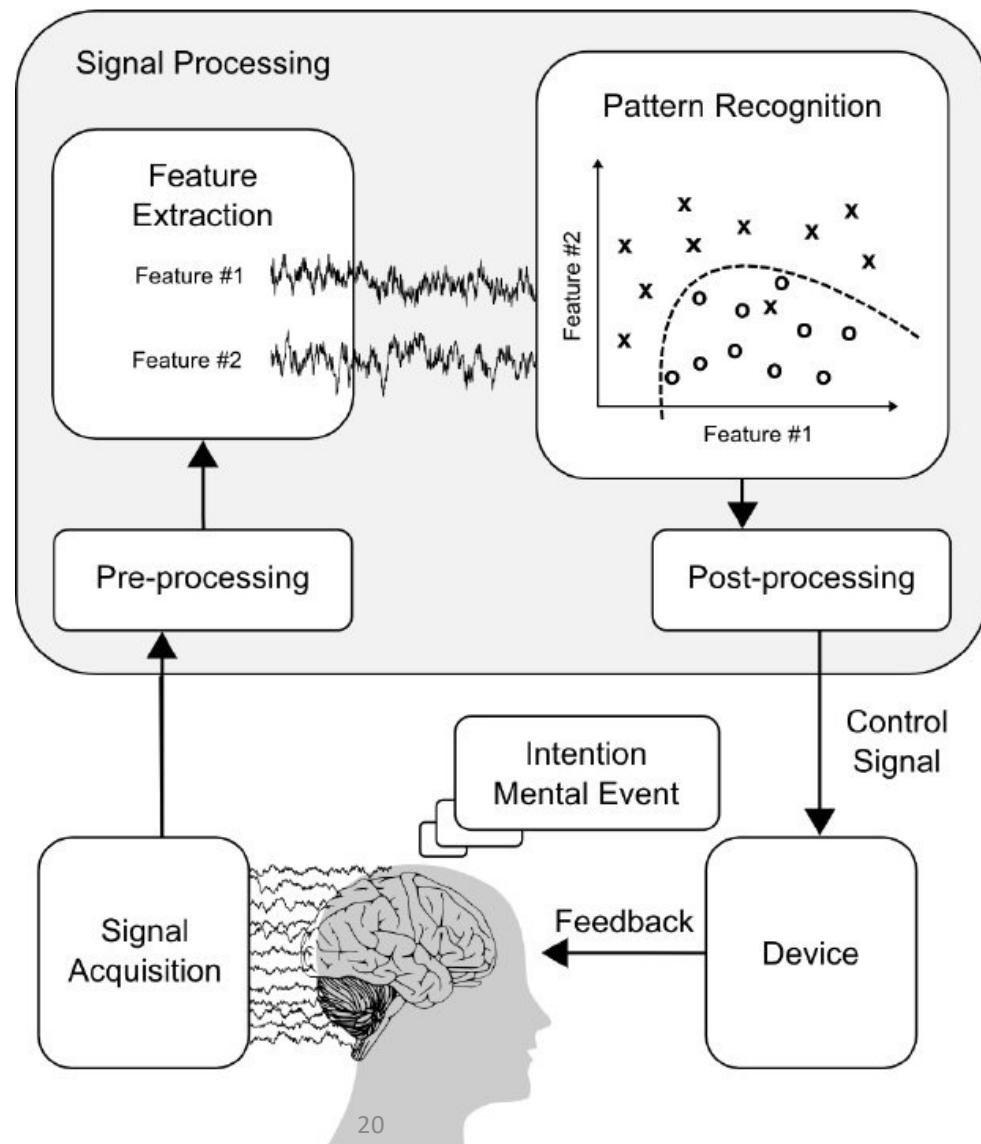
Elements of Natural User Interface



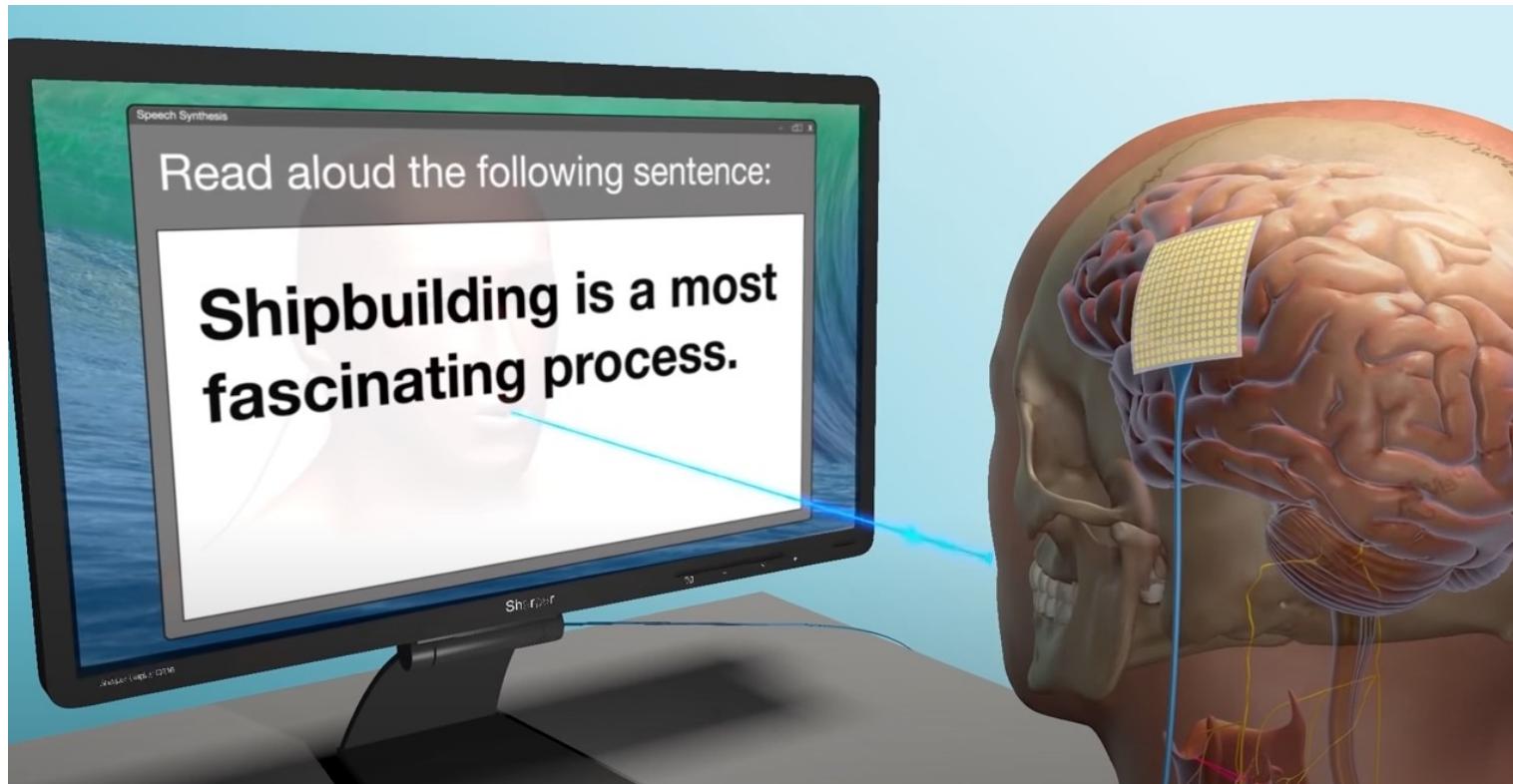
<https://www.sketchbubble.com/en/presentation-natural-user-interface-nui.html>

# Interfaces cerveau ordinateurs (ICO, ICM), *Brain Computer Interfaces* (BCI)

- Interface neuronale directe (IND)
- Utilisation de l'activité électrique générée par le cerveau lors de l'exécution de certaines activités ou de la pensée de l'exécution de ces activités



# Synthèse de la voix à partir de la reconnaissance des mouvements de la face



Synthetic speech generated from brain recordings, Research published in Nature on April 24th, 2019 Gopala K. Anumanchipalli, Josh Chartier and Edward F. Chang (<https://www.nature.com/articles/s41586-019-1119-1>)  
[https://www.youtube.com/watch?v=3pv0vT82Cys&ab\\_channel=UCSFNeurosurgery](https://www.youtube.com/watch?v=3pv0vT82Cys&ab_channel=UCSFNeurosurgery)

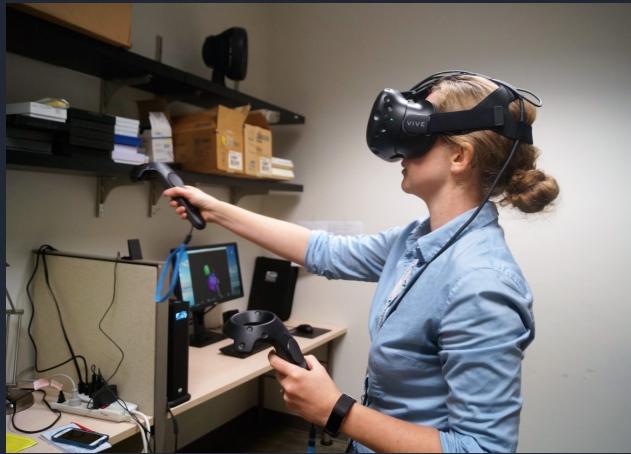
# BCI



Brain Computer Interface (VOSTF) INRIA, 2017, 7m23  
<https://www.youtube.com/watch?v=IMpv4gm1Pfo>



Augmentée : superposition de la réalité et d'éléments calculés par un système informatique en temps réel.



Virtuelle : simule la présence physique d'un utilisateur dans un environnement numérique



Mixte : combinaison d'objets du monde réel et d'objets virtuels

# Immersion, XR, *Extended Reality*

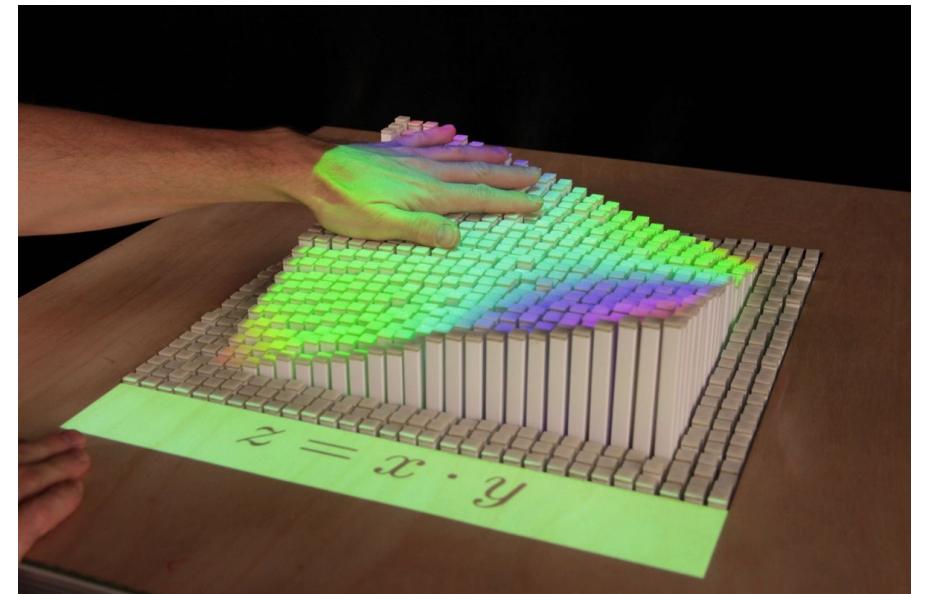
- Réalité virtuelle
- Réalité augmentée
- Réalité mixte

# Interfaces tangibles, Tangible User Interfaces (TUI)

- Interaction avec un ou des objets tangibles, réels qui permettent d'agir sur des objets numériques



En entrée



En sortie



# Les interfaces du futur ?

Creusez un peu et étonnez-vous/nous !

# NOUS SOMMES PASSÉS DE L'ÈRE DES IHM À L'ÈRE DES RELATIONS HOMMES-MACHINES

“



# Fondamentaux des interfaces homme machine

# Principaux composants d'une interface graphique Application Desktop

- Un pointeur/curseur utilisé pour la sélection et la manipulation
  - Flèche en mode graphique,
  - Curseur clignotant en mode textuel
- Un dispositif de pointage
  - Track ball
  - Souris
- Des icônes
- Un bureau
- Des fenêtres
- Des menus

# Qu'est-ce qu'une interface homme machine ?

- Ensemble des dispositifs matériels et logiciels qui permettent à un utilisateur humain d'interagir avec un système numérique interactif



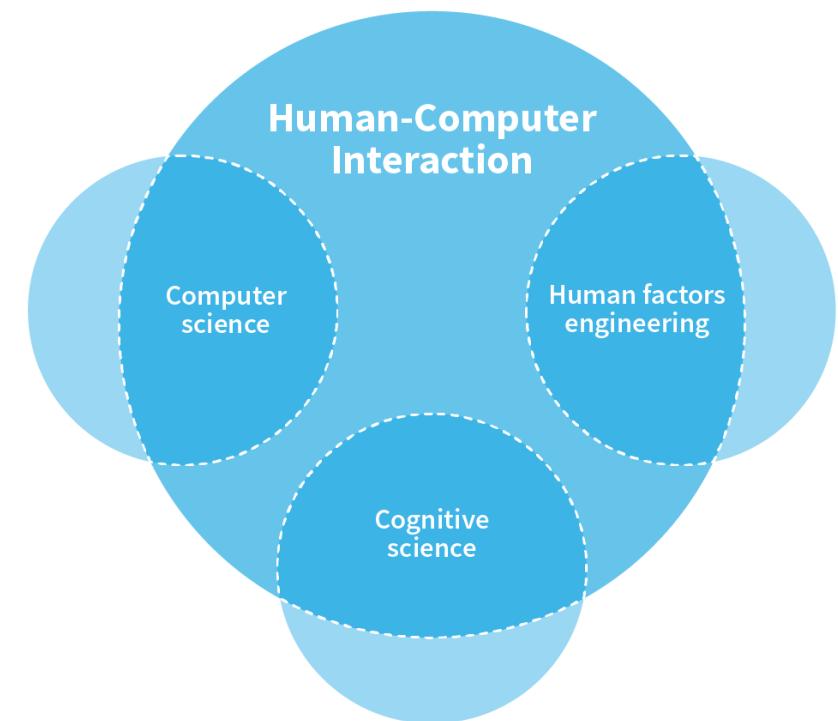
# Une science par excellence multi-disciplinaire

Lorsque l'on conçoit une Interface Homme Machine (IHM) il ne faut pas oublier que le partenaire principal est l'homme et non la machine !

Les domaines entrant en ligne de compte

- Informatique : la machine, le langage, l'algo
- Sciences cognitives : prise en compte de l'être humain, de ses biais
- Ergonomie : l'interaction entre l'homme et la machine
- Sociologie, psychologie : les émotions participent de notre réceptivité.

## The Multidisciplinary Field of HCI

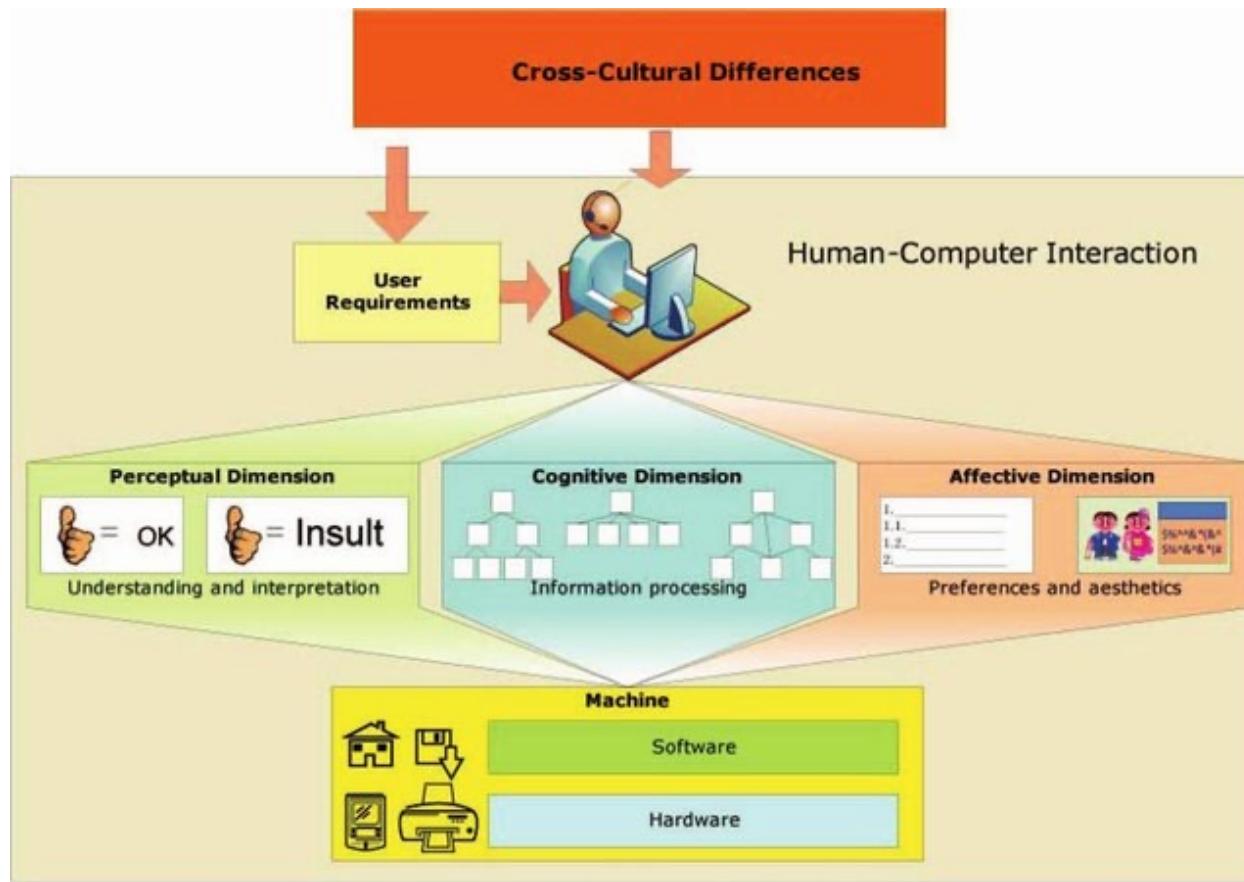


Interaction Design Foundation  
interaction-design.org

# Les participants à la réalisation d'une interface

- Le futur utilisateur
  - Il expose ses besoins, ses choix
- La machine et plus particulièrement le programme à interfacer
- Le concepteur de l'interface
  - Censé coder – et donc connaître ! – les besoins et les choix de l'utilisateur

# Les principes généraux de l'interaction homme machine



32

# Adaptation de l'IHM

Prendre en compte :

- Les spécificités physiques (+ accessibilité)
- L'expertise métier, l'expertise niveau manipulation
- Les potentielles spécificités culturelles
  - Formats de dates, de nombres
  - Signification des couleurs, de l'icônerie
  - Sens d'écriture
- Le contexte (grand public, industrie, loisir, système critique)
- Les contraintes techniques
  - Plateformes, Compatibilité Legacy, bande passante, mémoire, écran, etc.

## B.A. B.A.

- Ne pas chercher à être avant tout innovant, nouveau, joli, cool...
- Faire simple mais clair !
- Utiliser les conventions propres au système auquel est destiné l'interface
- Avant d'imposer votre interface aux autres, regarder la bien et tester la vraiment...

# Quelques conseils...

Des règles à respecter

# Restez cohérents

- Cohérence externe
  - dans le choix des symboles pour représenter les choses. Ex: La poubelle pour jeter les documents que l'on ne veut plus, rouge pour attention, vert pour allez-y.
  - choisir des termes, des images qui désignent bien ce qu'on veut dire en évitant les termes généraux ou ambigus que l'usager prendra du temps à interpréter dans le contexte.
- Cohérence interne
  - les symboles , les dispositions doivent être constants.
  - réutiliser le même mot pour désigner la même chose.

# Adaptabilité à l'utilisateur

- Prévoir plusieurs niveaux d'utilisation
  - Débutants (Menus, barres d'outils)
  - Confirmés (raccourcis, personnalisation)
- Prévoir plusieurs niveaux de transparence
  - Cacher les mécanismes internes de l'appli
  - Permettre l'administration et la mise au point

# Rétro-action (feedback)

- Visualisation de l'effet des actions
  - Le logiciel doit réagir pour montrer l'effet d'une action.
- Dans l'absence de rétroaction, l'usager ne sait pas s'il a bien choisi, ce qu'il a choisi, si la commande qu'il utilise est possible...
- Le système doit réagir de façon continue à ce que fait l'usager en l'informant des choix faits, des processus en cours.
  - Rapide et non ambiguë
    - Modification du curseur de la souris
    - Sélection en vidéo inverse
    - Boutons enfoncés

# Avertissement de fin de tâche

- Fin de tâche de fond
  - Envoi de mail, de fax
  - Fin d'impression
- Fin de succession de tâches atomiques

# Gestion des erreurs

- Types d'erreurs
  - Apprentissage empirique (droit à l'essai)
  - Erreur physique
  - Erreur logique ou d'interprétation
- Prévention des erreurs
  - Adapter l'affichage au contexte
    - Vérifier qu'une entrée est valide
    - Proposer des corrections
    - Rendre inactifs les items de menus interdits
  - Design (Taille et choix des icônes)
  - Bulles d'aide
- Traitement des erreurs

# Messages d'erreurs

- Spécifiques et précis
  - Éviter les « Syntax error N° »
  - Préférer « Parenthèse manquante ligne 10 »
- Constructifs, indiquer les corrections possibles
- Positifs
  - éviter la condamnation : ILLEGAL, INVALIDE

# Réversibilité des actions

- Possibilité de "défaire" ce qui a été fait : *undo*
  - Boutons annuler dans les boîtes de dialogue
  - Fonction d'annulation des actions
- Importante pour que l'usager n'ait pas peur d'essayer les fonctionnalités.
- Favorise l'exploration, l'essai erreur et permet à l'usager d'acquérir de l'expertise.
- La principale limite à l'utilisation des systèmes vient souvent du fait que les usagers n'ont pas assez exploré un logiciel pour en connaître tout le potentiel.

# Laisser la maîtrise à l'utilisateur

- L'utilisateur pilote l'application
- Limiter les situations de guidage total
- Préférer le droit à l'erreur
- Meilleure appropriation

# Ne tomber pas dans la tentation des Dark Pattern



How Dark Patterns Trick You Online, The Nerdwriter, 2018, 6m56  
<https://www.youtube.com/watch?v=kxkrdLI6e6M>

Dark Pattern : quand l'expérience utilisateur devient un enfer, 2019, Marie Directrice artistique, Novaway, UX/UI  
<https://www.novaway.fr/blog/ui-ux-design/dark-pattern-experience-utilisateur>

# Des initiatives de la tech

« THE CRITICAL ENGINEERING MANIFESTO »

<https://criticalengineering.org/fr>

al | ca | cn | de | dk | en | es | fi | fr | gl | gr | he | it | jp | nl | pt | ru | sl | sr



The Critical Engineering Working Group  
Berlin, October 2011–2021

Julian Oliver  
Gordan Savicic  
Danja Vasiliev

THE CRITICAL ENGINEERING MANIFESTO

0. L'ingénieur critique considère l'ingénierie comme le langage le plus percutant de notre époque modélisant nos manières de nous déplacer, communiquer et penser. Le rôle de l'ingénieur critique est d'étudier et d'exploiter ce langage – de montrer son influence.

1. L'ingénieur critique considère notre dépendance à la technologie comme un défi autant qu'une menace. Plus la dépendance est grande plus il importe d'en analyser le fonctionnement interne indépendamment des droits de propriétés et dispositions légales.

2. L'ingénieur est conscient que chaque avancée technologique met à mal notre héritage techno-politique.

3. L'ingénieur critique déconstruit et incite à la méfiance face aux expériences utilisateurs luxuriantes.

4. L'ingénieur critique ne se laisse pas impressionner par les contraintes de l'implémentation. Il en explore les méthodes d'influence et leurs effets spécifiques.

5. L'ingénieur critique reconnaît que chaque travail d'ingénierie formate l'utilisateur proportionnellement à la dépendance de ce dernier au travail d'ingénierie.

6. En décrivant les interrelations des dispositifs, corps, agents et forces des réseaux, l'ingénieur critique élargit le concept de 'machine'.

7. L'ingénieur critique observe l'espace entre la production et la consommation des technologies. En agissant rapidement, l'ingénieur critique peut déstabiliser cet espace provoquant des moments de déséquilibre et de déception.

8. L'ingénieur critique se tourne vers l'histoire de l'art, l'architecture, l'activisme, la philosophie et l'innovation pour y trouver des travaux exemplaires d'ingénierie critique. Les stratégies, idées, agendas, de ces disciplines seront adoptées, re-agencées et développées.

9. L'ingénieur critique constate que le code écrit s'immisce dans les secteurs du social et du psychologique régulant les comportements entre les personnes et les machines avec lesquelles elles interagissent. Prendant conscience de ce fait, l'ingénieur critique cherche à reconstruire les contraintes et actions sociales des utilisateurs en procédant à des fouilles numériques.

10. L'ingénieur critique considère l'utilisation de la vulnérabilité d'un système comme la forme la plus souhaitable de dénonciation.

Translation into French by [Benedicte Jacobs](#).

# Fondamentaux de la programmation évenementielle

Spécificités

# Spécificités de la programmation événementielle par rapport à la programmation impérative classique

Impératif : Le flot de contrôle est prévisible, on connaît plus ou moins l'ordre d'exécution des instructions.

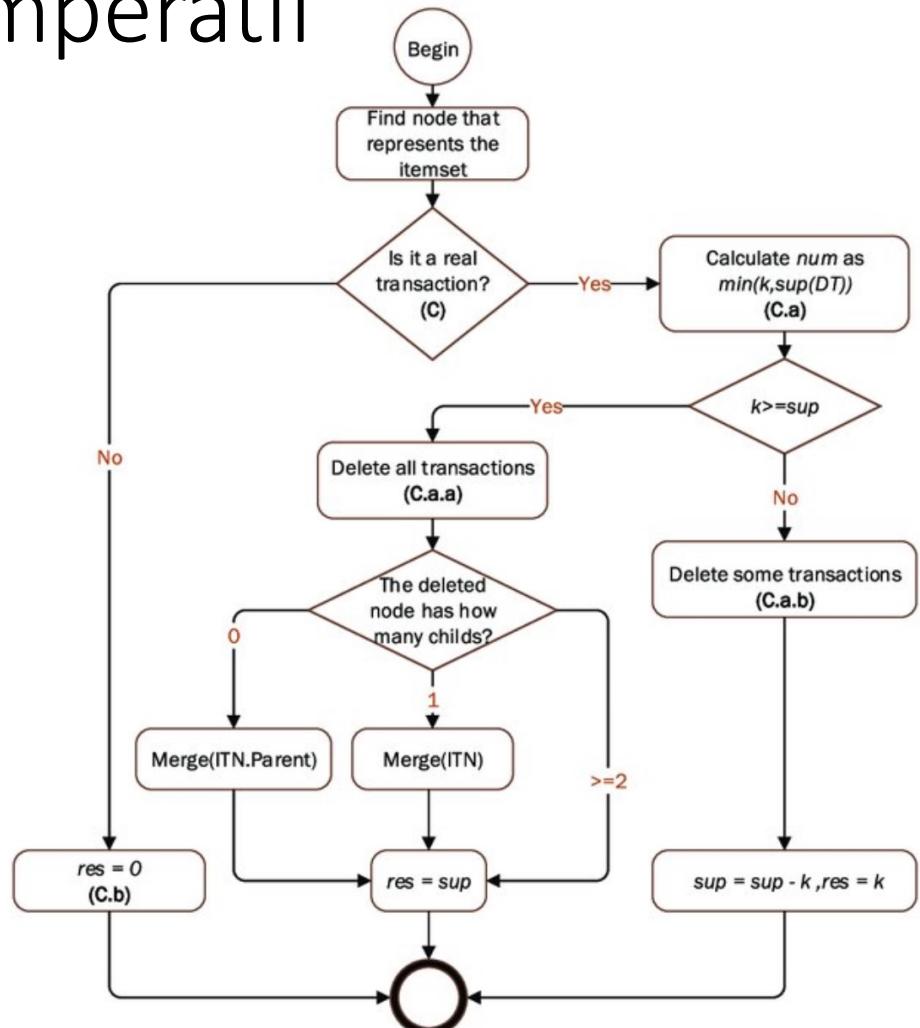
Évenementiel : Le flot de contrôle est imprévisible car il dépend de l'arrivée d'événements externes générés par une entité indépendante du programme.

# Flot de contrôle dans le cas impératif

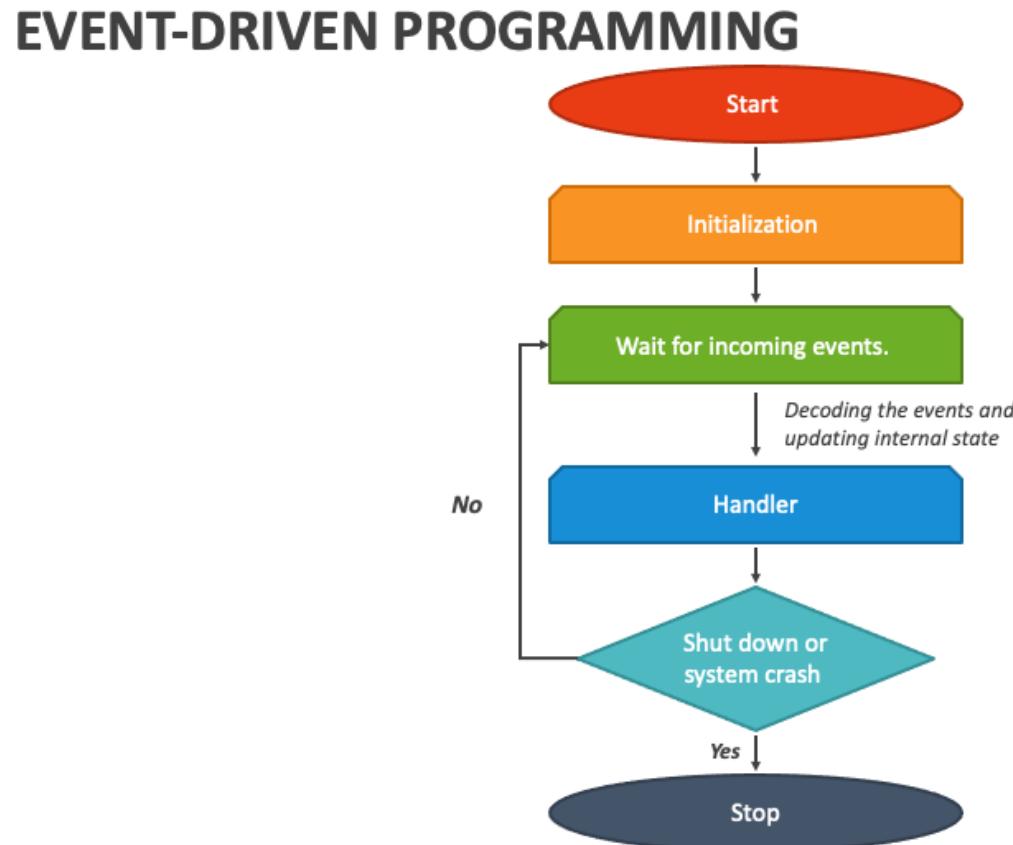
```

main(args){
    step_1
    step_2
    if cond_1 {
        step_3
        for ... {
            step_4i
        }
        step_5
    }
    else{
        step_6
        ...
    }
}

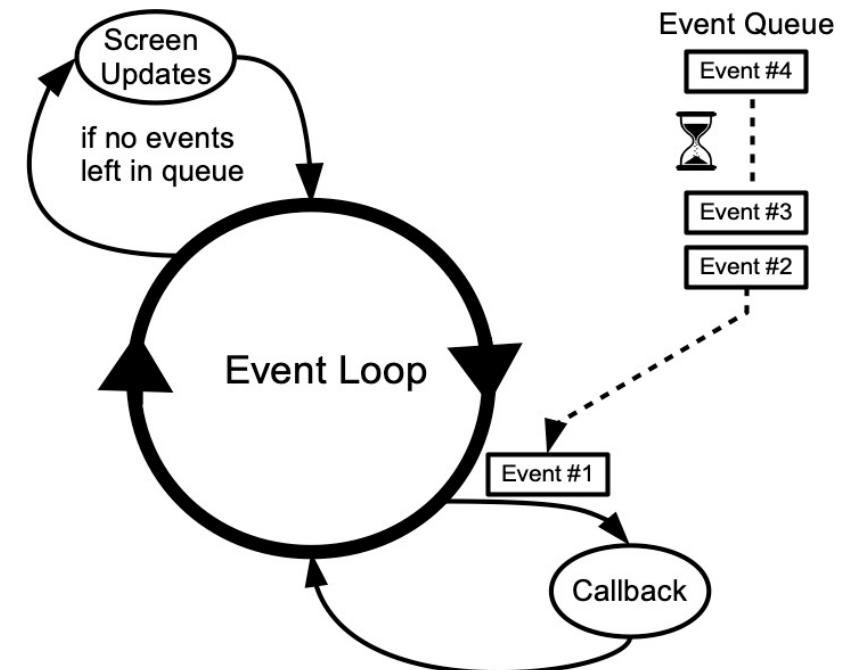
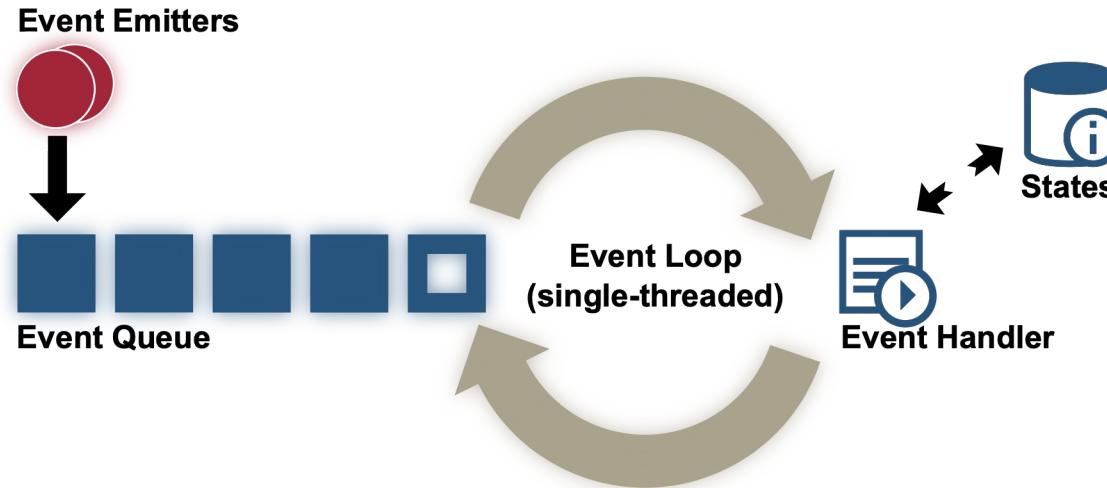
```



# Flot de contrôle dans le cas événementiel

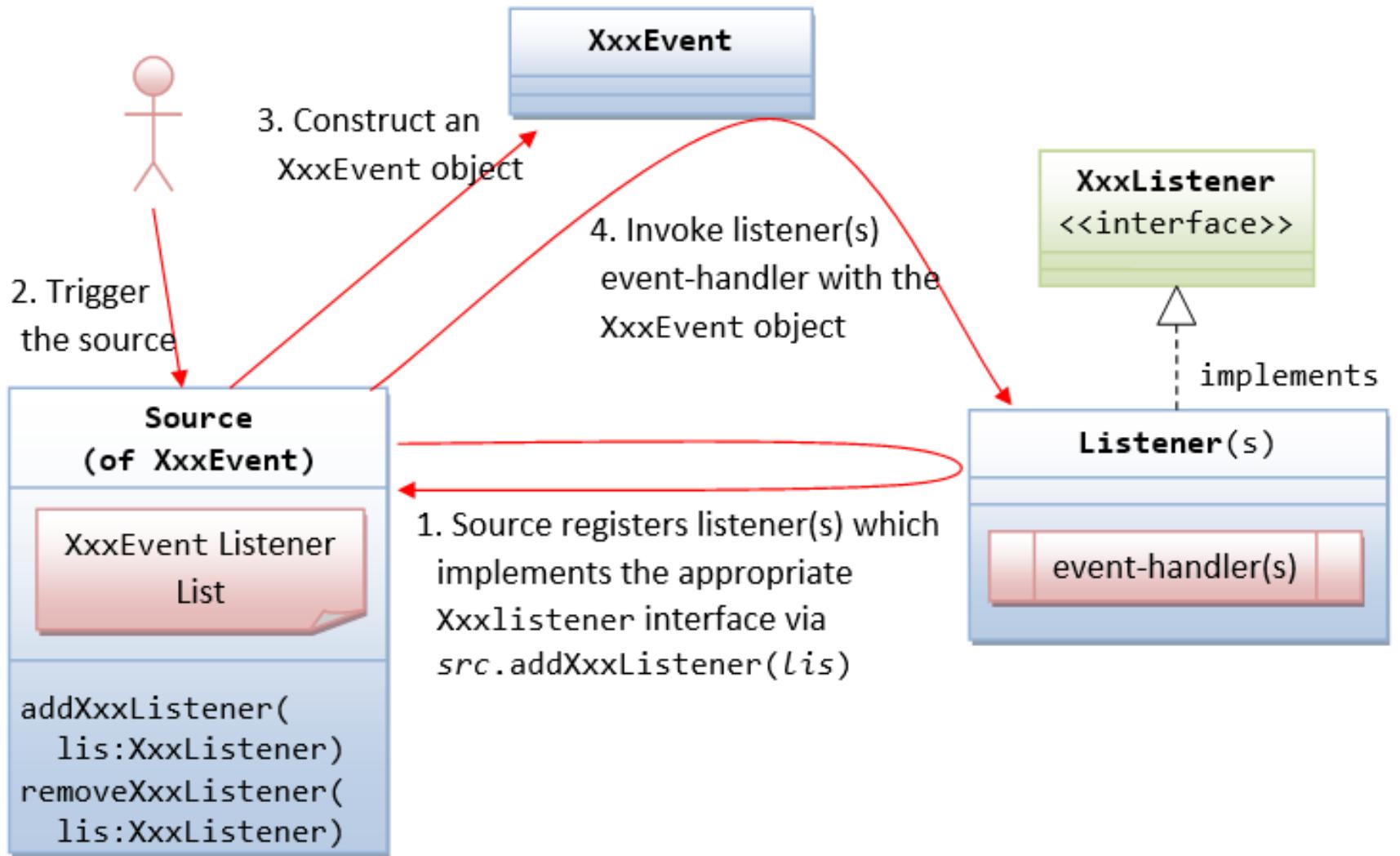


# La boucle infinie événementielle



Explain Event Loop,  
<https://gist.github.com/kassane/f2330ef44b070f4a5fa9d59c770f68e9>

Tutorial Tkinter (Tool Kit Interface), adaptation de Tk (bibliothèque d'interfaces graphiques multiplateforme)  
[https://profjahier.github.io/html/NSI/tkinter/doc\\_tk\\_allegee/tutorial/eventloop.html](https://profjahier.github.io/html/NSI/tkinter/doc_tk_allegee/tutorial/eventloop.html)



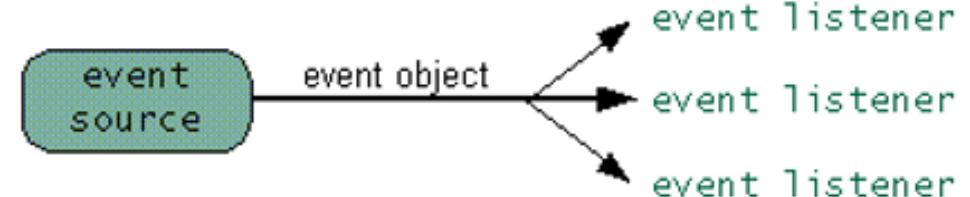
# Les différents composants

La source de l'événement : bouton, click souris, redimensionnement fenêtre, etc.

L'événement : un objet contenant les caractéristiques de l'événement (x,y du pointeur de souris, état des boutons, touche enfoncée, etc.) et de sa source.

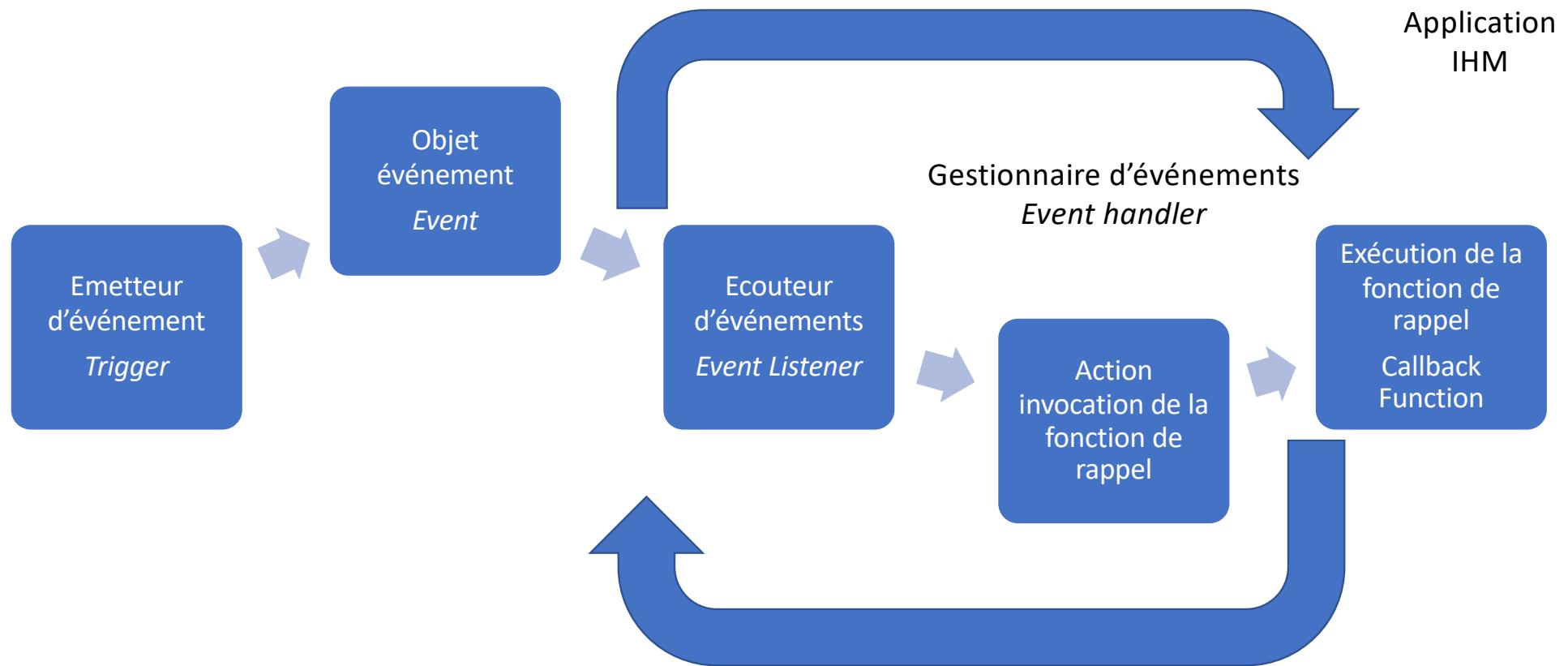
Le récepteur ou l'écouteur de l'événement qui s'est enregistré en tant que tel auprès de la source.

Le composant source génère un événement  
L'écouteur le réceptionne et le traite



Un même composant  
peut avoir plusieurs écouteurs,  
un par type d'événement

# Modèle de la programmation événementielle



Boucle événementielle : toujours à l'écoute des événements

# Différents types d'événements

- Bloquants
- Non bloquants

# Boucle événementielle

```
while True:  
    # Get the next event from the operating system  
    event = get_next_event()  
  
    # Get the function that is assigned to handle this event  
    a_function_to_handle_the_event = event-handlers[event]  
  
    # If a function has been assigned to handle this event, call the function  
    if a_function_to_handle_the_event:  
        a_function_to_handle_the_event() # Call the event-handler function  
  
    # Stop processing events if the user gives a command to stop the application  
    if window_needs_to_close:  
        break # out of the event-loop
```



## Différents outils pour le design/prototypage d'interface homme machines

---

Figma, Balsamiq, Sketch, Canva, etc.

# A vous de jouer !

- Choisissez le langage et la librairie graphique pour votre projet.
- Entraînez-vous à sa manipulation en programmant une calculatrice simplifiée avec les chiffres 1, 2, 3 et 4 et les opérateurs + et -.
- Préparez une présentation synthétique d'environ 5 diapositives présentant concrètement les étapes de la réalisation de votre calculatrice.