

Profile

Preference

J-VOTING

Par Julien, Léo, Thomas, Pierre et Jade

Voter



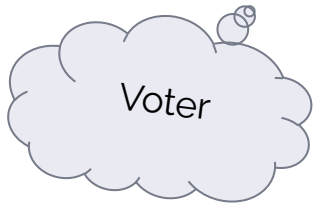
Alternative



M. B :



Mme A :



Madame A

Monsieur B

Madame C

Shrek



iPhone 11

Raclette



Il existe plusieurs types de préférences et donc de profils !

Those are Alternatives

This is a Preference

This is a Voter

This is a Profile



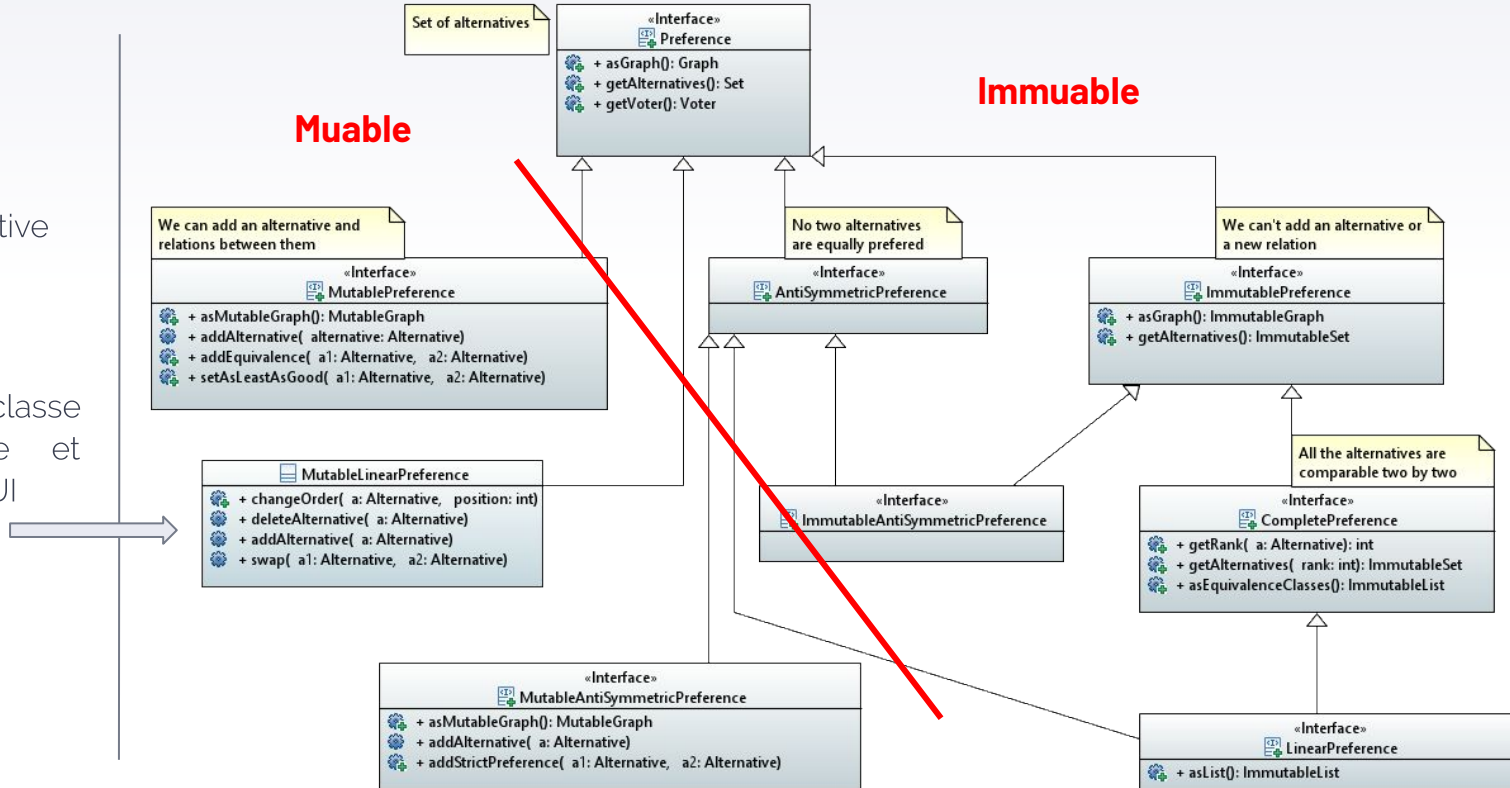
Préférence

Une préférence c'est :

- Un voteur
- Une liste d'alternative

Objectif :

Créer une nouvelle classe `MutableLinearPreference` et l'implémenter dans le GUI



OldCompletePreference to CompletePreference

1ère mission

Problème :

2 classes qui se chevauchent.
CompletePreference plus pertinent dans le
contexte du projet.

Enjeux :

Adapter le code et factoriser les méthodes pour
remplacer OldCompletePreference par
CompletePreference.



Bilan :

Itération 1 et 2; Travail non abouti
mais bien avancé.

MutableLinearPreference

2ème mission

Enjeux :

► Implémenter 4 méthodes pour le GUI

- ▶ addAlternative()
- ▶ deleteAlternative()
- ▶ swap()
- ▶ changeOrder()



► S'inscrivant dans l'architecture actuelle

► Utiliser cette classe dans le GUI

► Protéger la structure : Implémentation de decorators

Bilan :

4 itérations :

- ▶ Classe fonctionnelle
- ▶ Tests pertinents.
- ▶ Classe approuvée dans master.



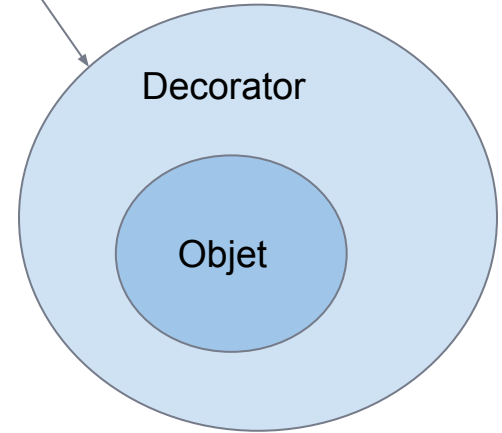
MutableLinearPreference	
	+ changeOrder(a: Alternative, position: int)
	+ deleteAlternative(a: Alternative)
	+ addAlternative(a: Alternative)
	+ swap(a1: Alternative, a2: Alternative)

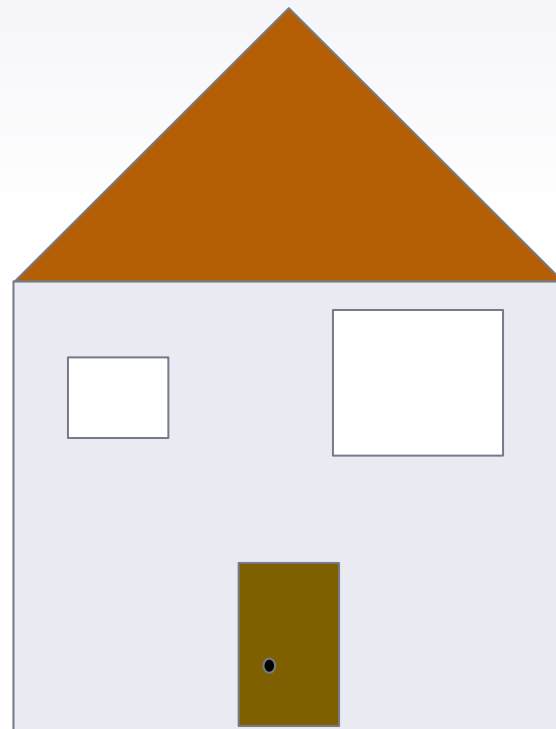
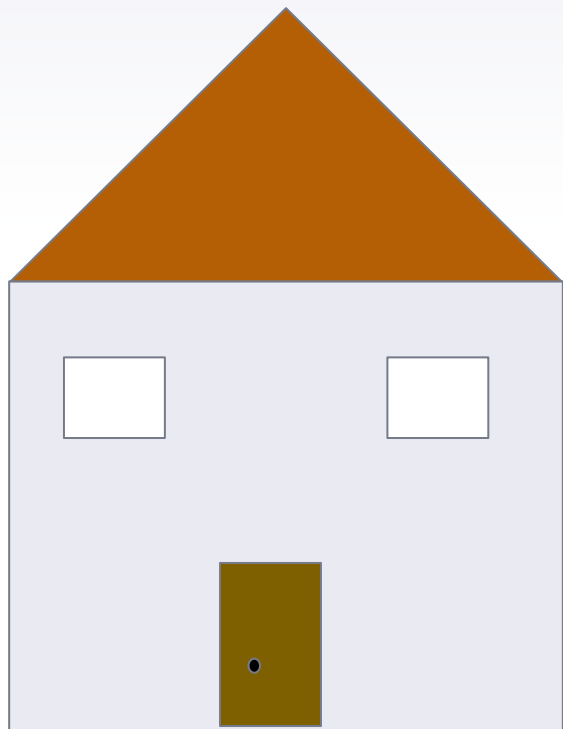
Qu'est-ce qu'un decorator ?

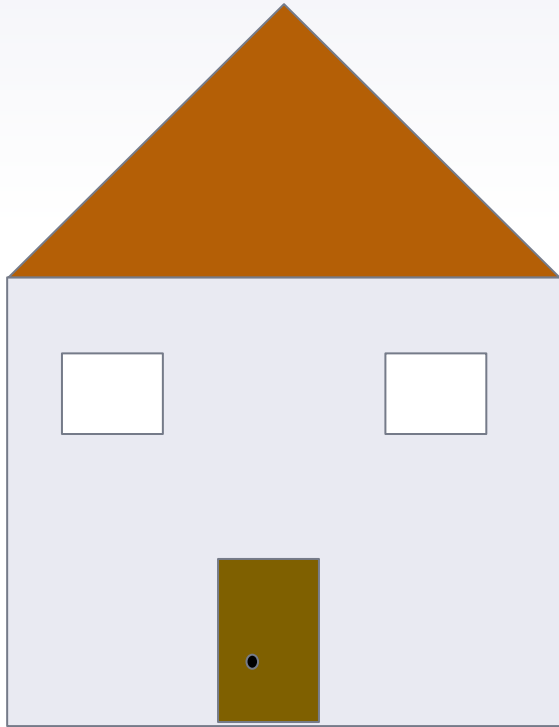
empêche l'utilisateur de
casser la structure



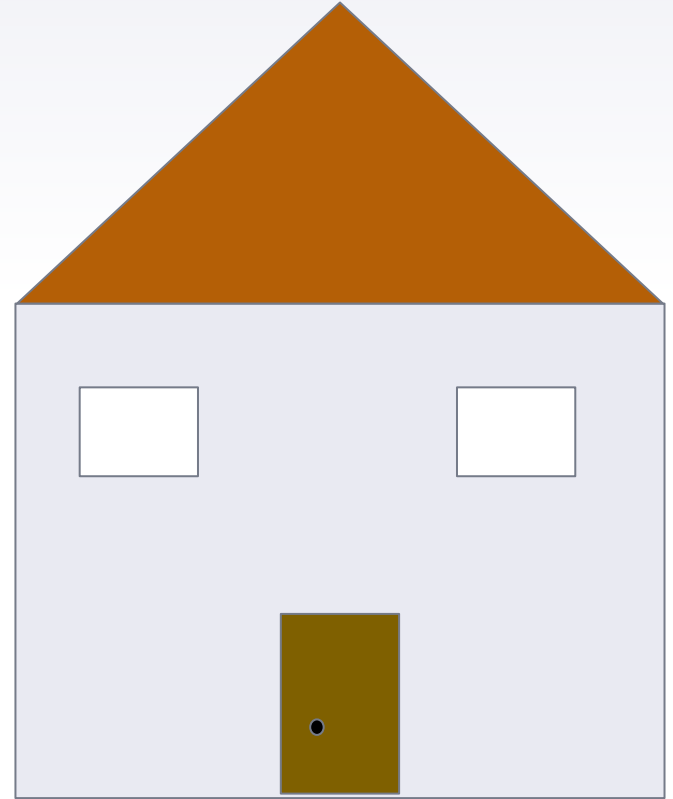
enrobe l'objet







`.addAlternative(x)`



Qu'est-ce qu'un Profile ?

Différents types de Profiles :

	Strict	Non Strict
Complet	$v1 : a1 > a2 > a3$ $v2 : a2 > a1 > a3$	$v1 : a1 = a2 > a3$ $v2 : a2 > a1 = a3$
Incomplet	$v1 : a1 > a2 > a3$ $v2 : a1 > a3$	$v1 : a1 = a2 > a3$ $v2 : a2 > a1 = a3 > a4$

- Nouvelle architecture des Profiles
- Besoin d'un Profile Strict Complet



Un Mutable Strict Profile ?

- Profil conçu pour s'insérer dans notre GUI.
- Une `Map<Voter,MutableLinearPreference>`.
- Deux `BiMap` pour pouvoir nommer voters et alternatives.
- Reflète les fonctions désirées dans le GUI.

Mutable Strict Profile

voter 1 : $a1 > a2 > a3$

voter 2 : $a2 > a1 > a3$

voter 3 : $a3 > a2 > a1$



L'univers des GUI dans J-Voting

Objectifs de cette section :

Vous **GUI**der dans l'interface graphique de J-Voting...

...Comprendre les ai**GUI**llages qu'il y a eu dans nos choix techniques...

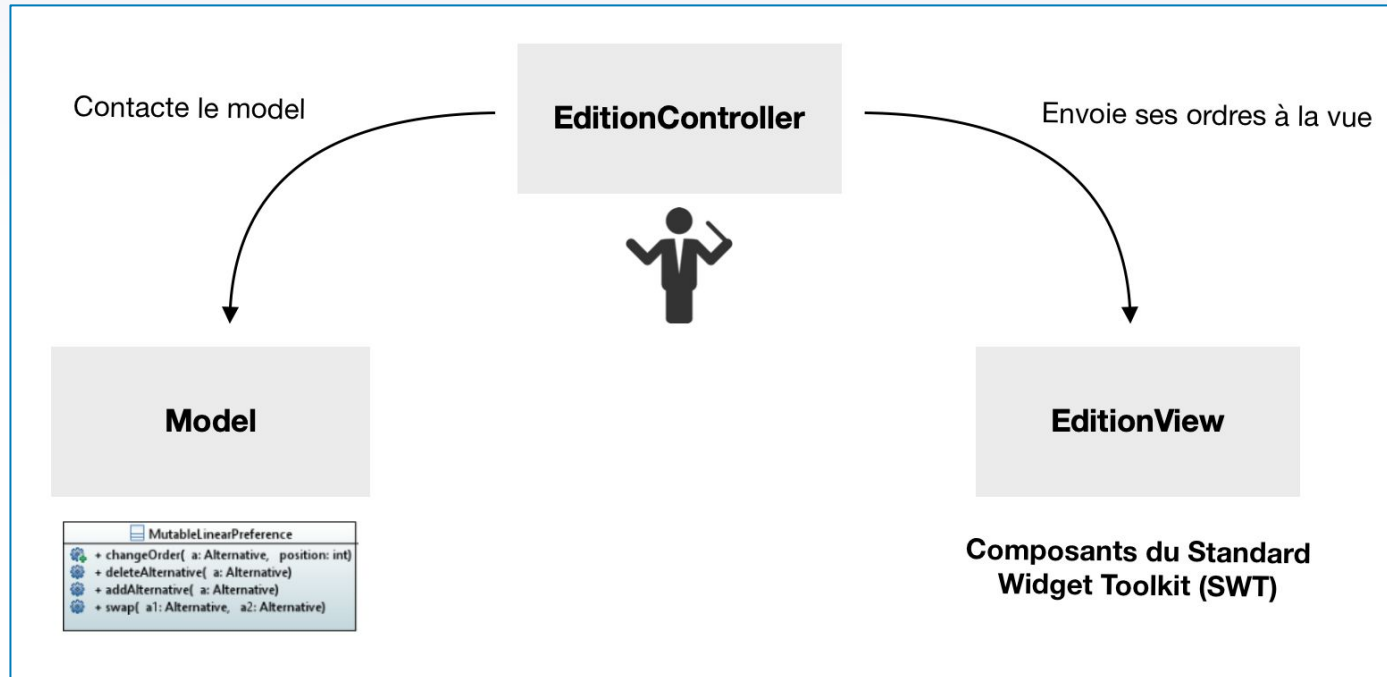
...Découvrir comment on a ai**GUI**sé l'architecture de l'interface...

La démo

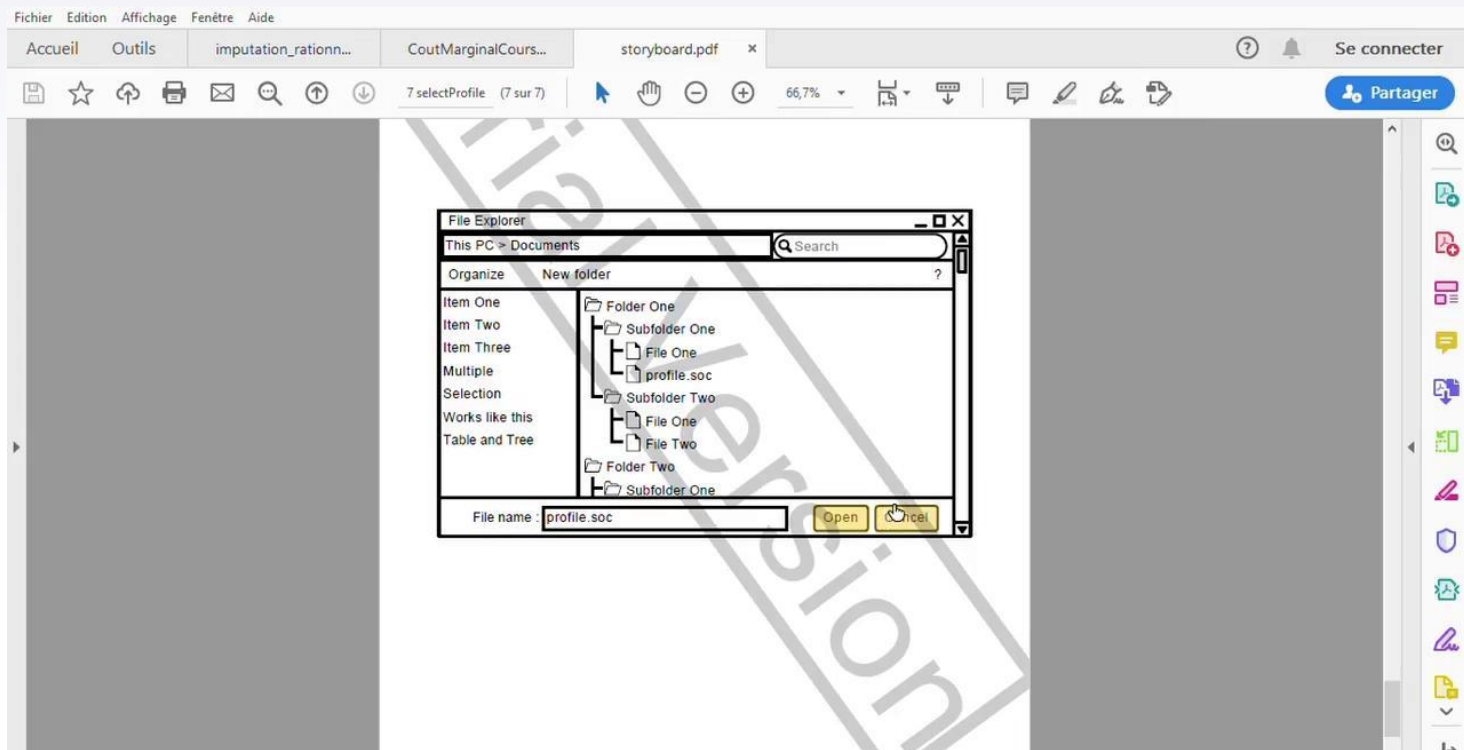
► Démonstration du GUI



L'architecture Model View Controller



Sketch



Perspectives d'avenir



- Consolider les architectures des préférence et des profils.
- Finir la version du GUI permettant la visualisation des données.

Le GUI au long terme

- Visualisation des données
- Analyses des données



Le GUI au long terme





Conclusion

- Merci pour votre attention