

---

# UE Projet L2 Informatique.

But du projet : Réaliser la modélisation d'un jeu, qui permettra d'implanter différents jeux ayant une base commune.

LEGROS Simon,  
RATEFIARIVONY Zo Tahina,  
SHARFELDIN Esam,  
MONCHEAUX Kilian.

---

# Avancement du projet.

- Modélisation complète.
- Implémentation des méthodes complète.
- Documentation (Javadoc) complète.
- Tests unitaires complets.

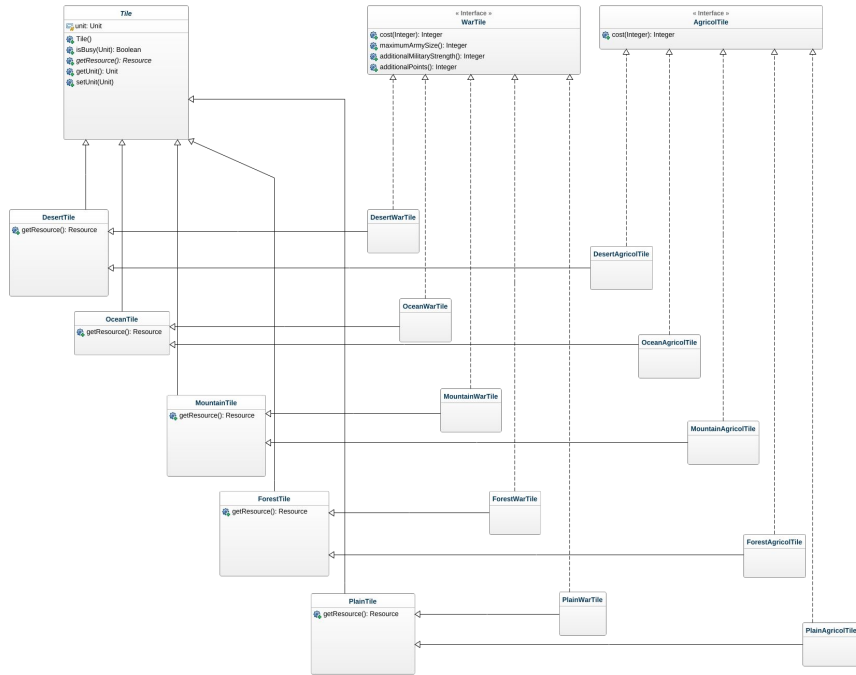
---

# Modélisation (1) : des actions.

- Choix de la modélisation des actions
  - En utilisant une interface
    - Définir plus facilement différents types d'actions
- Le développement d'autres jeux
- La logique du jeu
  - Ce choix permet de traiter toutes les actions de la même manière  
au niveau de la la logique

# Modélisation (2) : Les Tuiles.

## Première modélisation :



- Respecter le principe ouvert-fermé.
- Modélisation avec interfaces pour tuiles de chaque jeu.
- Trop Compliquée.

---

### Modélisation finale des tuiles :

- Une classe abstraite Tile.
- Etendue par 5 classes concrètes.
- Utilisation d'attributs publics et statics.

---

## Extensibilité (1) : points positifs

- Nouveau mode de jeu ? ... juste écrire la logique dans `play()`
- classe abstraite Board générique : différents types de plateaux, avec leur propres caractéristiques
- Idem pour Unit, Player et GameException

---

## Extensibilité (2) : points négatifs

- Tuiles : les comportements spécifiques à chaque jeu sont accessibles dans tous les jeux
- Joueurs et unités : des classes qui pourraient être plus générales
- Ressources : le problème de l'ajout de nouveaux types de ressources

---

## Extensibilité (3) : Actions et stratégies

- Le choix des actions par le joueur est codé “en dur” dans les classes qui héritent de Player
- Étendre Player et ses dérivées pour ajouter de nouvelles stratégies
- Alternative possible : une interface Strategy, implémentées par des classes dont les joueurs peuvent posséder des instances
- Comment s’assurer de la légalité des actions ?



---

# Travail en équipe (1)

- Pas eu besoin de chef d'équipe.
- usage IDE collaboratif pour la modélisation
- git : bonnes pratiques

---

## Travail en équipe (2): tâches

- 2 sous-binômes A et B
- A chaque mise à jour de la modélisation, partage moitié-moitié entre A et B
- Flexibilité entre A et B

Tâches de A	Tâches de B
doc de X	doc de Y
implémentation de X	implémentation de Y
tests de Y	tests de X

---

## Difficultés (1) :

- Difficultés autour du travail à distance
- Visualisations de la modélisation à partir du cahier des charges

---

## Difficultés (2) :

- Recherche de compromis entre une modélisation idéale et une modélisation pratique : exemple des tuiles
- Modélisation en partant des sous-éléments : difficultés à cerner les besoins en partant du bas, et nécessité de modifications en cours de développement
- Ecriture des tests des classes abstraites