

# Animation d'algorithmes

Projet de Master 1 --- Olivier Ridoux

# Objectif

- comprendre des algorithmes
- **voie déductive**
  - preuves de programme
  - logiques de programme
  - invariants
- **voie inductive**
  - test
  - trace
  - animation

# Comment ?

- ❑ faire des expérimentations d'algorithmes
- ❑ algorithme  $\neq$  entrée-sortie
- ❑ automatiser la visualisation des étapes du calcul

→ ANIMATION ←

- ❑ langage d'algorithmes, de données, d'animations
- ❑ rôles multiples
- ❑ service en ligne

# Scénarios

## ☐ polycopié dynamique

### - l'enseignant

un algorithme, une animation, une donnée

### - l'étudiant

regarde,

change la donnée, l'animation, l'algorithme

*what if?*

## ☐ algorithmique

### - l'enseignant

donne une spécification

### - l'étudiant

rend un algorithme

## ☐ recherche

### - le praticien

fait des expériences (*joue avec les algorithmes*)

# Historique

- ❑ 2005-2006
  - objectifs bridés
  - prototype encourageant
  
- ❑ 2006-2007
  - débrider le prototype

# Version 2005-2006

- ❑ langage d'algorithme
  - le plus simple possible
  - séquence, WHILE, FOR, COND, affectation
  - «L'omnibus de Turing»
  
- ❑ langage de données
  - le plus riche possible
  - scalaires, paires, listes, arbres, ensembles, ...
  
- ❑ infrastructure Web
  - client-serveur, applette
  - grand ouvert

## Version 2006-2007

- ❑ développer le langage d'algorithmes
  - structures de données
  - récursivité
  - multi-processus
  - assertions
  
- ❑ développer les animations
  - chronogrammes
  - cumuls
  
- ❑ développer des rôles d'utilisateur
  - administrateur, étudiant, enseignant, visiteur, ...
  - authentification, autorisation

**déployer le système à l'IFSIC !**

# Concepts et méthodes

- ❑ compilation
  - **génération de code, machine virtuelle**  
pour un **tout petit** langage
- ❑ application Web
  - **client-serveur, applette**
- ❑ animation d'algorithme
  - **voir le web**
- ❑ *extreme programming*
  - **cycle de vie itératif**  
court, incluant **test** et **intégration**
  - **travail coopératif**  
binômes reconfigurables  
code public

# Challenge

- ❑ réutiliser !
- ❑ prendre en compte les algorithmes de «L'omnibus de Turing»