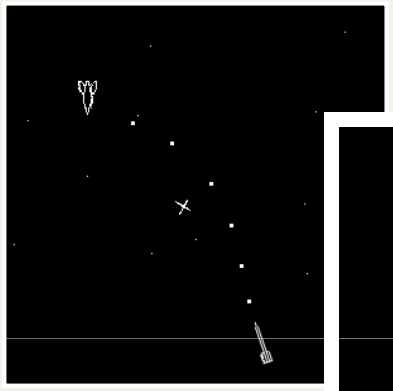
ECOLE NATIONALE
DU JEU ET DES MEDIAS
INTERACTIFS NUMERIQUES

Technologies et Métiers des loisirs numériques : interfaces humain-machine, 3D et multimédia (et JV)

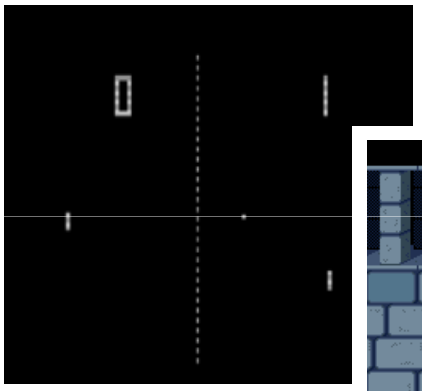
Alexandre Topol

ENJMIN / CEDRIC
Conservatoire National des Arts & Métiers

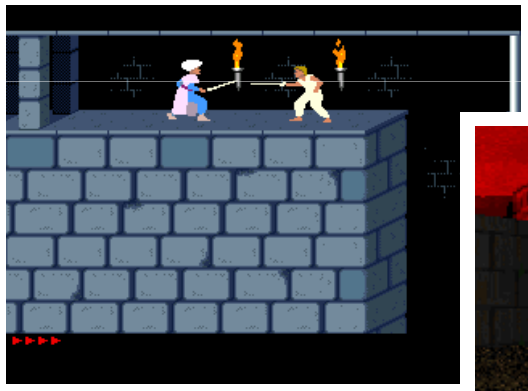
20 ans d'évolution ...



1961
Spacewar!
Steve Russel
(MIT)



1972
PONG
Nolan Bushnell
(Atari)



1989
Prince of Persia
Jordan Mechner
(Brøderbund)

1993
Doom
John Carmack
(Id Software)

ECOLE NATIONALE
DU **JEU** ET DES **MEDIAS**
INTERACTIFS NUMERIQUES

… DES JEUX …

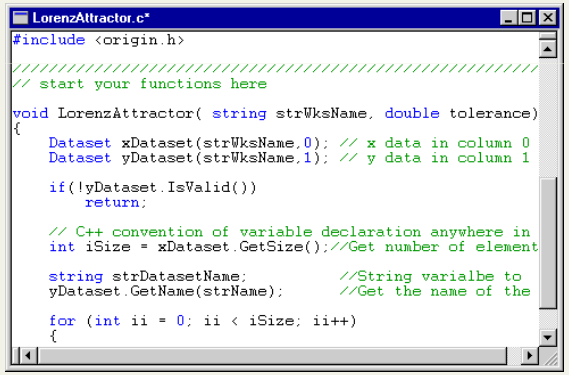
1997
Quake 2
Id Software
(Activision)

2004
Far Cry
Crytec
(UbiSoft)

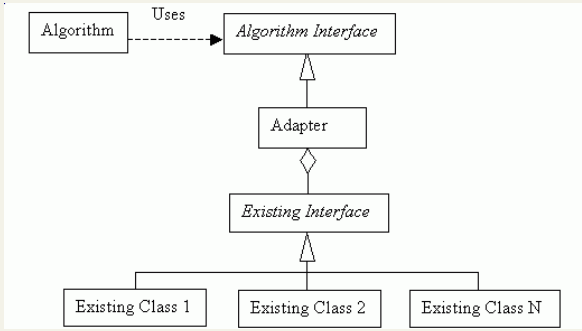
2008
Assasin's Creed
(UbiSoft)

... DES TECHNIQUES ...

1960
ASM
BASIC



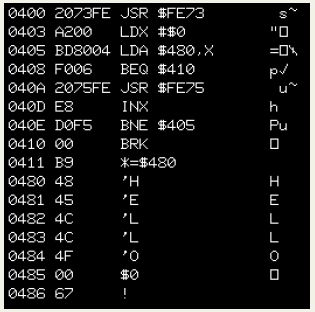
1972
C



1983
C++

… DES MÉTHODES …

1993
Mods

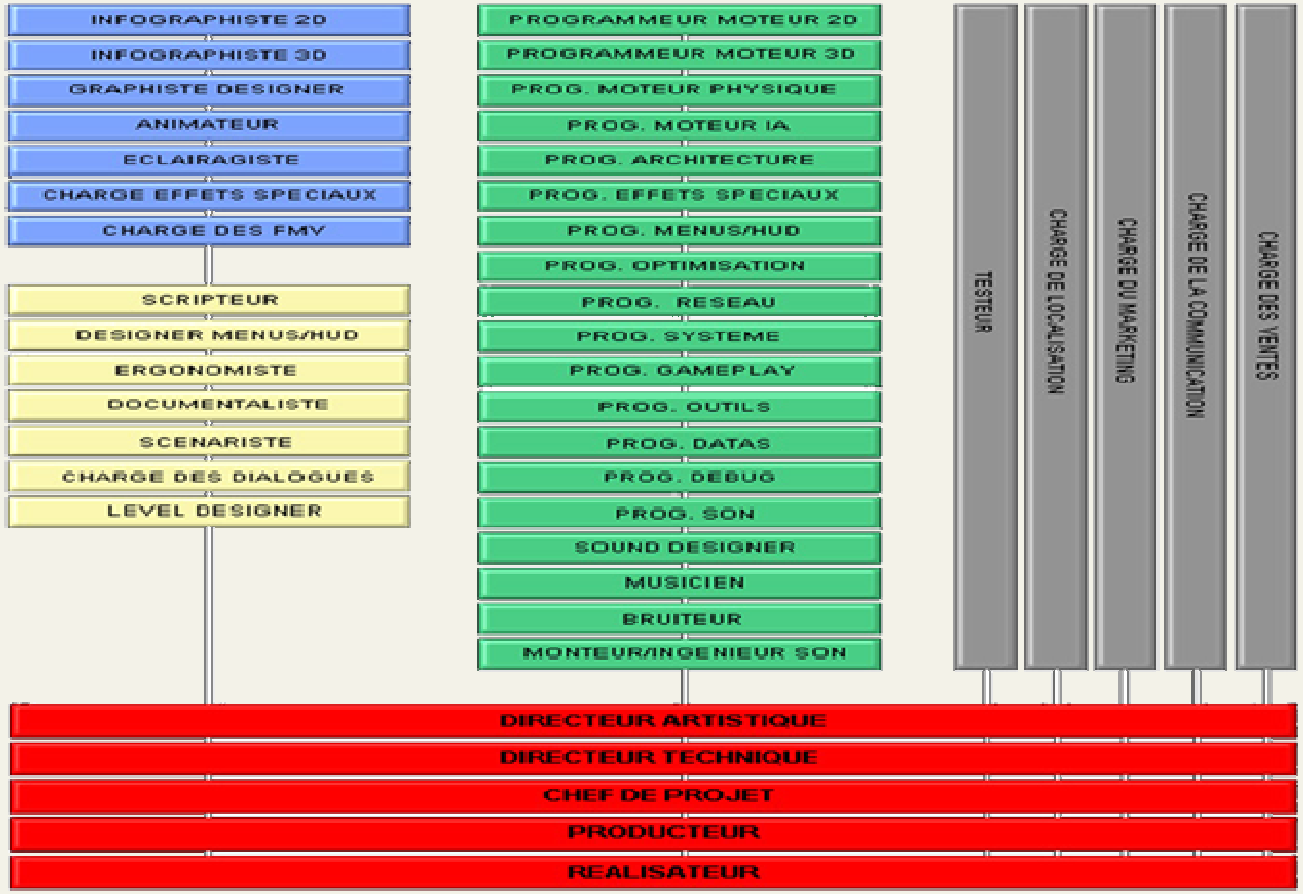


1995
3dfx ASM

1995
DirectX 1.0

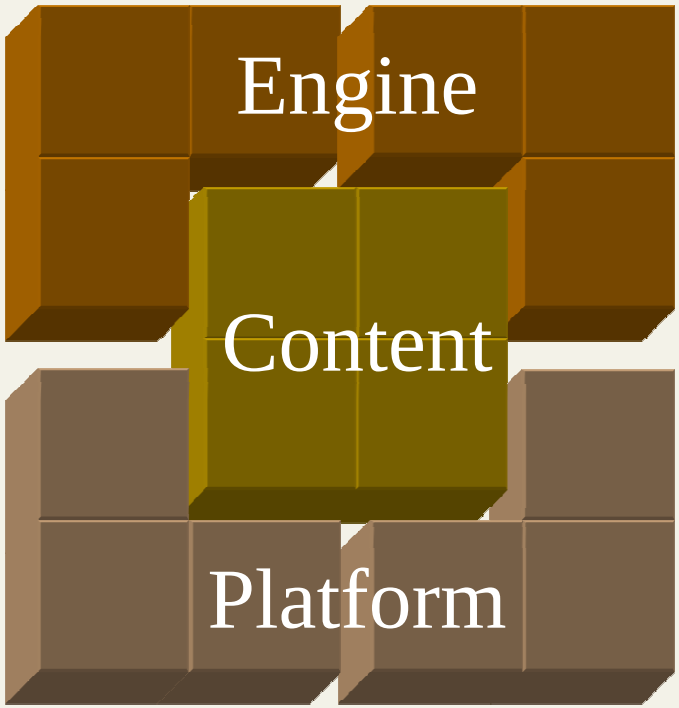
2002
Renderware

... ET DES ACTEURS POUR LES FAIRE

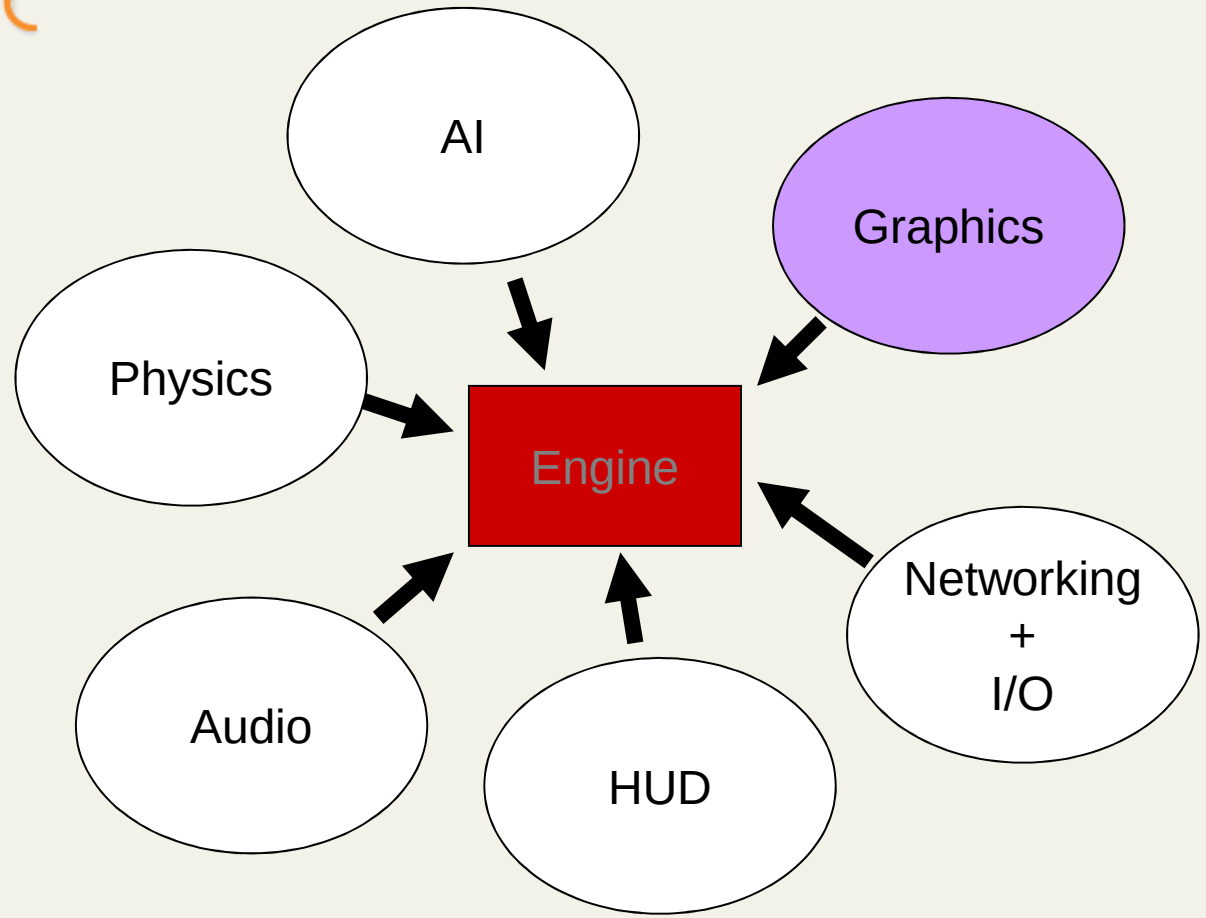


ECOLE NATIONALE
DU **JEU** ET DES **MEDIAS**
INTERACTIFS NUMERIQUES

Les composants d'un jeu



Les composants d'un moteur de jeu



Moteur de jeu

- Les studios choisissent de l'acheter ou de le faire
  - Quake Source, Unreal engines
  - Renderware, Gamebryo, Unity3D middlewares
  - Ils ne sont pas que des APIs mais également des outils d'aide à la création
- Ou d'acheter certains éléments
  - Combien de personnes peuvent faire un moteur physique ?
  - Et combien de studios peuvent supporter le coût d'en faire un ?
  - Video codecs, etc
- Un moteur peut être spécilisé/optimisé pour un genre de jeu

Programmes interactifs & immersifs

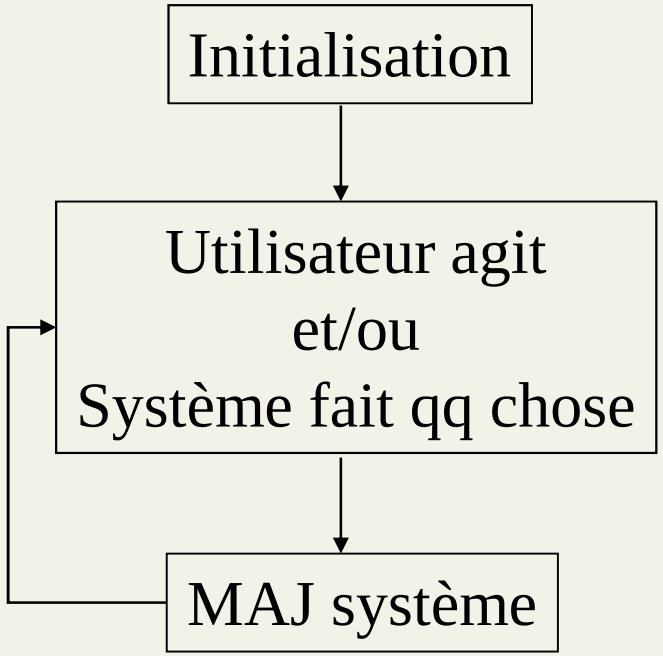
Battlefield 2

- Les jeux sont des simulations interactives
- Ils cherchent à immerger le joueur
- Fonctionnalités importantes pour les programmes interactifs ?
- Fonctionnalités importantes pour les simulations immersives ?

Fonctionnalités importantes

- L'utilisateur contrôle l'action/la narration
  - Le contrôle doit être direct et immédiat
- Le programme fournit des indications sur son état
  - L'utilisateur doit savoir et comprendre ce qu'il arrive
  - L'utilisateur doit obtenir une confirmation que ses actions ont été prises en compte par le programme

Structure d'un programme interactif



- Programmation événementielle
  - Toute réaction arrive en réponse à un événement
- Événements proviennent de :
  - L'utilisateur
  - Du système
- Les événements sont aussi appelés *messages*
  - Un événement produit l'envoi d'un message …
- 2 manières de gérer les événements : blocage ou non

Boucle temps-réel

- Le noyau central d'un système interactif :

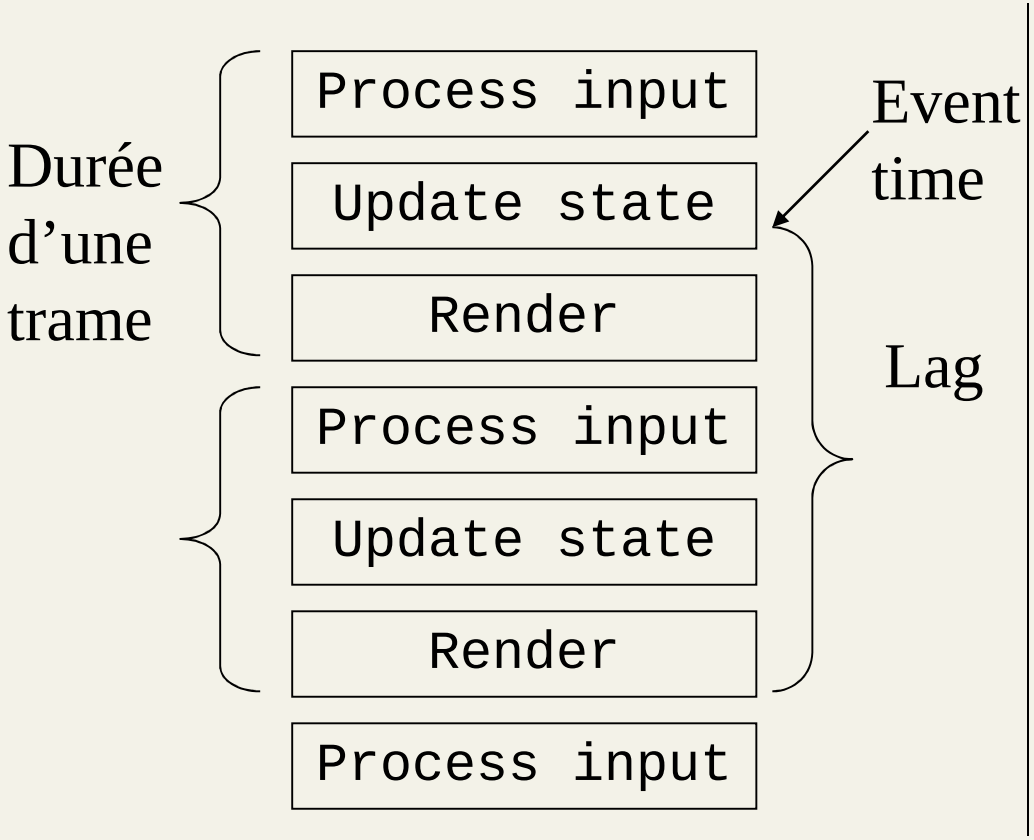
```
while ( true )
        process events
        update animation
        render
```

- Que peut-on (doit-on) faire d'autre dans cette boucle ?
- Le nombre d'exécution de cette boucle par seconde est le *framerate*
  - # frames per second (fps)

Lag (latence)

- Le *Lag* = laps de temps écoulé entre une action et la perception de son résultat
  - Trop de *lag* et la causalité est distordue
  - Si l'action est un mouvement dont la réponse est visuelle, trop de *lag* peut rendre certaines personnes malades
  - Trop de *lag* rend la vidée difficile (en général difficulté pour toute tâche de désignation)
- Une trop grande variabilité du lag rend également l'interaction difficile
  - Les utilisateurs peuvent s'adapter à un lag constant mais pas à une latence variable.
- D'un point de vue cognitif, le *lag* est la variable importante

Calcul du *Lag*



- La latence n'est pas le temps mis pour calculer une trame !

Réduire le *Lag*

- Réponse triviale : un matériel + rapide et des algos + efficaces
- En pratique
  - on peut définir un *framerate* cible et on essaye de faire le plus de chose pendant cette durée impartie
    - Cela dépend très fortement du matériel
    - D'où la nécessité de proposer des options à l'utilisateur
  - on gère le temps comme une ressource : combien de temps octroie-t-on pour chaque aspect du jeu (graphiques, IA, son, …
  - on sépare les calculs
    - Le but est de libérer du temps de calcul pour garder un *lag* action utilisateur/notification acceptable
    - Pour certains aspects un lag important peut être acceptable : la réaction d'un ennemi, un changement d'animation, …
    - Technique : mettre à jour différentes parties à des rythmes différents :
      - Par exemple : graphique à 60fps, IA à 10, la physique à 30

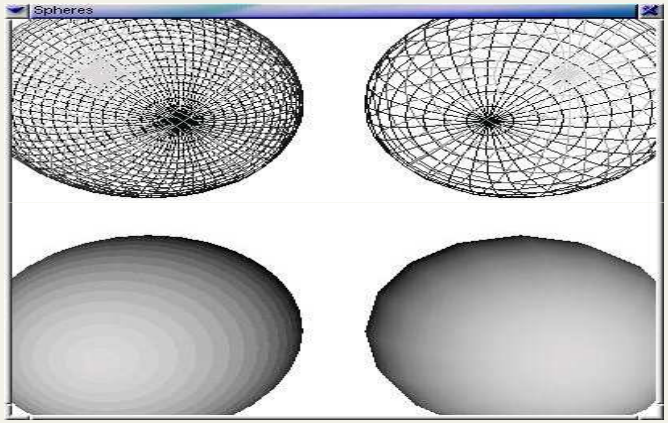
Visualisation des scènes : deux méthodes

- Projection de facettes :
  - Les objets doivent être "triangulisés" (*tessaleted*)
  - Et tout triangle ou polygone subit les opérations du pipeline 3D
  - Transformations, visibilité, *clipping*, projection, coloriage
- Lancé de rayons :
  - On utilise ici les équations de surface des objets donc pas besoin de "trianguliser" les objets
  - Calcul pour chaque pixel de l'écran de l'équation partant de l'observateur et passant par ce pixel
  - Pour chaque rayon, calcul des intersections avec les objets pour déterminer la couleur du pixel

Visualisation des scènes : deux méthodes

Par projection de facettes

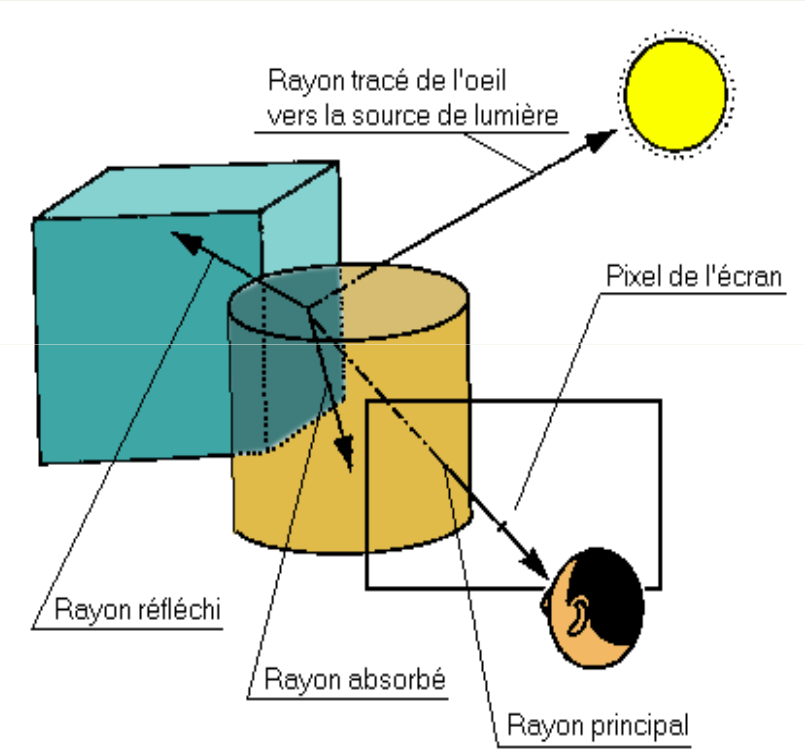
1. Transformation des formes complexes en ensemble de triangles



2. Transformation, illumination, projection et coloriage de chacun des triangles → pipeline 3D

« quantité plutôt que qualité »

Par lancé de rayons



« qualité plutôt que quantité »

Visualisation des scènes : deux méthodes

Par projection de facettes

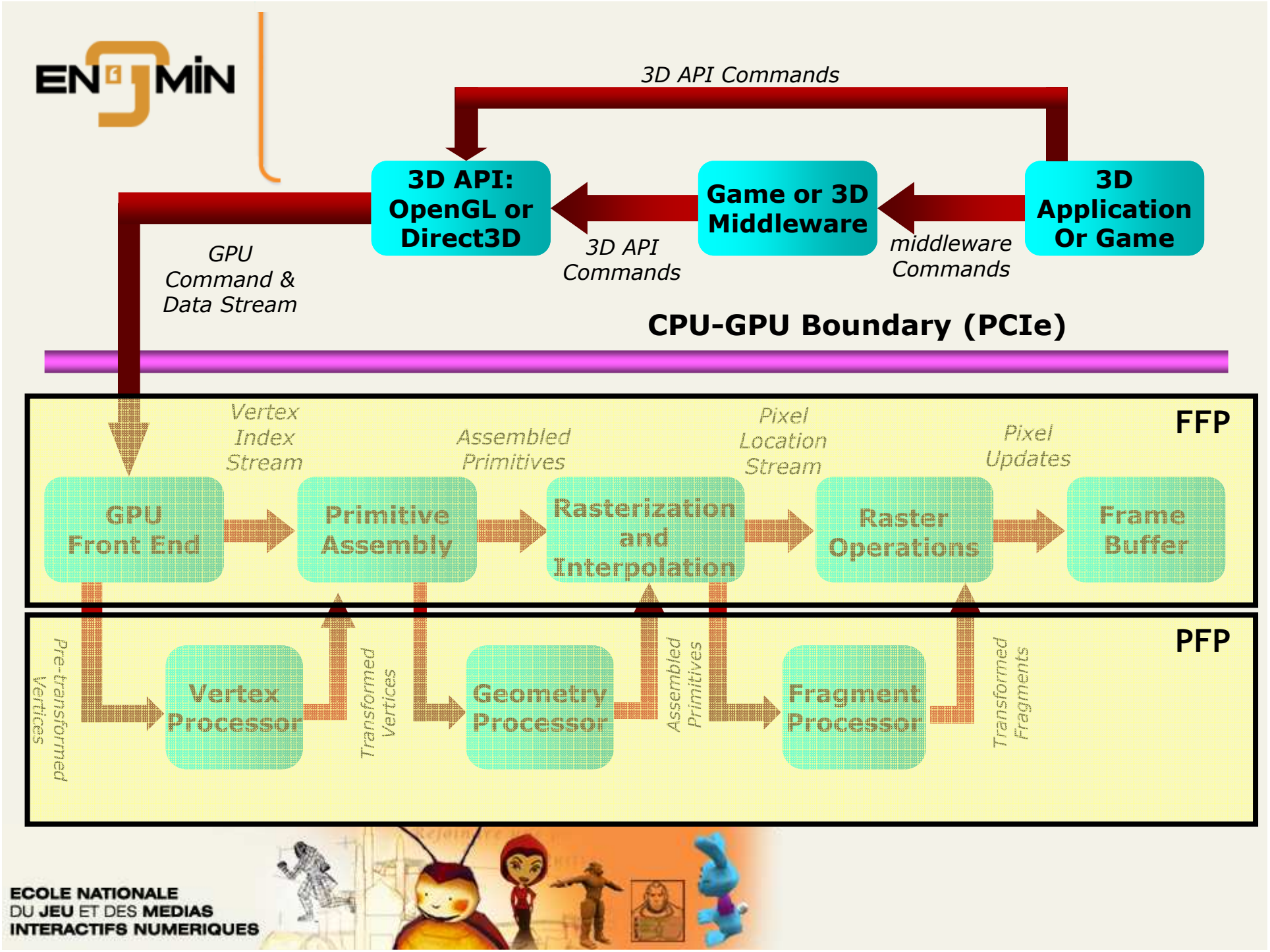
© Activision

« quantité plutôt que qualité »

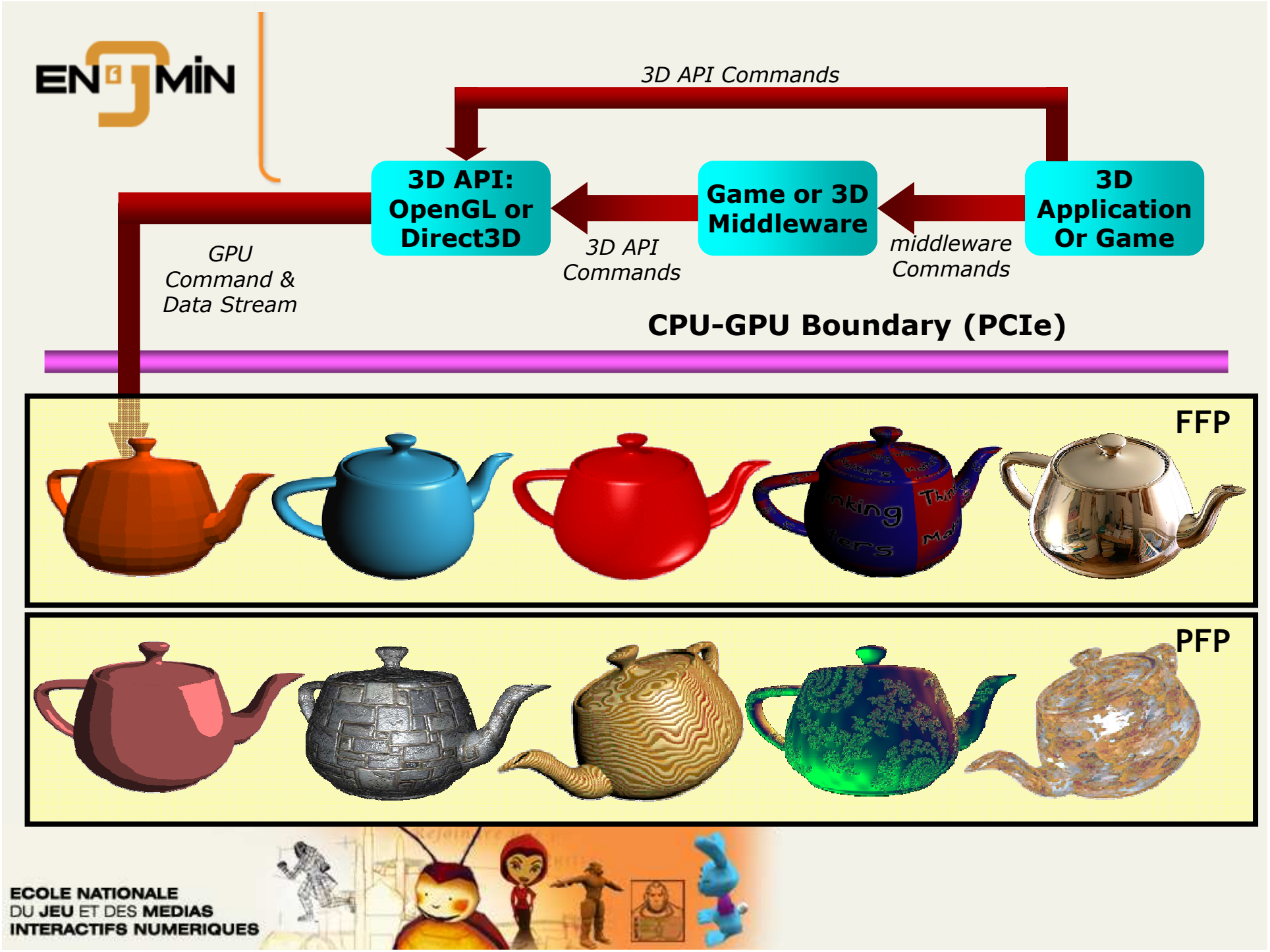
Par lancé de rayons

© Dreamworks

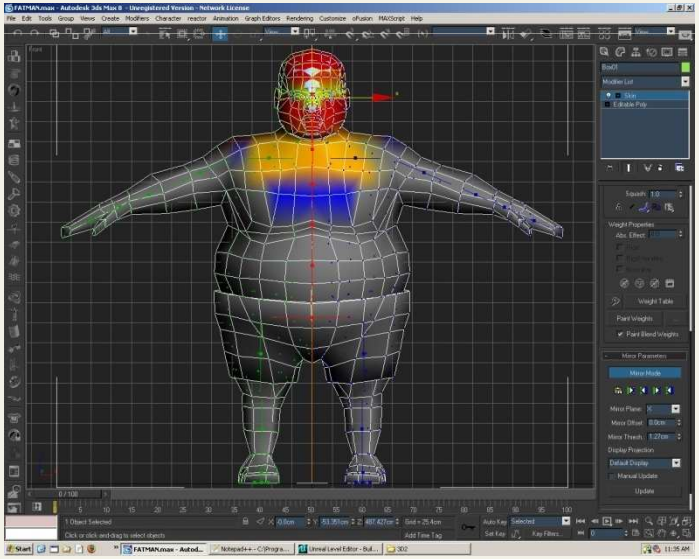
« qualité plutôt que quantité »



ECOLE NATIONALE
DU JEU ET DES MEDIAS
INTERACTIFS NUMERIQUES


3D API Commands
3D API: OpenGL or Direct3D
Game or 3D Middleware
3D Application Or Game
GPU Command & Data Stream
3D API Commands
middleware Commands
CPU-GPU Boundary (PCIe)
FFP
PFP
ECOLE NATIONALE
DU JEU ET DES MEDIAS
INTERACTIFS NUMERIQUES

Animation

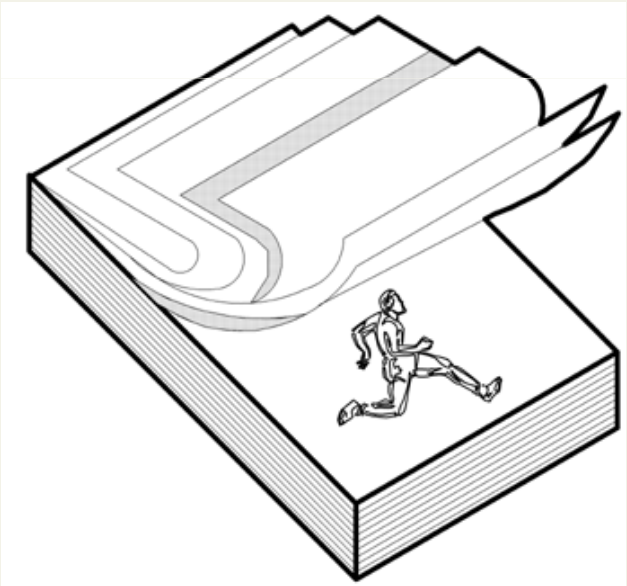


Jeff Lew

ECOLE NATIONALE
DU JEU ET DES MEDIAS
INTERACTIFS NUMERIQUES

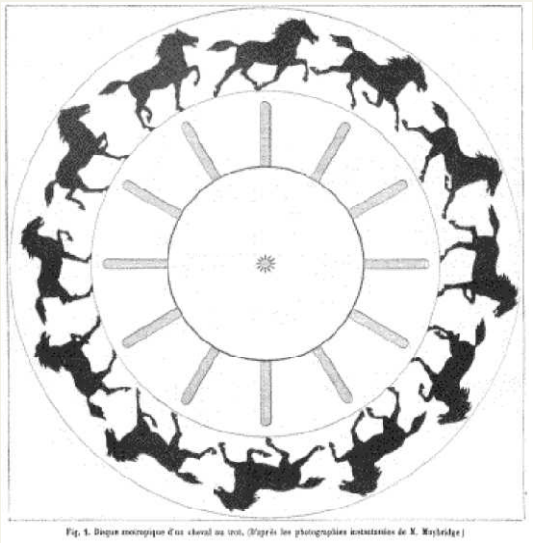
Qu'est-ce qu'une animation ?

- Animation = séquence d'images fixes affichées successivement
- Peut être fait de différentes manières :
  - En dessinant toutes les images
  - En laissant l'ordinateur calculer certaines transitions
  - En laissant l'ordinateur calculer tout



COMMENT ANIMER ?

- Image par image



revue *La Nature* http://cnum.cnam.fr

E. MUYBRIDGE - Zoopraxinoscope

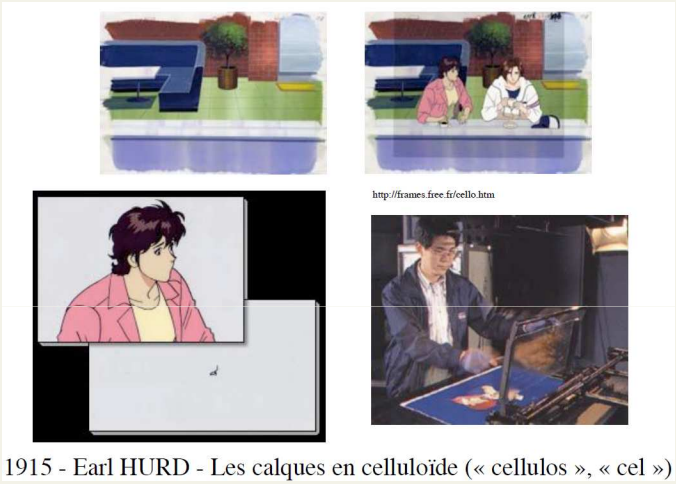
1895 - Auguste et Louis LUMIERE (Lyon)- Le cinématographe

http://www.loc.gov/rr/print/swann/artwood/aw-animation.html

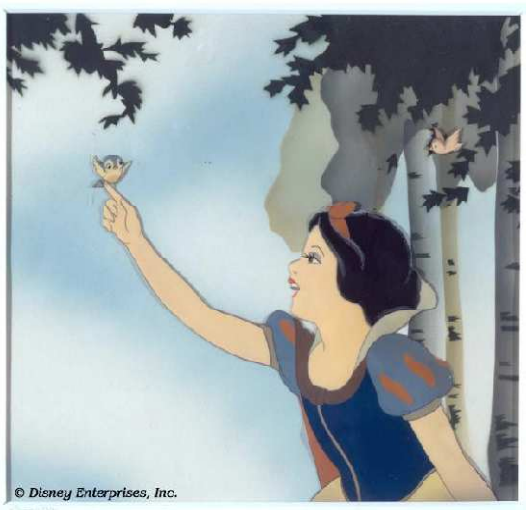
ca 1910 - Windsor McCAY - Gertie the dinosaur

Comment animer ?

- Célluloides



1915 - Earl HURD - Les calques en celluloïde (« cellulos », « cel »)



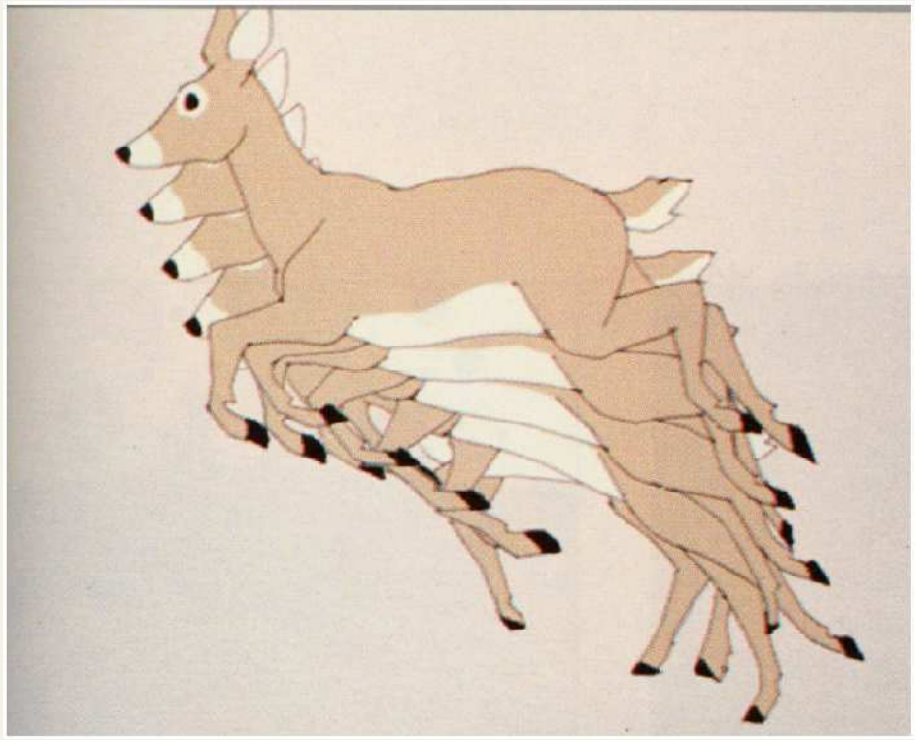
1937 - Walt DISNEY - Snow White

Multiplane camera

(musée du Walt Disney studio, Burbank)

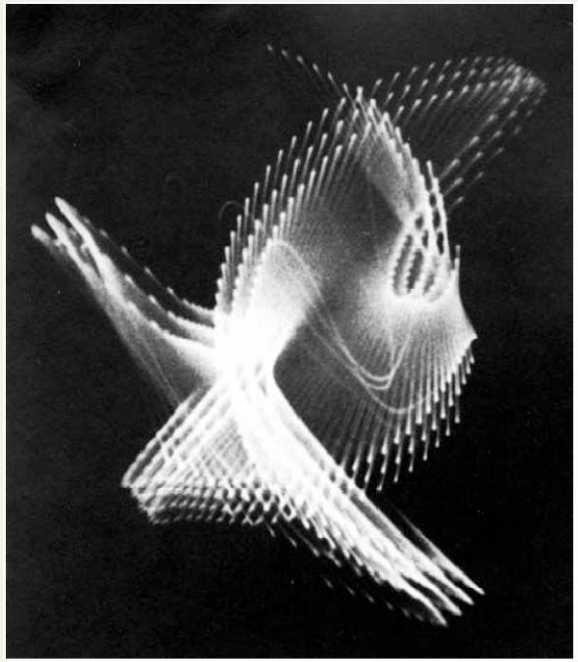
Comment animer ?

- *Keyframing* analogique

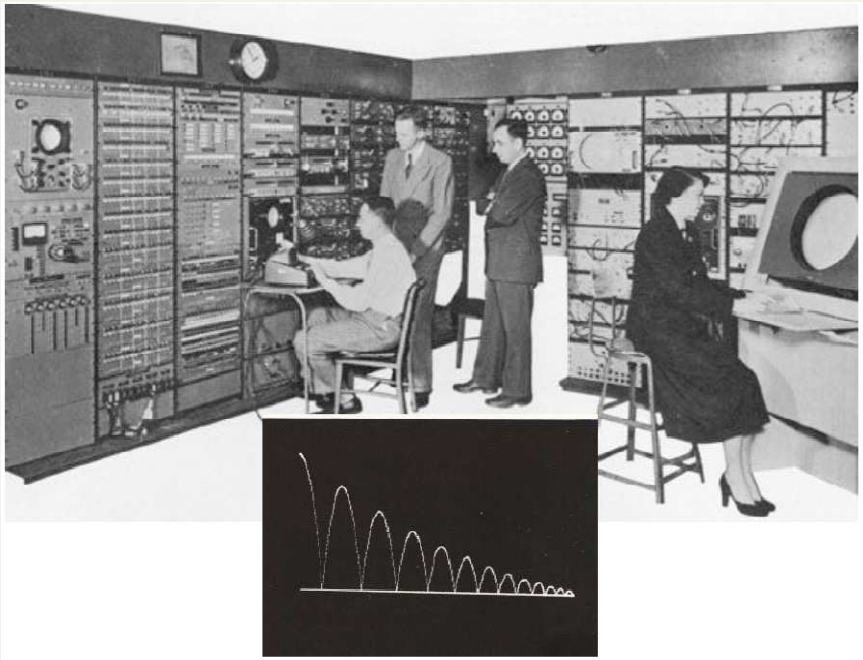


Comment animer ?

- Les première animations (elles étaient procédurales !)



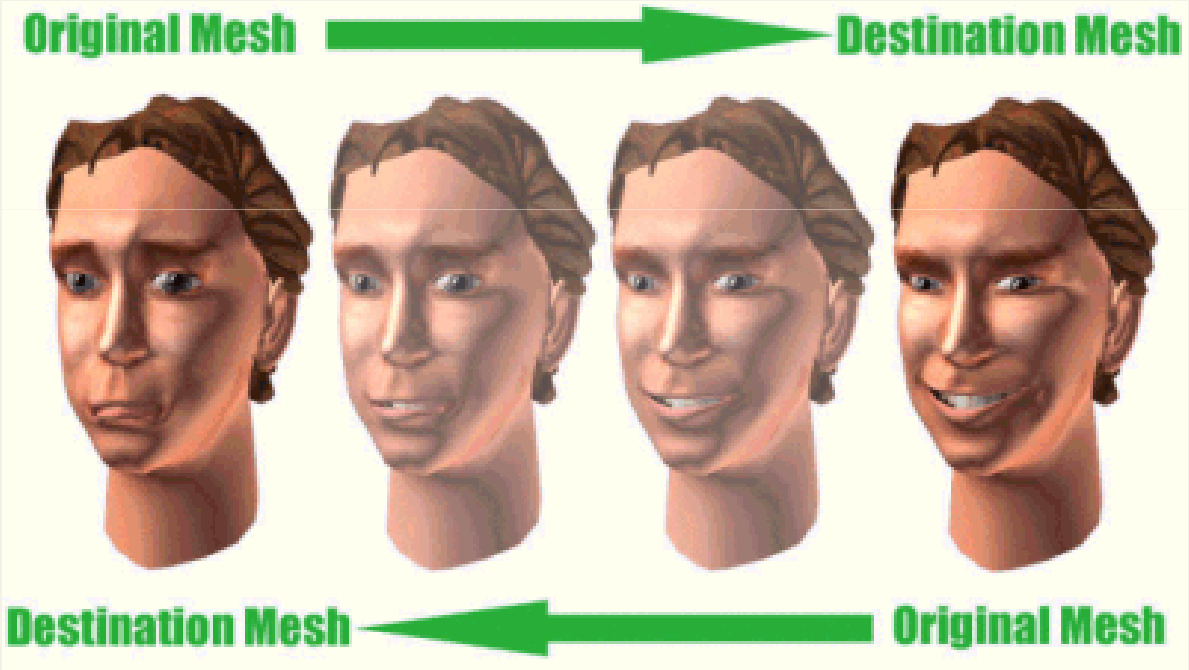
1950 - Ben Laposky



1949-51 - Projet WHIRLWIND

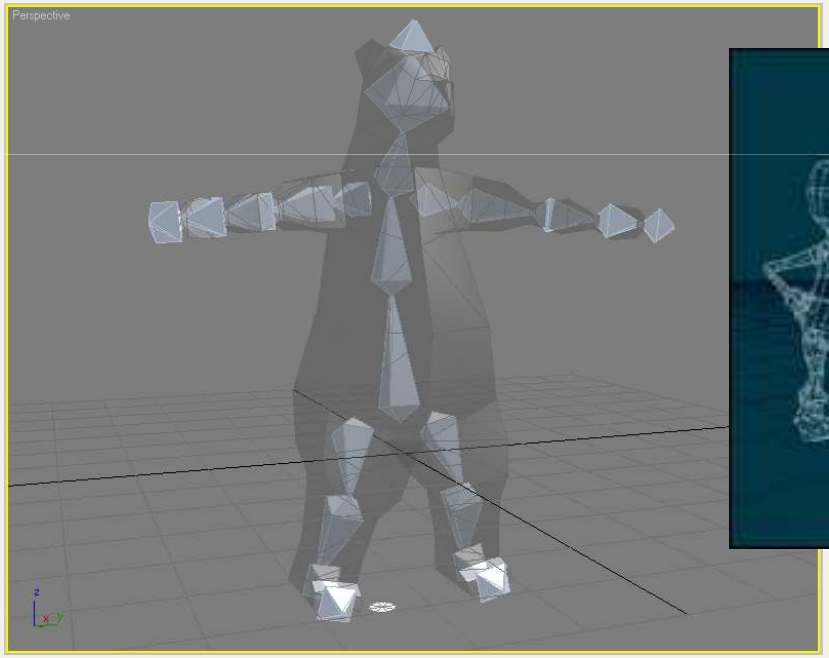
Animation par ordinateur

- *Keyframing*
  - → Interpolation de valeurs (positions, couleurs, textures, …)



Animation par ordinateur

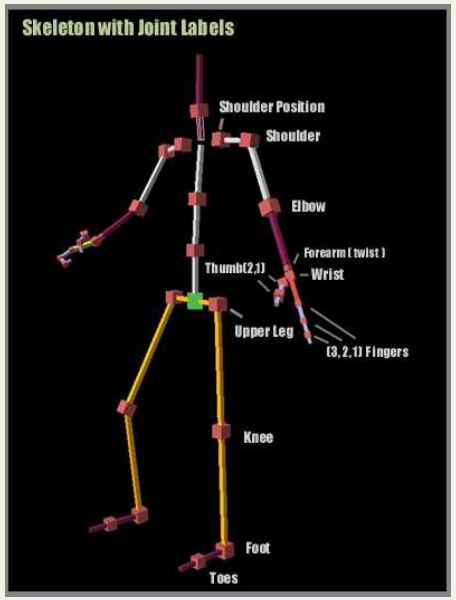
- *Boning & skinning*
  - → Associer un squelette avec un maillage
  - → Bouger un os du squelette bouge la portion du maillage associée

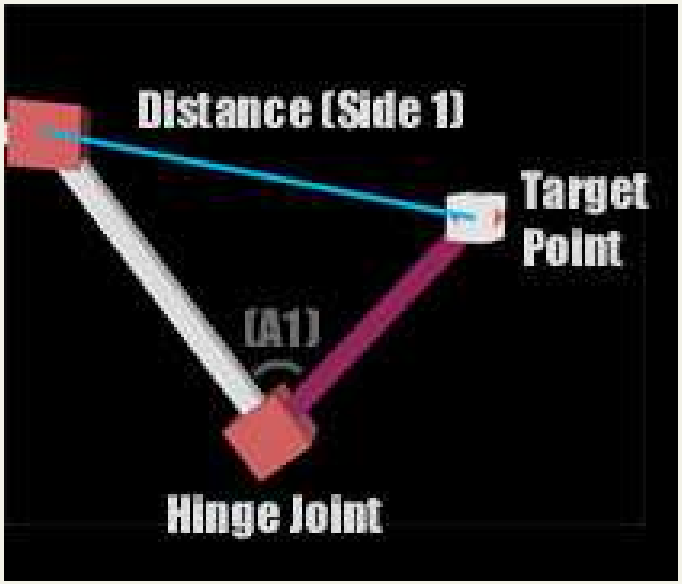


Jeff Lew

Animation par ordinateur

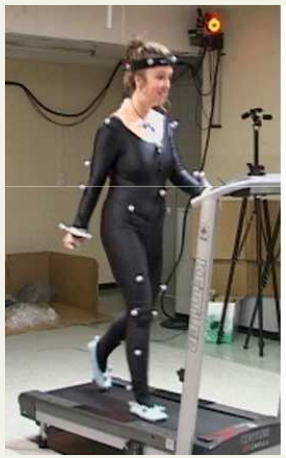
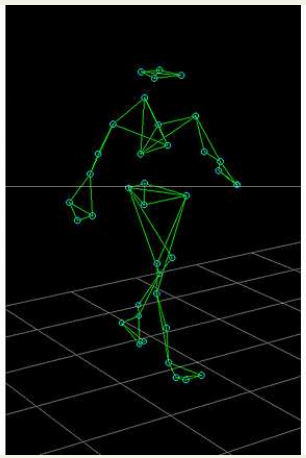
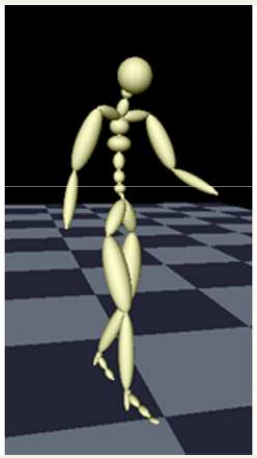
- *Inverse Kinematics*
  - → Etant donnée une position finale, calcule les angles entre les os
  - → Prise en compte de degrés de liberté et de contraintes

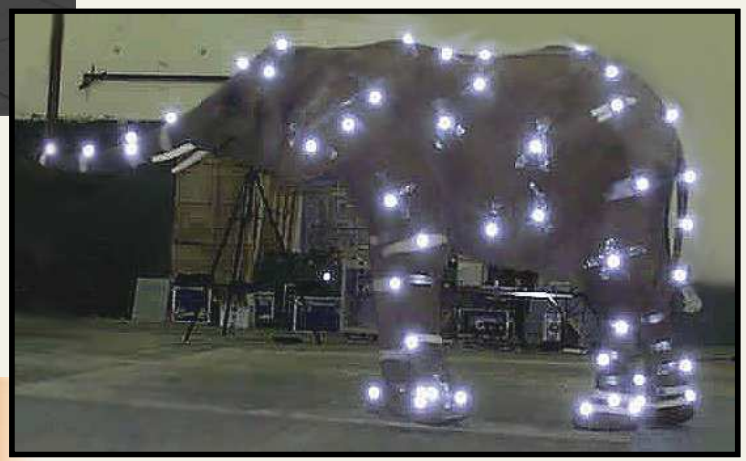




Animation par ordinateur

- Motion Capture
  → Enregistrement de mouvements

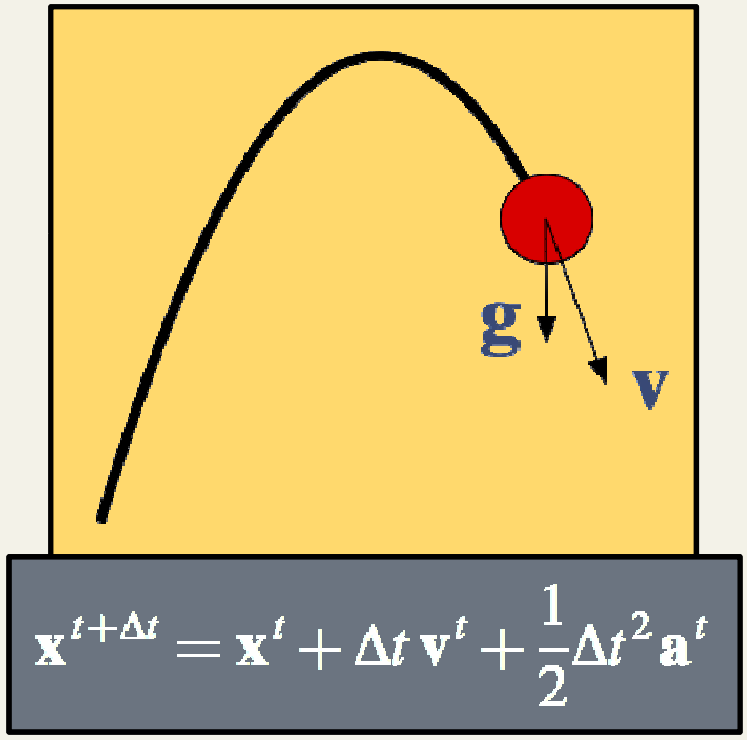



Animation procédurale

- Réduire le nombre d'animations à créer
  - Parce que c'est coûteux
  - Parce qu'elles ne sont pas facilement réutilisables (morphologies, jeux différents)
  - La physique, les émotions nécessitent une quasi infinité d'animations
- Comment ?
  - Principalement physique :
    - ⇨ Corps rigides
    - ⇨ Particules
    - ⇨ Masses-ressorts
    - ⇨ Rag-dolls
  - Mais aussi IA
    - ⇨ Pathfinding
    - ⇨ Émotions
    - ⇨ Foule

Animation procédurale

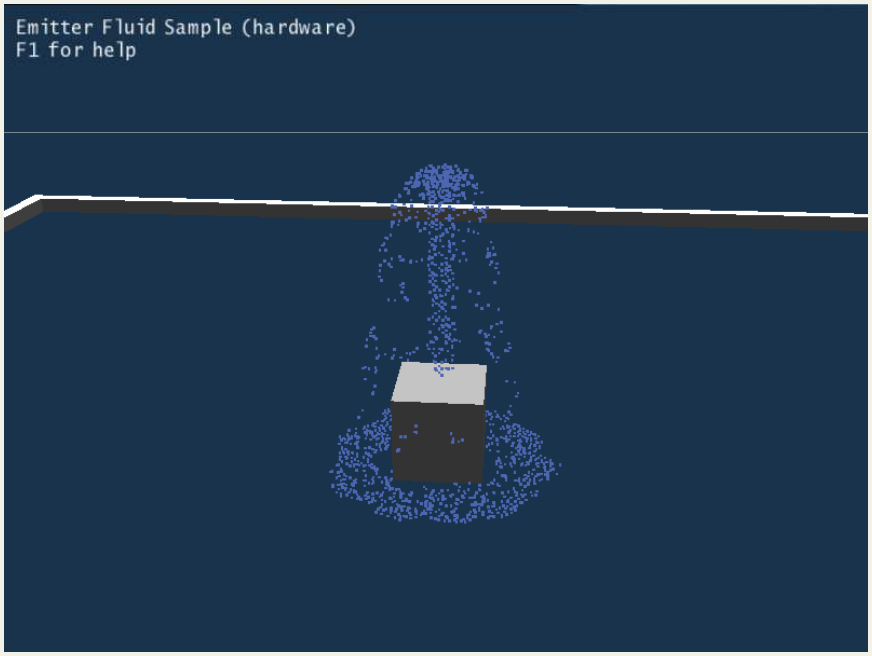
- Corps rigide = taille constante, pas de déformation
  - Ils suivent la 2ème loi de Newton



$$\mathbf{x}^{t+\Delta t} = \mathbf{x}^t + \Delta t\,\mathbf{v}^t + \frac{1}{2}\Delta t^2\,\mathbf{a}^t$$

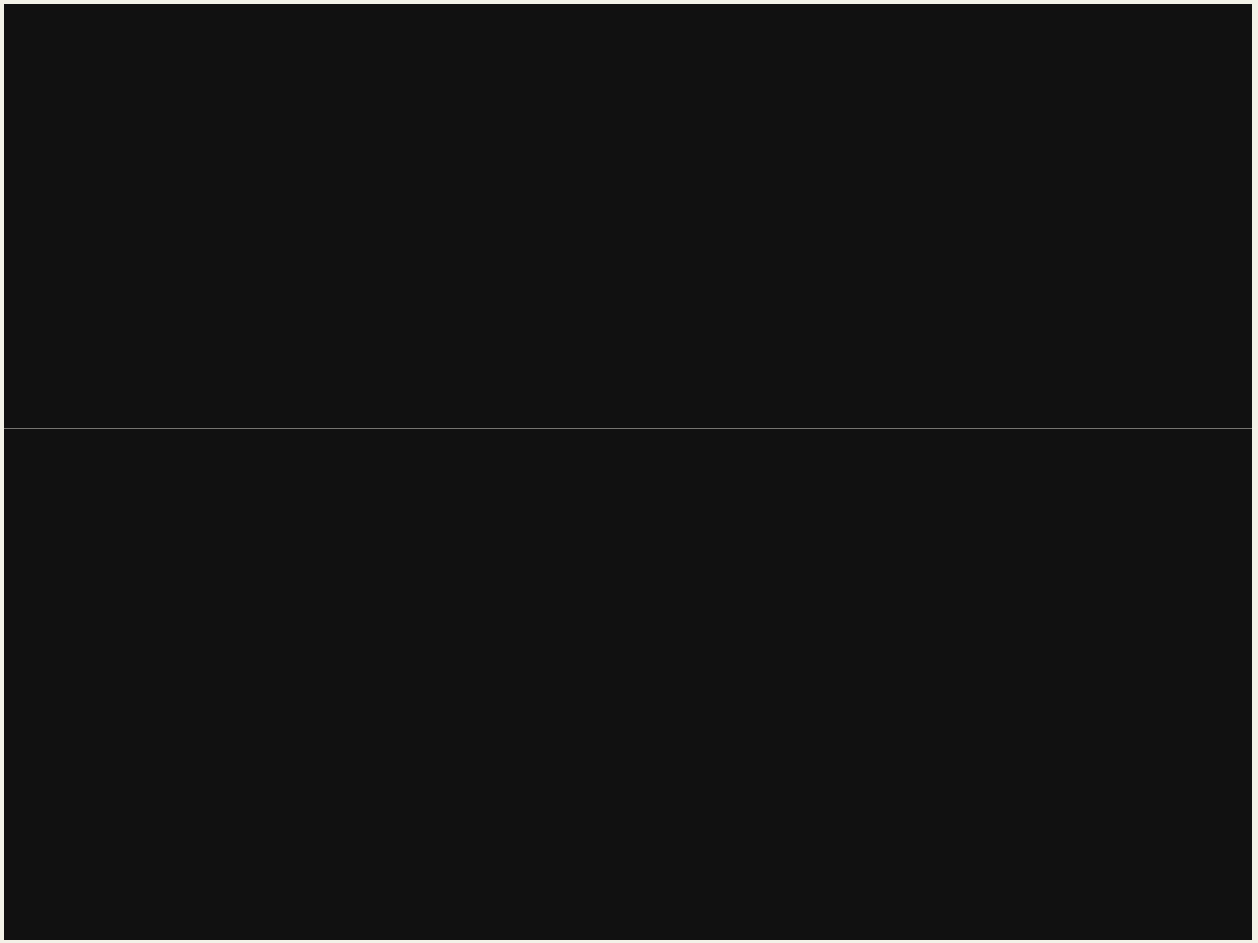
Animation procédurale

- Particules = solides indéformables (sans collisions)
  - Mêmes lois physiques (i.e. gravitation-attraction-repulsion)



© NVidia – PhysX SDK

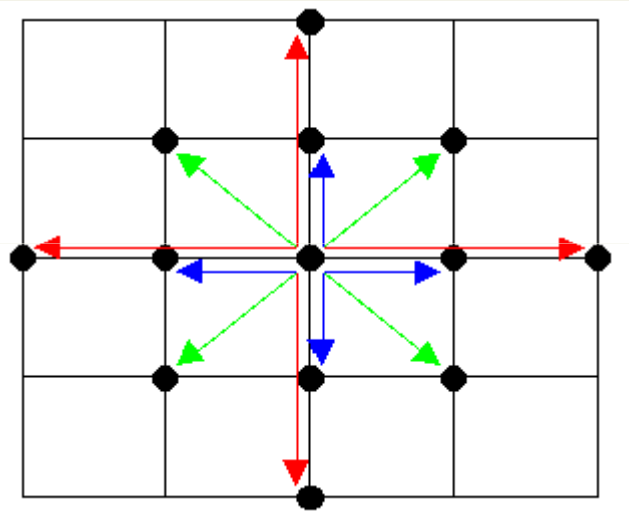
Animation procédurale



ECOLE NATIONALE
DU JEU ET DES MEDIAS
INTERACTIFS NUMERIQUES

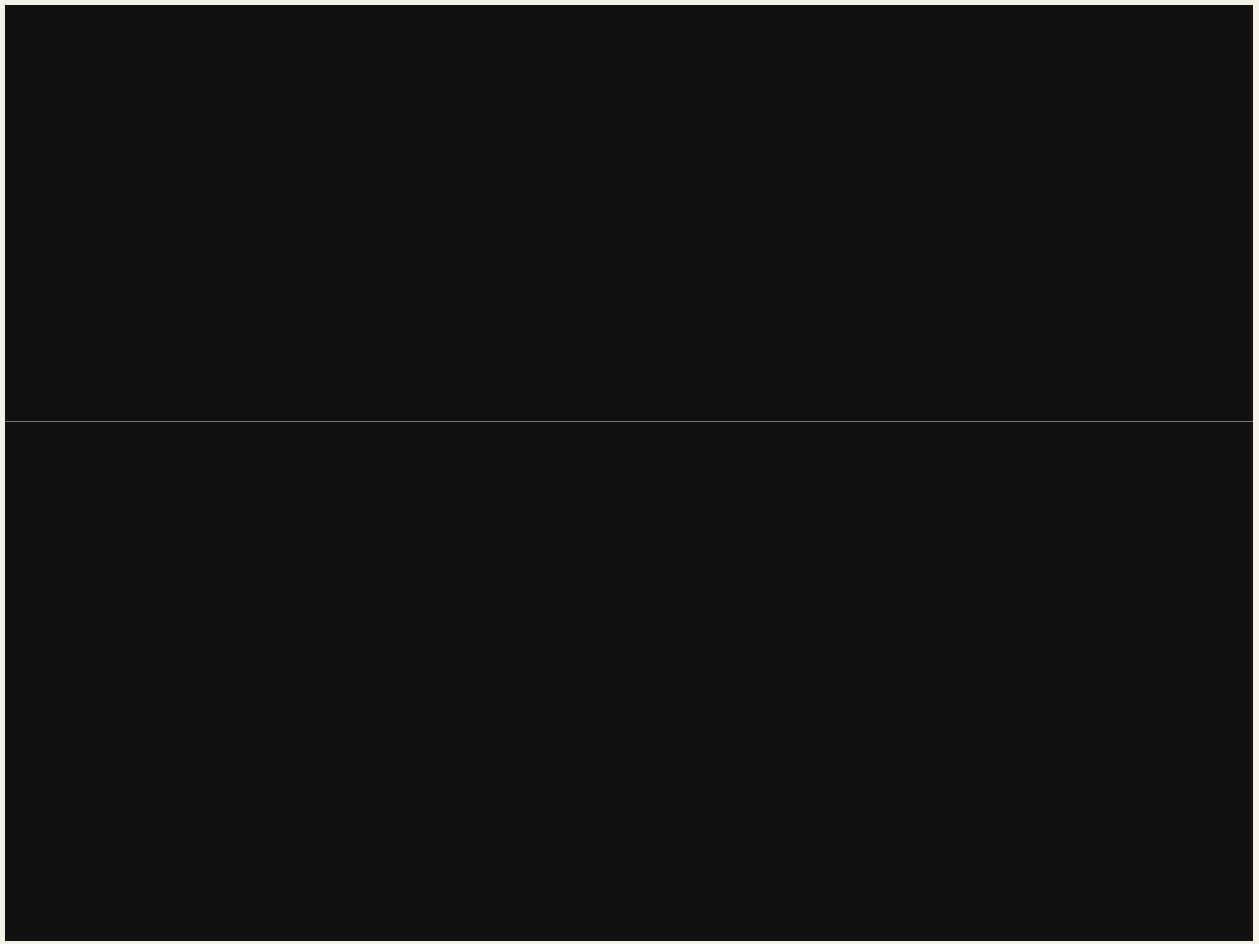
Animation procédurale

- Systèmes masses-ressorts = particules liées les unes aux autres



- Permettent d'animer les cheveux,
  les vêtements, et objets déformables

Animation procédurale

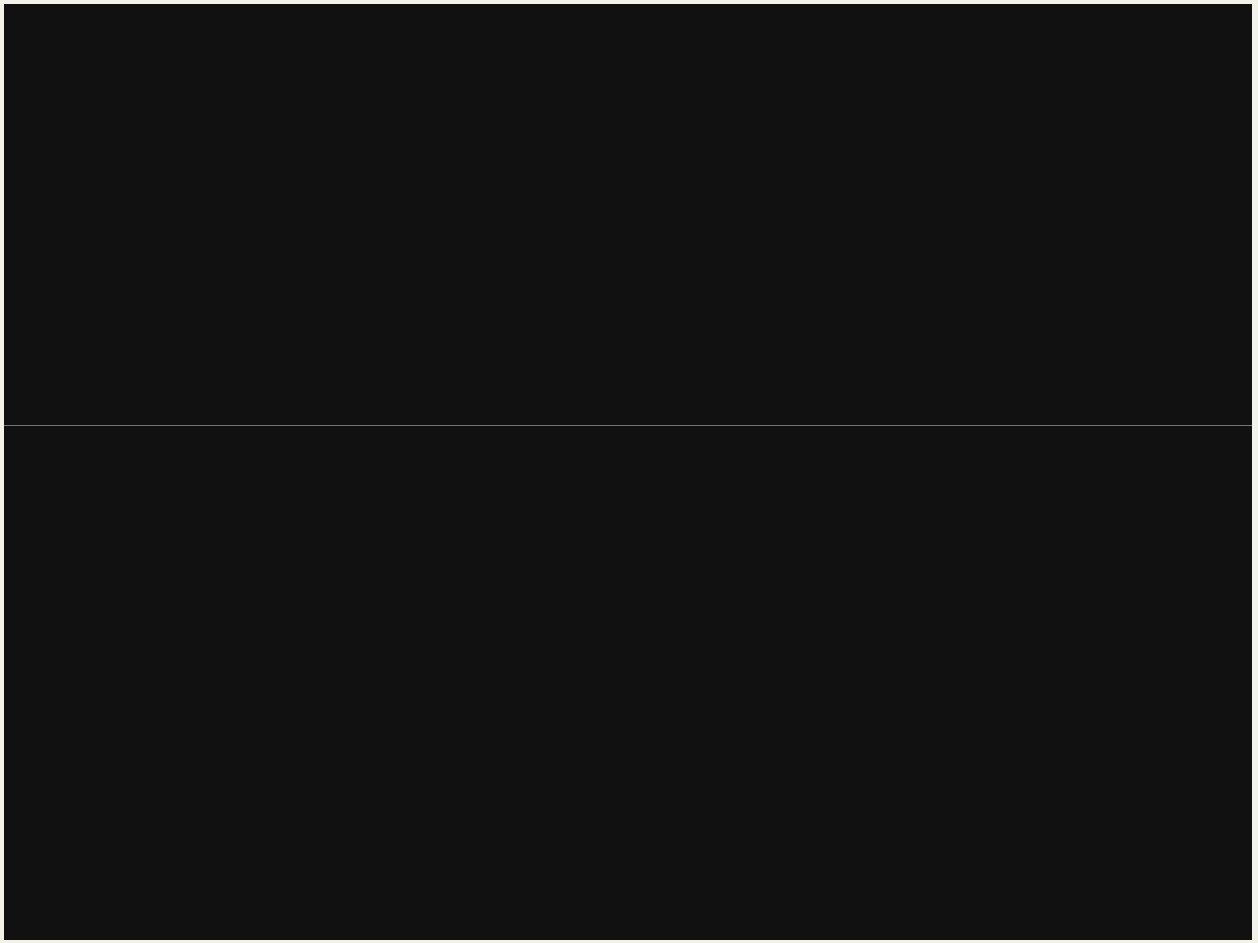


Animation procédurale

- *Rag-dolls* = les os du squelette sont gérés comme des solides
  - Ils sont reliés par des jointures qui ont des contraintes
- Simplifications :
  - Calculs pour certains os
  - Calculs pour certains DDL
- *Blending* facile entre une animation KF, IK et une animation physique *ragdoll*

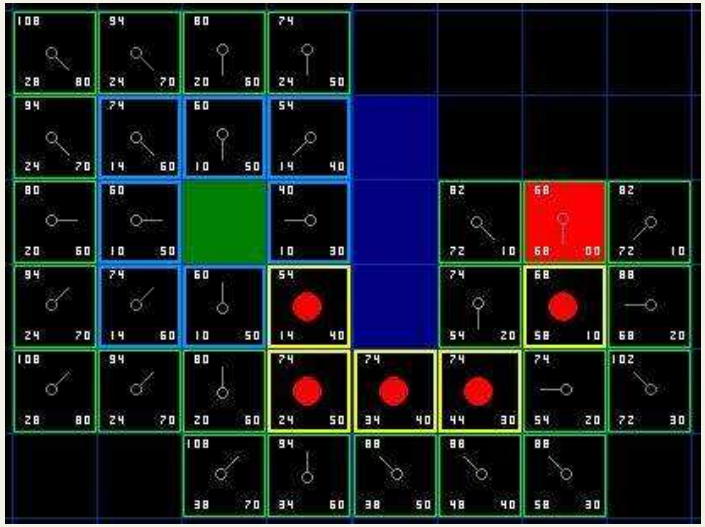
© Ubisoft – Assassin's creed

Animation procédurale



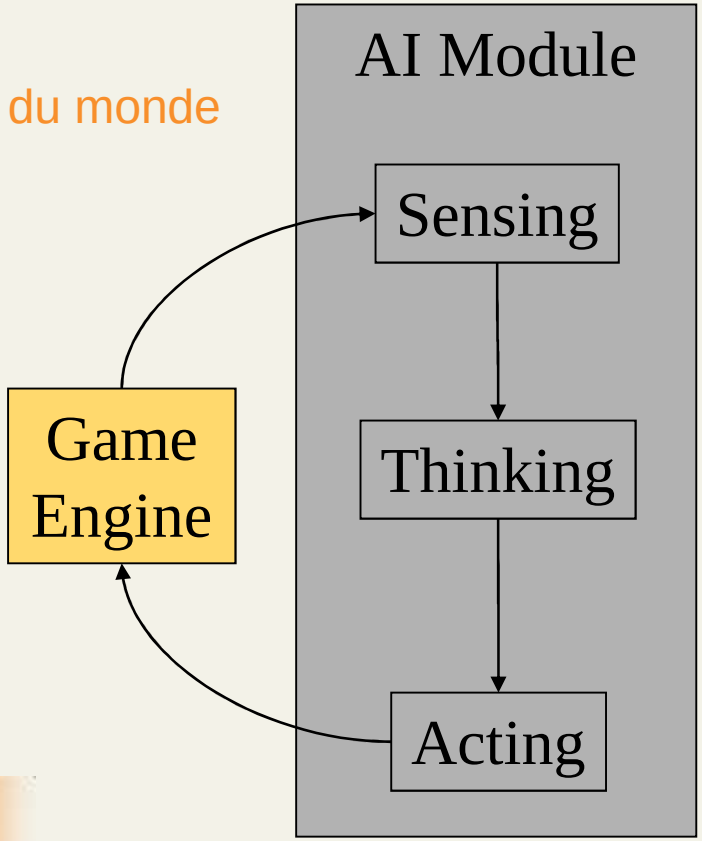
Animation procédurale

- IA
  - Path finding : trouver son chemin dans un labyrinthe
  - Comportements adaptatifs (Black & White, Creatures): utilisation des événements passés pour accommoder les réactions
  - Foule : bouger automatiquement plusieurs PNJ



Intelligence artificielle

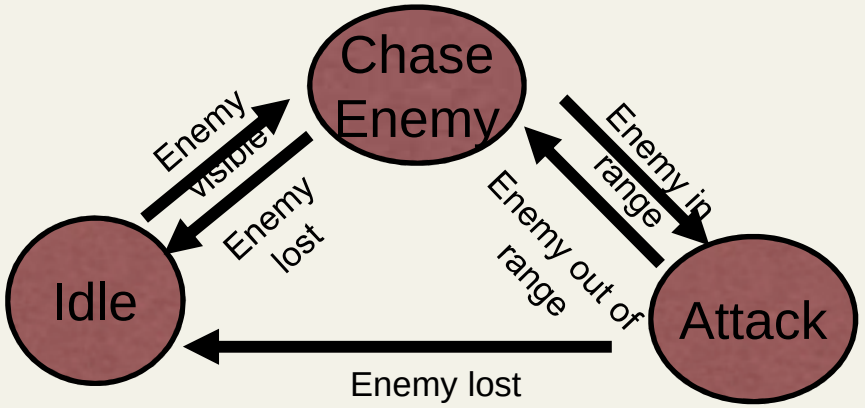
- L'IA fait partie de la boucle de jeu. Les calculs sont faits après les actions du joueur et avant l'affichange
- Elle se décline en 3 phases :
  - La captation permet d'établir l'état du monde
  - Ca peut être très simple puisqu'on peut considérer les événements
  - Ou complexe si l'on veut une sémantique de plus haut niveau
- La phase de réflexion décide ce qu'il y a à faire
- La phase d'action donne les ordres



Intelligence artificielle

- 2 modes d'appels du moteur d'IA :
  - *À une fréquence fixée (polling)*
  - En fonction d'événements (*event driven)*
- Programmer une IA, c'est chercher à satisfaire ces critères :
  - Recherche de solution – décider d'une stratégie et fait en sorte de l'appliquer
  - Réactive – répond immédiatement aux changements
  - Gérer l'expérience – engrange des connaissances
  - Développement rapide et efficace
  - Utilisation faible du CPU et de la mémoire
  - ➔ Ils sont antagonistes en général
- Grand nombre de solutions :
  - Machines à états finis, graphes de décisions, réseaux neuronaux, logique floue, …

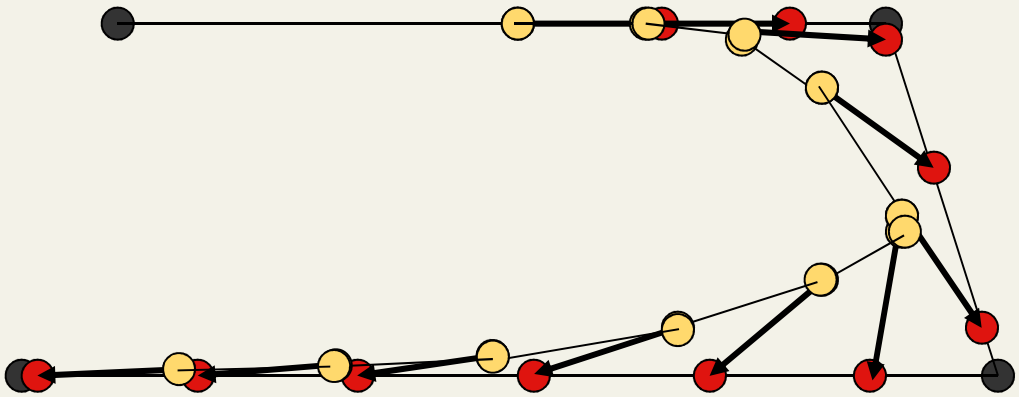
Intelligence artificielle



- Les machines à états finis
  - Un ensemble d'états dans lequel peut être l'agent
  - Connectés par un ensemble de transitions dépendant des changements dans le monde
- Exemple : les quake bots
  - Se balader aléatoirement si je ne perçoit aucun ennemi
  - Si voit un ennemi, attaque
  - Quand entend un ennemi, le rechercher
  - Quand meurt, renaît
  - Quand niveau de vie bas et voit ennemi bat en retraite

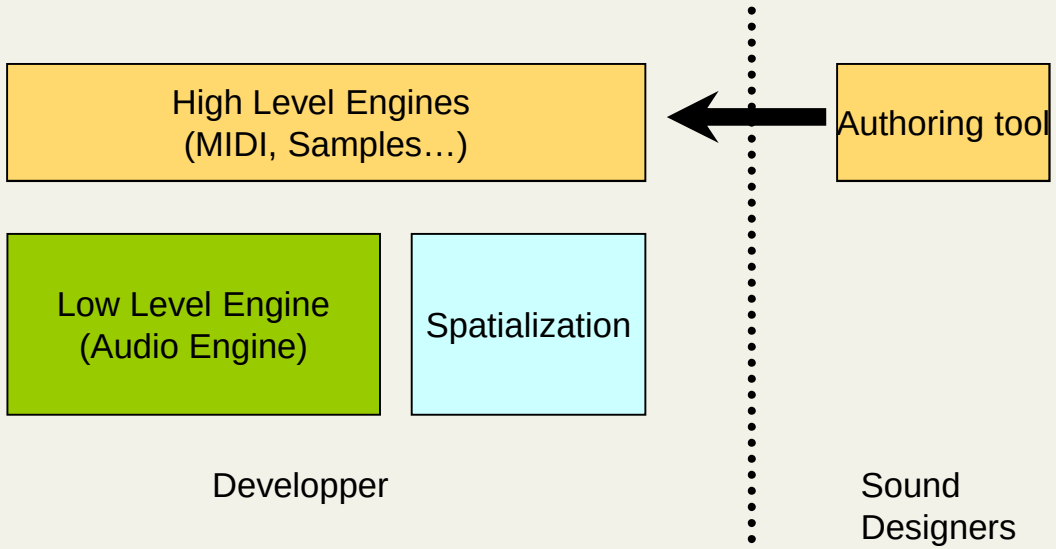
Intelligence artificielle – *Path finding*

- Problématique très courante dans les JV :
  - Dans un FPS : comment un PNJ va de salle en salle ?
  - Dans un RTS : le joueur donne l'ordre à des unités d'aller quelque part. Comment y vont-elles ? Comment s'évitent-elles ?
- A* est l'algorithme générique généralement choisi
  - Il permet de minimiser une valeur : généralement la distance ou le temps pour aller d'un point A à un point B
  - Mais ne suffit pas pour avoir un résultat rapide et efficace
- *Chase the point* : les agents vont vers une cible qui se déplace le long du chemin trouvé par A*

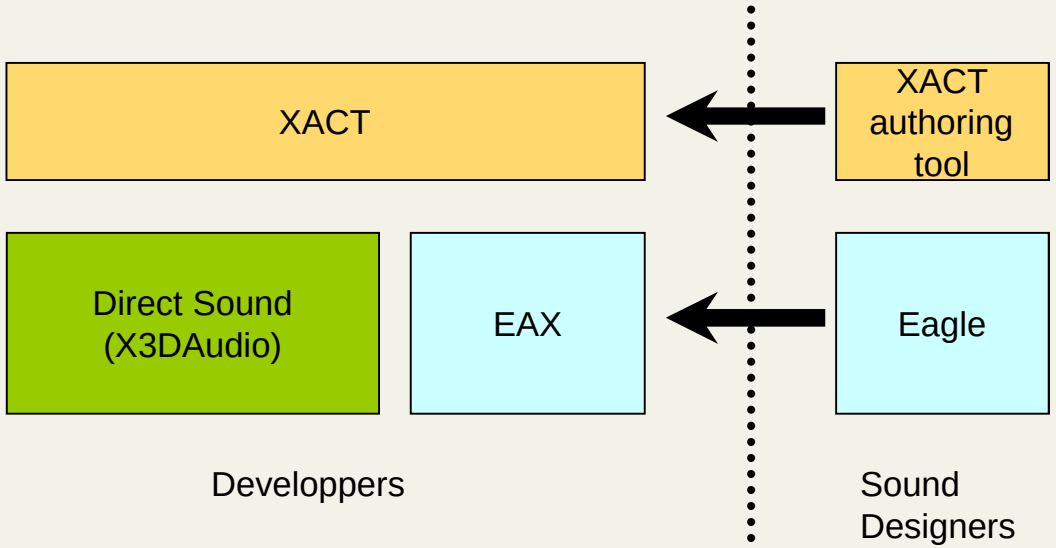


Audio

- Generic Engine

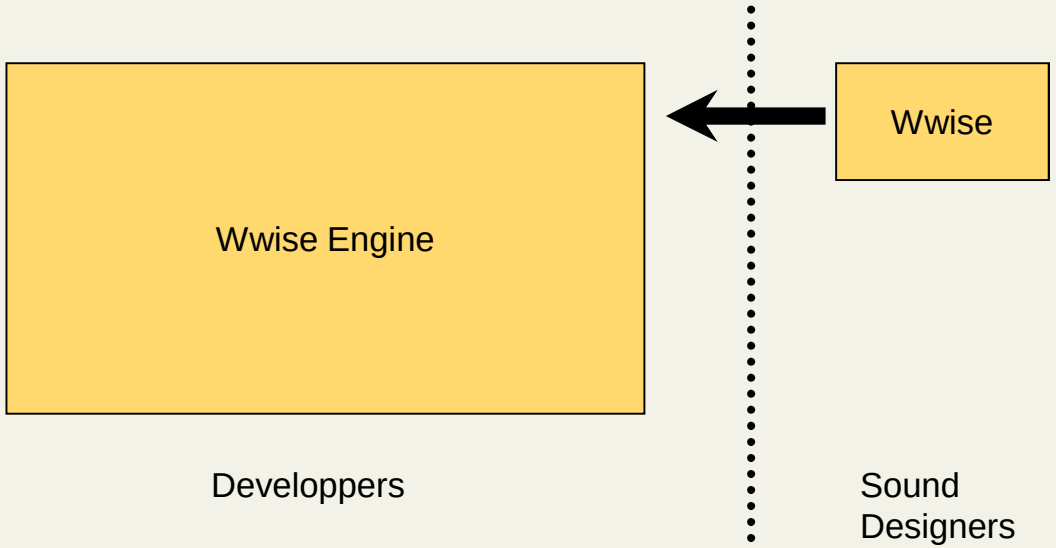


- Microsoft


XACT
XACT authoring tool
Direct Sound (X3DAudio)
EAX
Eagle
Developpers
Sound Designers

Audio

- Audiokinetic

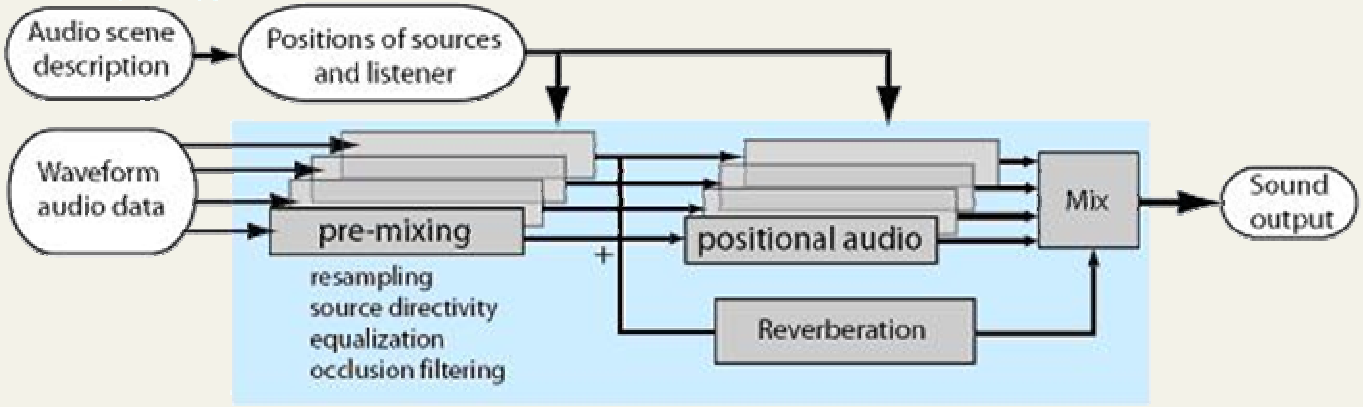


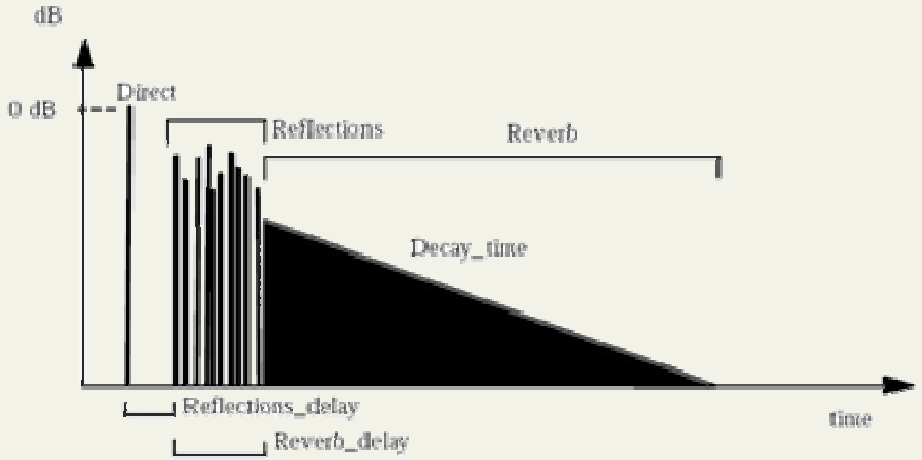
Audio

- APIs basées sur un modèle psycho-accoustique (*vs* physique)
- Mêmes fonctionnalités :
  - Chargement des samples
  - Positionnement des sources sonores
  - Positionnement du listener
  - Mixage
  - Réverbération
  - Occlusion

Audio

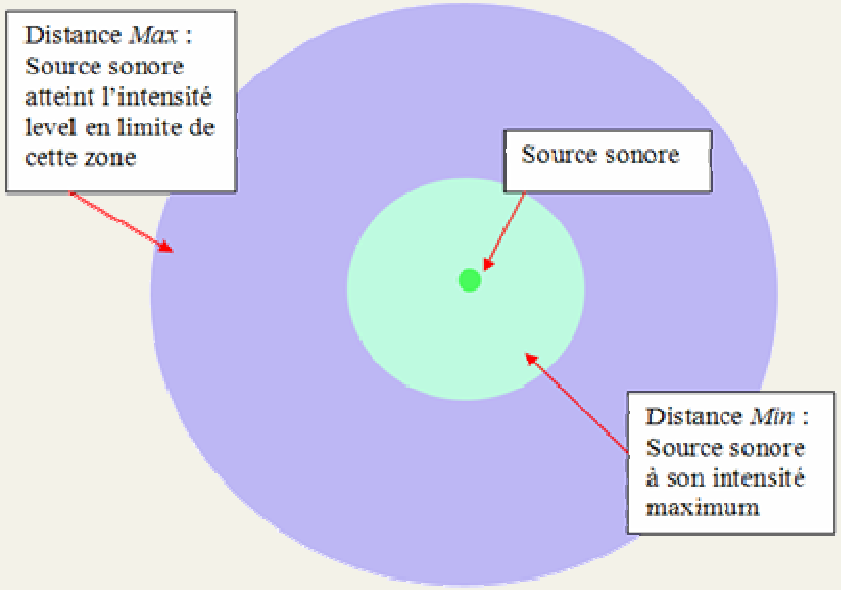
Traditional pipeline





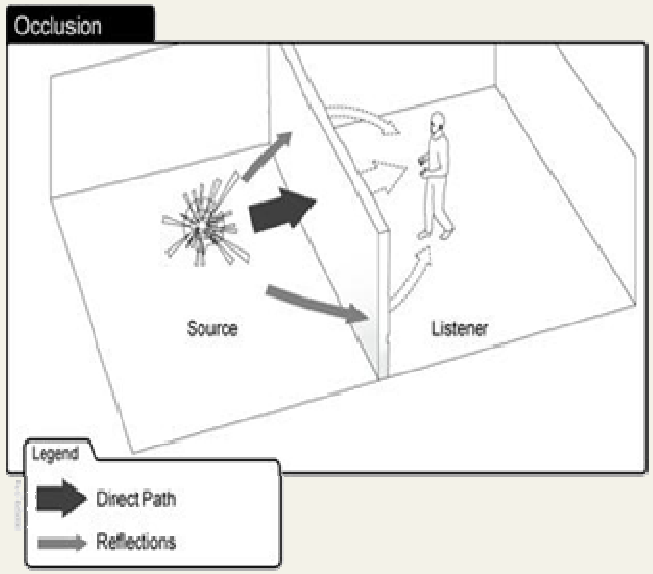
Audio

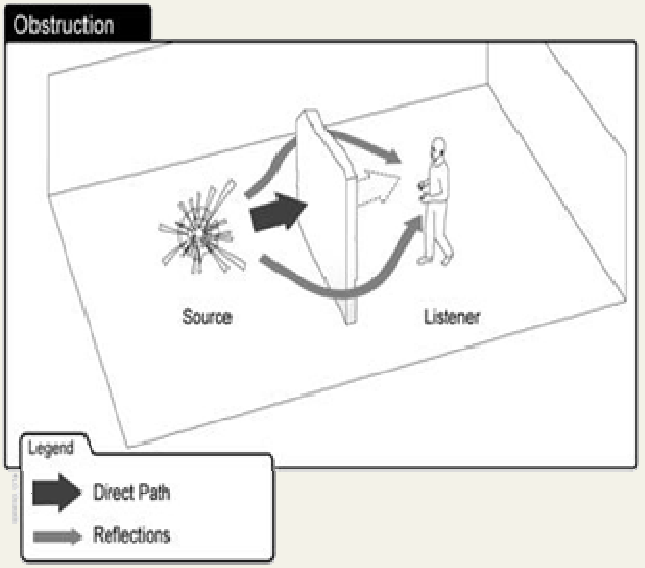
- Audible sphere



Audio

- Occlusions and obstructions





L'IHM au centre du débat

- Constat :
  - L'informatique envahit la vie:
    - ⇨ PC, distributeur de billet, borne de réservation (avion-train), téléphone portable, voiture, ...
  - au travail : réseau, bureautique professionnelle
    - ⇨ systèmes embarqués, etc.
  - Or, tout le monde n'a pas les mêmes capacités
- Questions :
  - est-ce important d'avoir une bonne interface ?
  - que doit-on attendre d'une bonne interface ?

IHM : I comme ...

- **Interface** ?
  - Ne désigne que le vecteur de communication
  - Les boutons, les menus, les couleurs ou les animations ne suffisent pas à rendre un système *utilisable*
- **Interaction** ?
  - La séquence d'actions nécessaires pour accomplir une tâche
  - La réponse du système à une action de l'utilisateur
- **Termes indissociables**

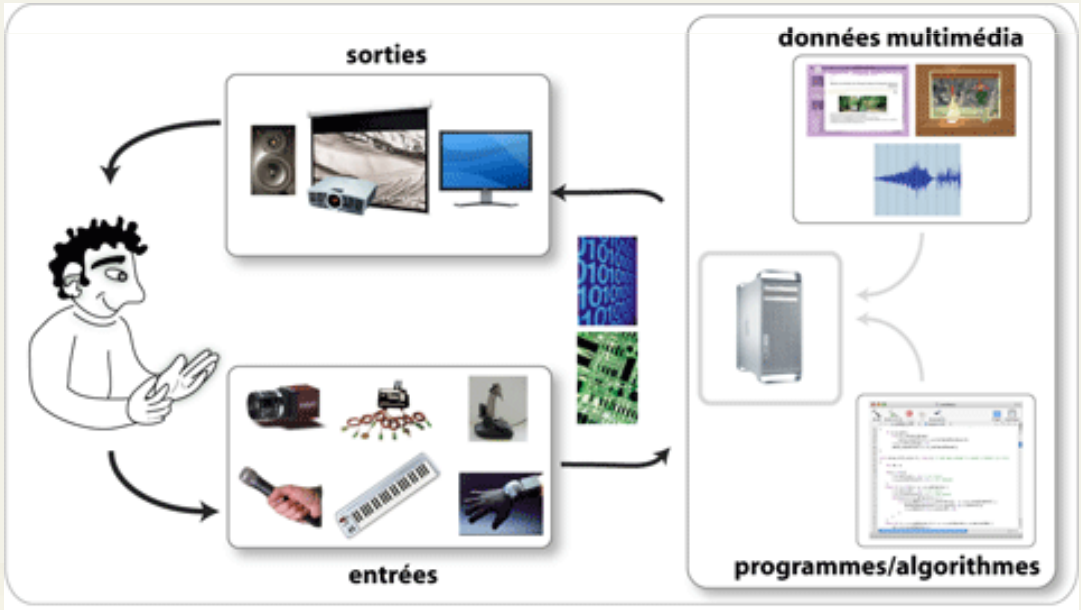
IHM : HM comme …

… comme Homme (être humain)

- des interfaces utilisables

… comme Machine

- des fonctionnalités utiles

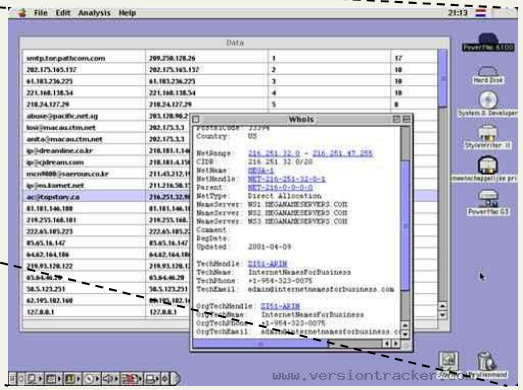


D'où la définition ...

- L'Interaction Homme-Machine

L'Interaction Homme-Machine est une discipline consacrée à la conception, à la mise en œuvre et à l'évaluation de systèmes informatiques interactifs destinés à des utilisateurs humains ainsi qu'à l'étude des principaux phénomènes qui les entourent

Qu'est-ce que l'IHM



Importance des IHM

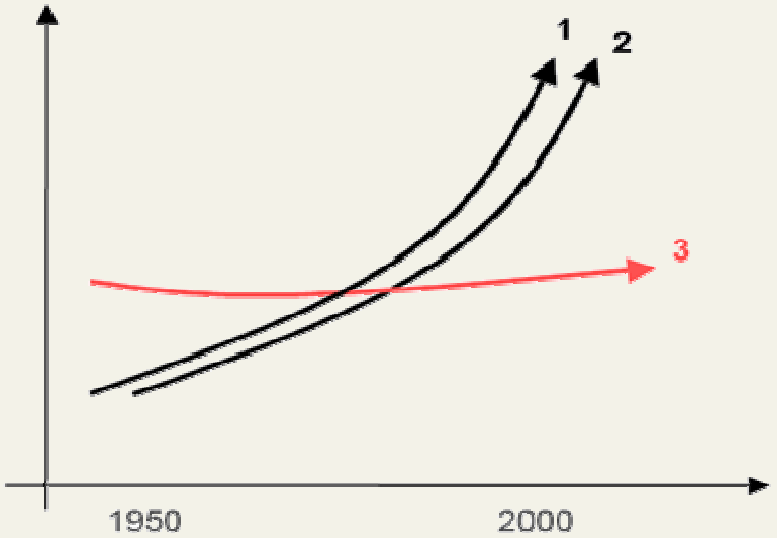
- Que veut-on d'une IHM ?
  - interface invisible, facilité d'utilisation ?
  - apprentissage ? affordance ?
  - universelle ?
  - évolutive (ajout de fonctionnalités), etc.
- Aspects socio-économiques (rejet, argument de vente) et aspects techniques (réalisation et utilisabilité)
- IHM = carrefour des compétences

Importance des IHM

1 - le matériel progresse sans cesse (Moore)

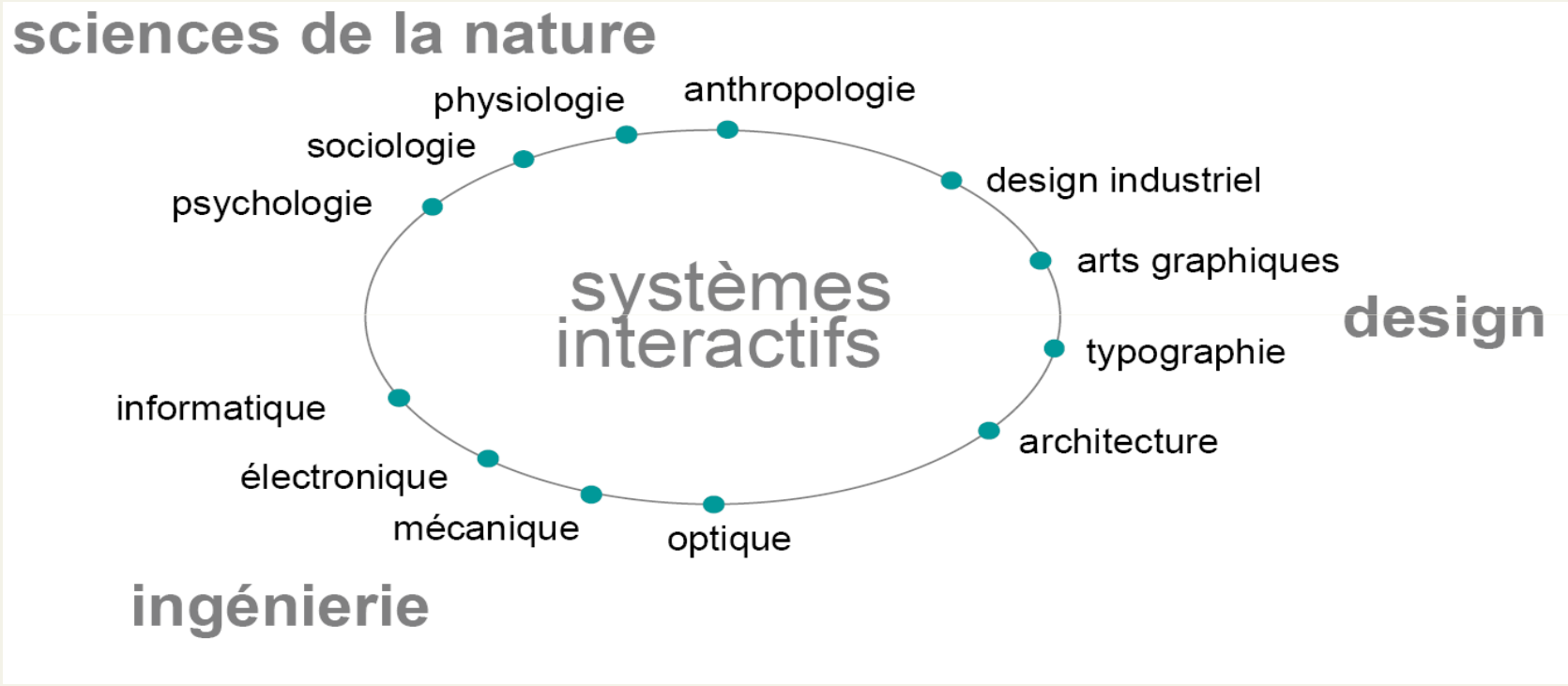
2 - les fonctionnalités promises aussi (Buxton)

3 - l'homme, lui, ne change pas, ou presque…



Limites des capacités de *perception* et d'*action* : le temps de la frustration !

Les acteurs de l'IHM



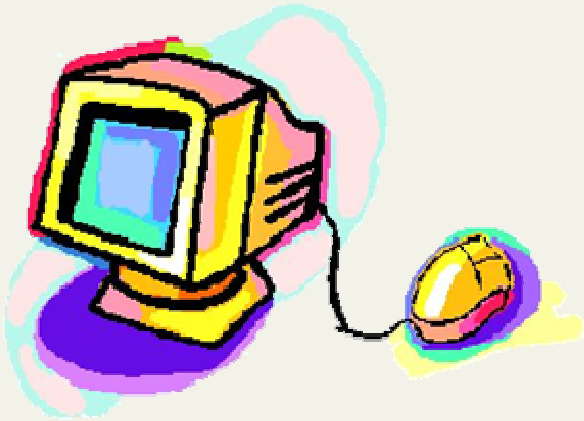
Interfaces WIMP

Windows (fenêtres)

Icônes

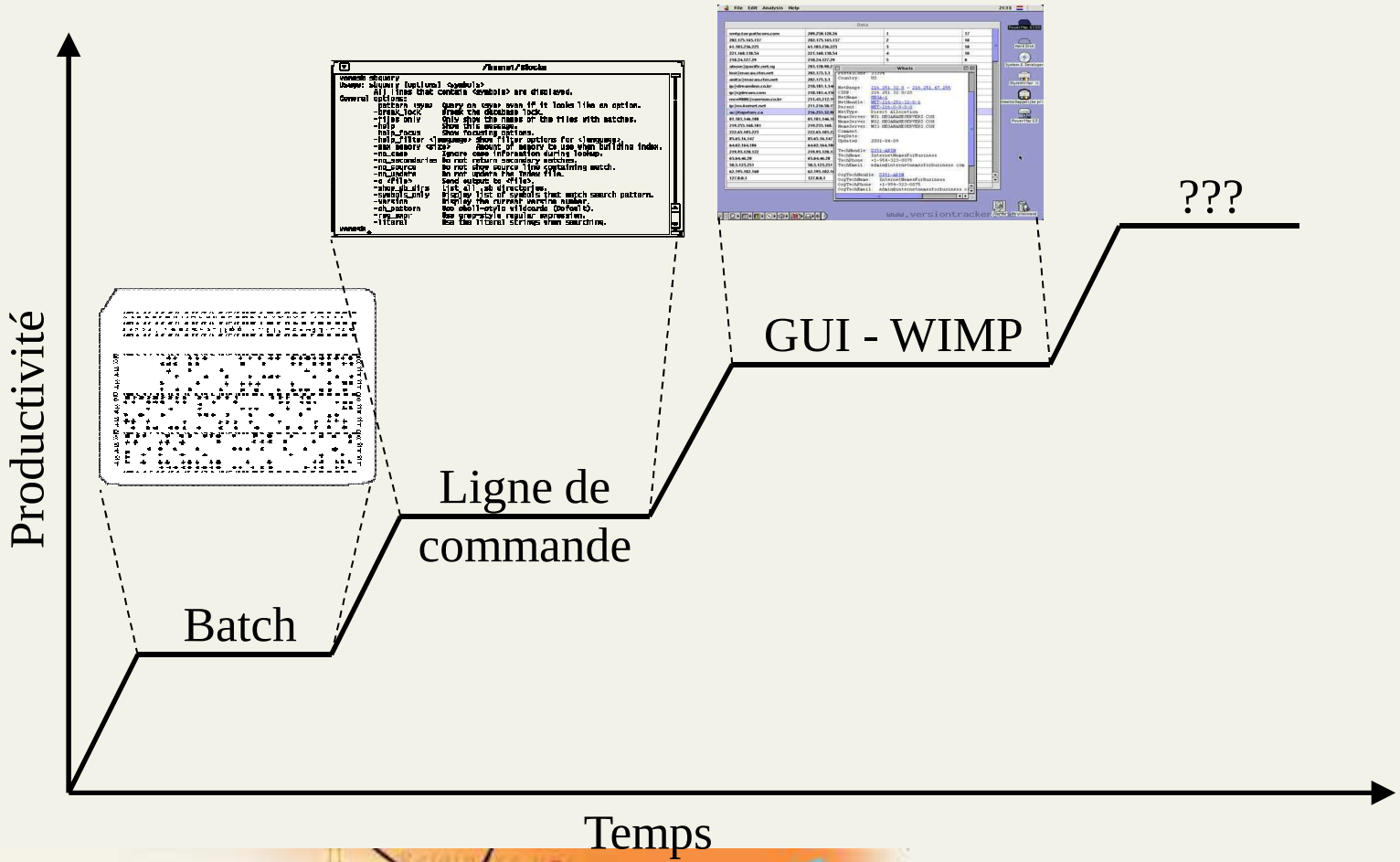
Menus, boîtes de dialogue, etc.

Pointage, sélection, tracé



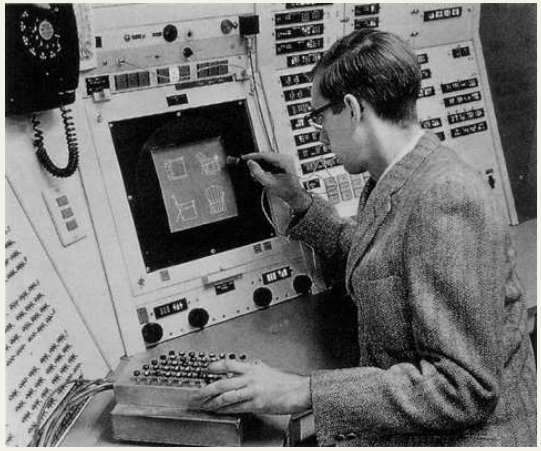
- Avantages :
  - apprentissage rapide
  - environnements standards
  - des techniques d'interaction simples à utiliser et à programmer
- Comment en est-on arrivés là en 1970 ?
- Comment en est-on restés là en 2010 ???

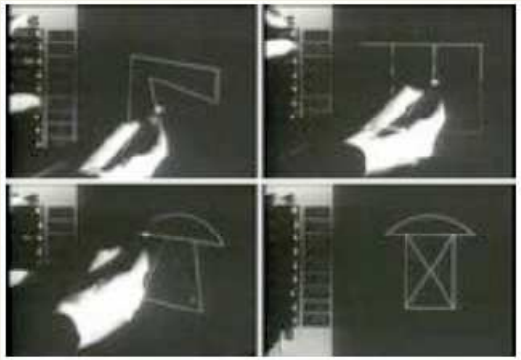
Repères (pré)historiques

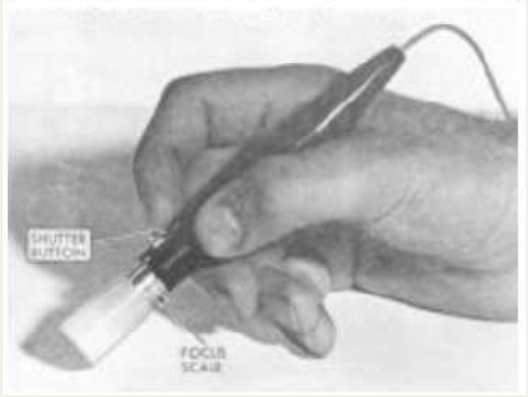


Repères historiques

- Années 60, deux inventions clés :
  - Sketchpad (Sutherland, ~1963)
    - ⇨ Écran cathodique et stylo optique
    - ⇨ Désignation directe des objets
    - ⇨ Techniques nouvelles : zoom, ruberband

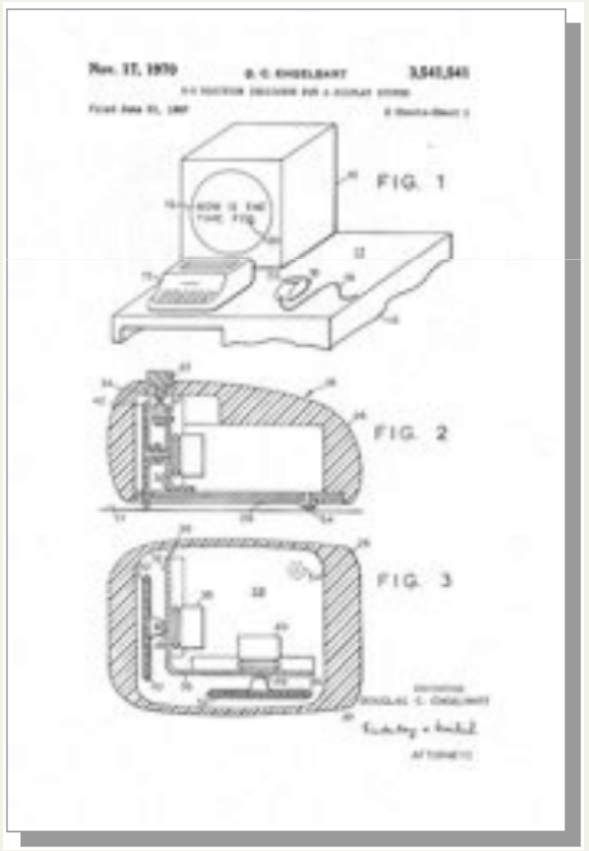






Repères historiques

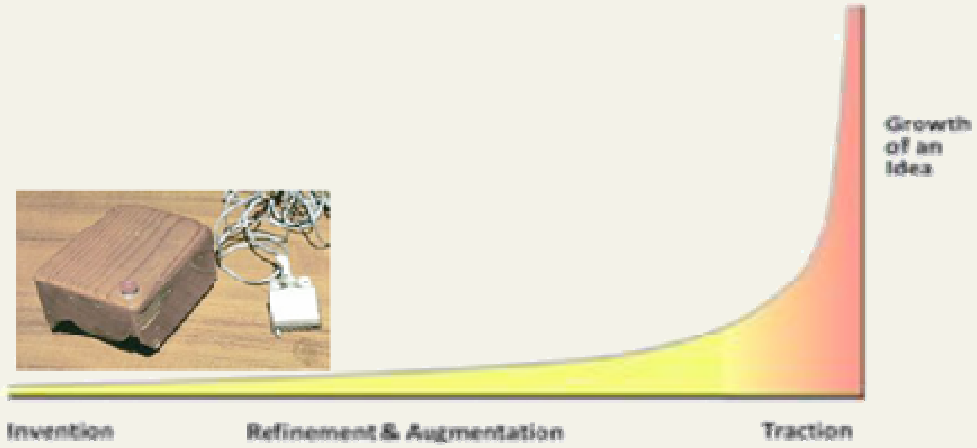
- Années 60, deux inventions clés :
  - Sketchpad (Sutherland, ~1963)
    - ⇨ Écran cathodique et stylo optique
    - ⇨ Désignation directe des objets
    - ⇨ Techniques nouvelles : zoom, ruberband
  - Souris (Engelbart, 1964)
    - ⇨ 2 roues : capter les mouvement sur 2 axes
    - ⇨ Pointage plus performant qu'avec d'autres dispositifs (en 1982) : joystick, clavier



Engelbart, Mouse Patent, 1970

Repères historiques

De bonnes idées peuvent prendre longtemps (30 ans !) pour mûrir…

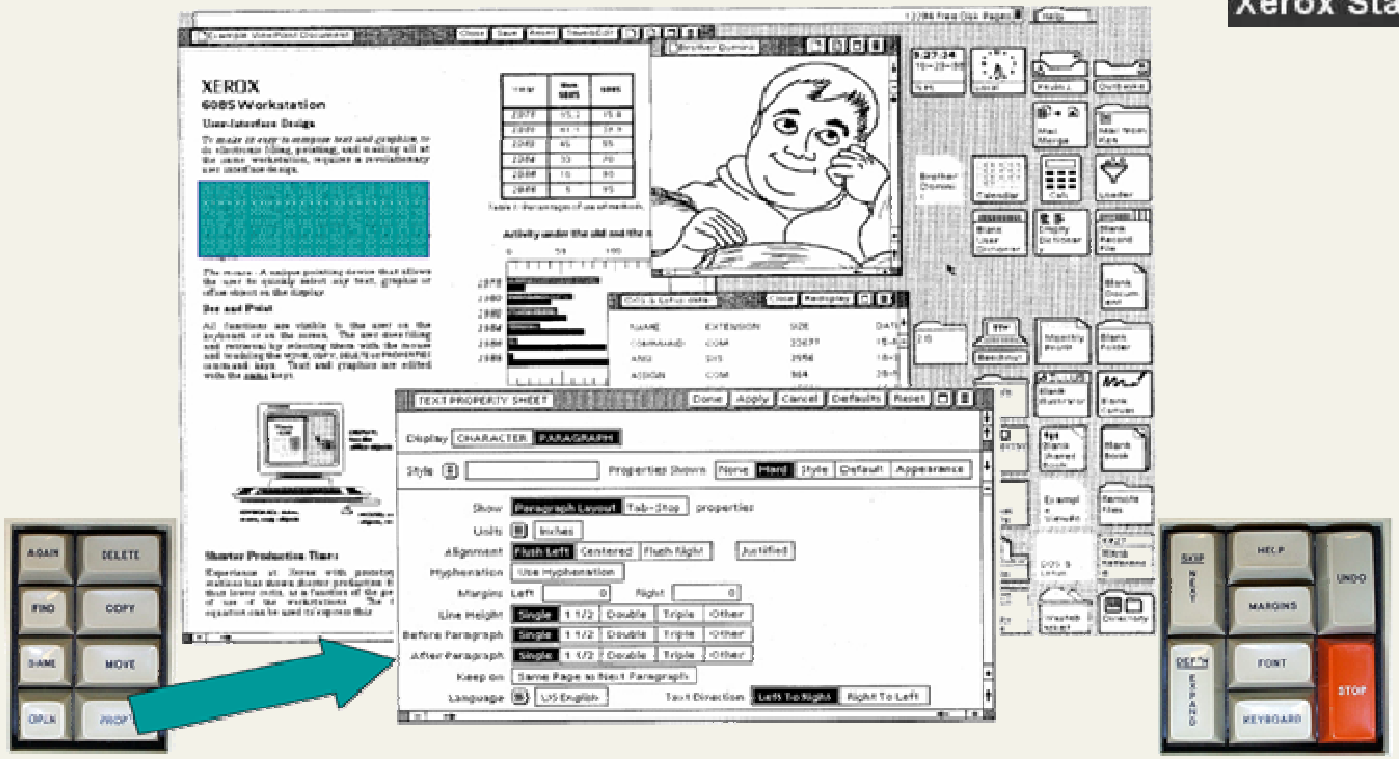


[Buxton (2008), *The long nose of innovation*]

(The Island, 2005)

Repères historiques

- Le Xerox Star (1975)



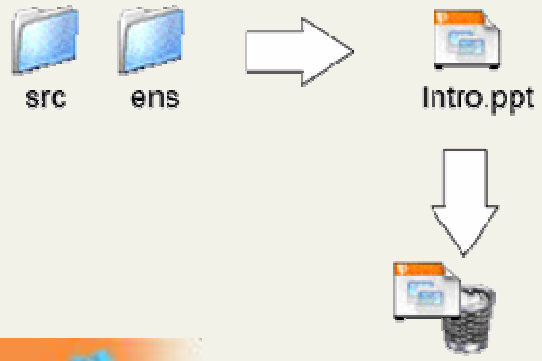
Principes

Manipulation directe (B. Shneiderman, 1983)

- représentation continue des objets et actions d'intérêt
- utilisation d'actions "physiques" (ex : pointer, déplacer) plutôt que des commandes à la syntaxe complexe
- opérations rapides, incrémentales et réversibles, dont les actions sur les objets sont immédiatement visibles
- Approche progressive pour faciliter l'apprentissage

"Point and click instead of remember and type"

```
% ls
src/ ens/
% ls ens/
intro.ppt
% rm ens/intro.ppt
```



Principes

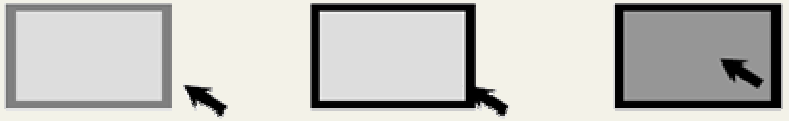
Couplage perception/action

- *Agir pour percevoir*
  - perception de la profondeur par des mouvements de la tête
  - perception de la texture d'un objet en déplaçant le doigt sur sa surface
- *Percevoir pour agir*
  - ajuster les mouvements du bras pour saisir un objet
- Importance de la continuité de ce couplage

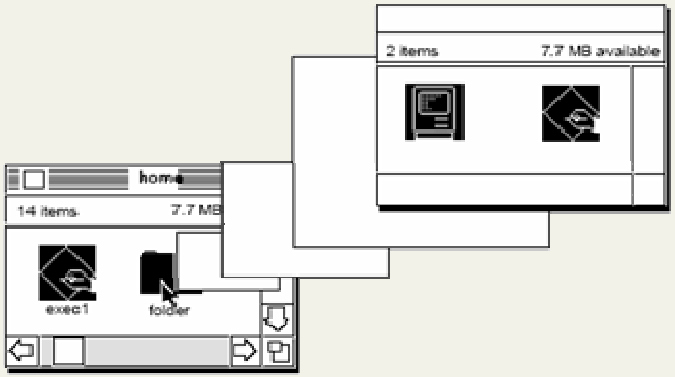
Principes

Le *feedback* (retour d'information)

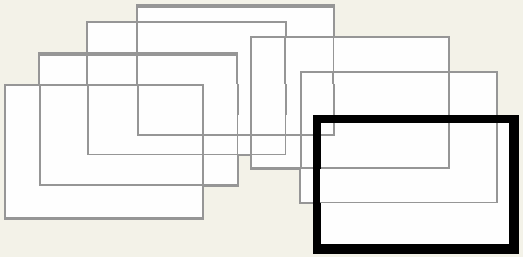
- Pointage

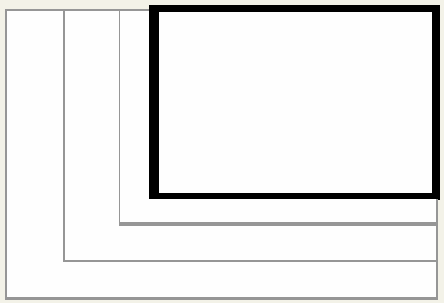


- Sélection



- Tracé, déplacement, transformation





Peu de choses ont changé depuis Xerox Star

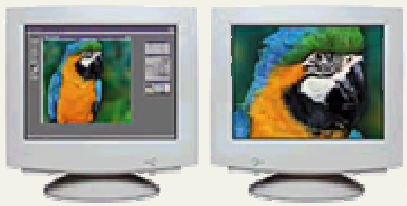
- Augmentation de la surface d'affichage

9", 512x342 pixels

23", 1920x1200 pixels

Facteur x4

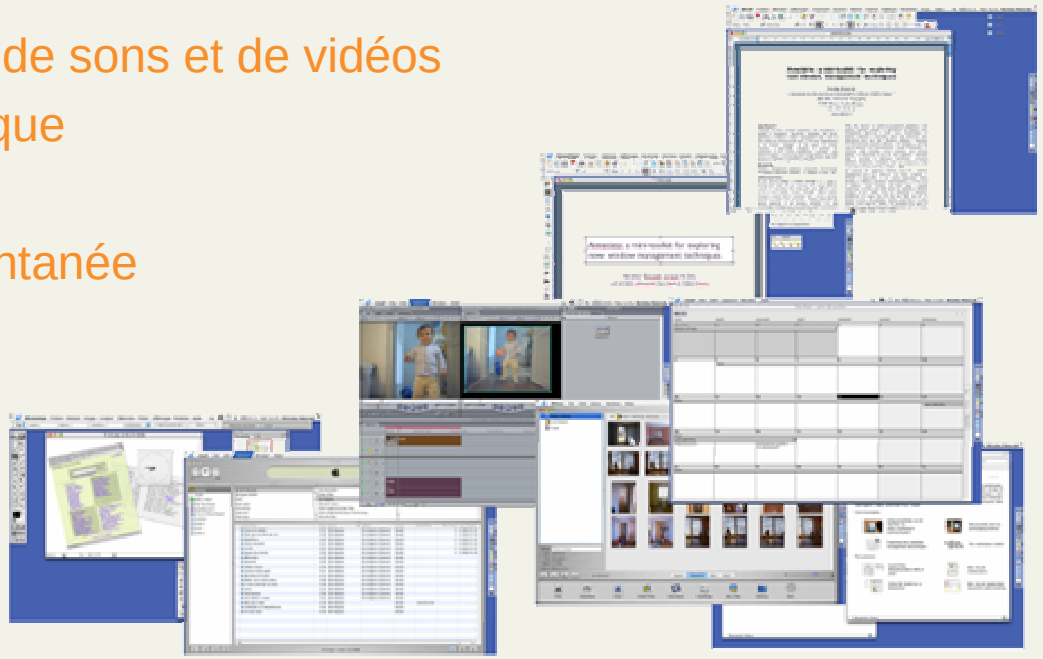
- Utilisation de multiples écrans réels



.. Et pourtant beaucoup de choses ont changé

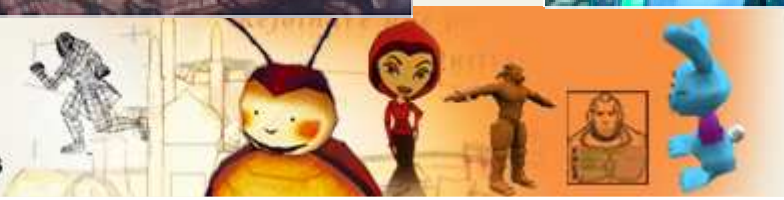
- Les activités se multiplient
  - traitement de texte, tableur
  - jeux
  - édition d'images, de sons et de vidéos
  - courrier électronique
  - Web
  - messagerie instantanée
  - *media players*
  - etc.

Stockage x10^6



Pourquoi n'en est-on pas là …

- Réalité virtuelle : Le Cobaye
- Interfaces gestuelles : Minority report, Avatar
- Bureau de travail du futur : The Island



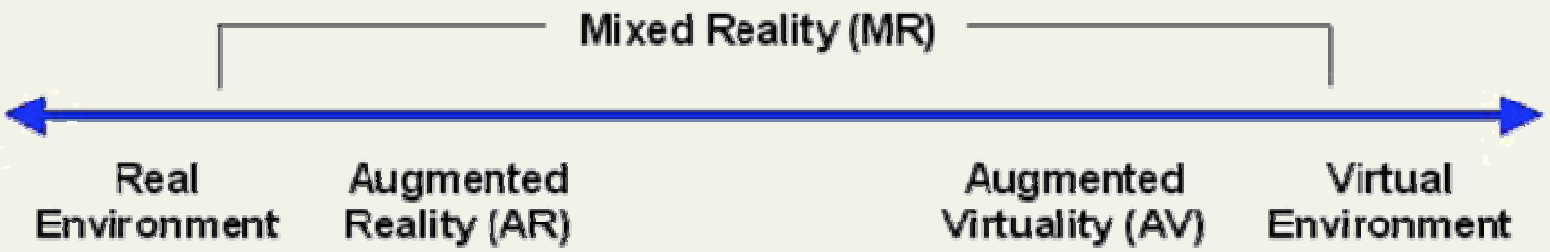
Patience ... que de progrès déjà

- Stéréoscopie
- Tactile
- Haptique
- Réalité virtuelle
- Réalité augmentée
- Interaction tangible

(Hinckley et al., *Passive Real-World Interface Props for Neurosurgical Visualization*, 1994)

Plusieurs successeurs à WIMP

- Le paradigme WIMP est unique et couvre un large spectre
- Plusieurs solutions envisageables car
  - Un outil est tâche dépendant
  - Un outil est contexte dépendant

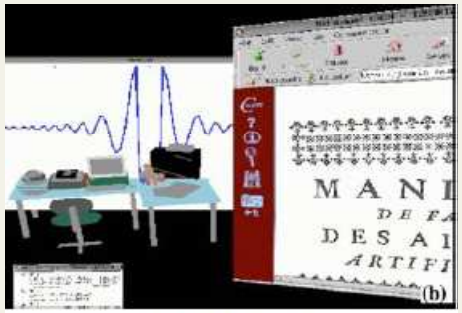


[Milgram and Kishino, 1994, *Mixed reality continuum*]

Plusieurs successeurs à WIMP

- Exemple : la manipulation de documents

[Card et al. 1996]



[Topol, 2002]

[Chapuis et Roussel, 2005]

Démo

Plusieurs successeurs à WIMP

- Exemple : le biofeedback pour les jeux vidéo