

Projet Logiciel



Réaliser une modélisation permettant
la création d'un jeu

Equipe :

Aziz BOURAHAMA

Benedictus Kent RACHMAT

Dorian ABRAHAM

Corentin DUVIVIER

Aperçu des jeux

Réalisation d'une modélisation de jeu qui permet à un développeur de créer un nouveau jeu en se servant de ce qui est déjà réalisé ou en ajoutant modérément des classes pour s'adapter simplement aux spécificités de son jeu .

Un jeu de développement agricole

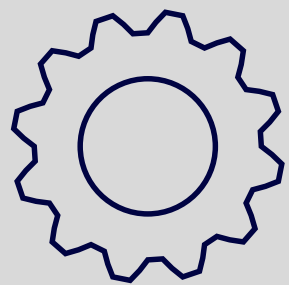
Dans ce jeu les joueurs ont pour objectif de gagner de l'or en déployant des ouvriers agricoles qui exploitent des territoires.



Un jeu de guerre

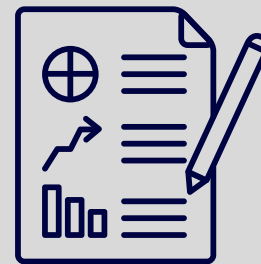
Dans ce jeu les joueurs ont pour objectif de déployer des armées sur des territoires.

L'état de notre projet :



Code

Les différentes parties du projet ont bien été traitées.



Documentation

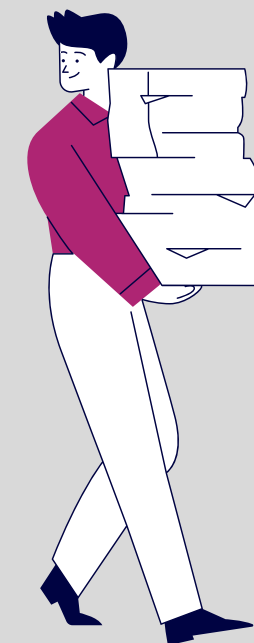
Présence :

- D'un readme à jour.
- De la javaDoc/Commentaires.



Tests

Le code a bien été couvert de tests.

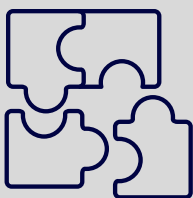


Deux hiérarchies de modélisation :



A l'aide de l'héritage et des classes abstraites :

- Une classe abstraite regroupant les comportements en commun.
- Des classes héritant des spécificités de chaque jeu.



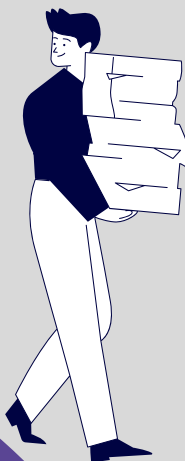
A l'aide d'une interface :

- Une interface qui définit une méthode.
- Des classes qui implémentent cette dernière.



Organisation en équipe :

- Se fixer des objectifs à court terme.
- Je code tu testes, tu codes je teste.
- Travailler en sous-équipe de 2.
- Utilisation des branches.



yes!
✓





Difficultés rencontrées :

- L'utilisation du gestionnaire de version Git en équipe.
- Opinion différente entre les membres.
- Modification de code pouvant affecter une autre partie.
- OS



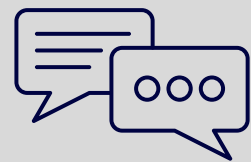


Analyse du contenu de notre projet :

Tout au long de ce projet, nous avons veillé à :

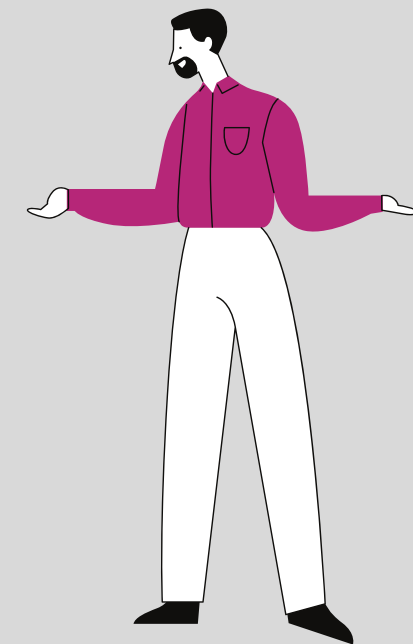
- Des noms parlants pour les paquetages, classes et méthodes, des noms qui nous apprennent quelque chose de façon plus générale.
- Simplifier les codes en sous-méthodes.
- Identifier/supprimer le code dupliqué.
- Répartition judicieuse des classes en paquetages.

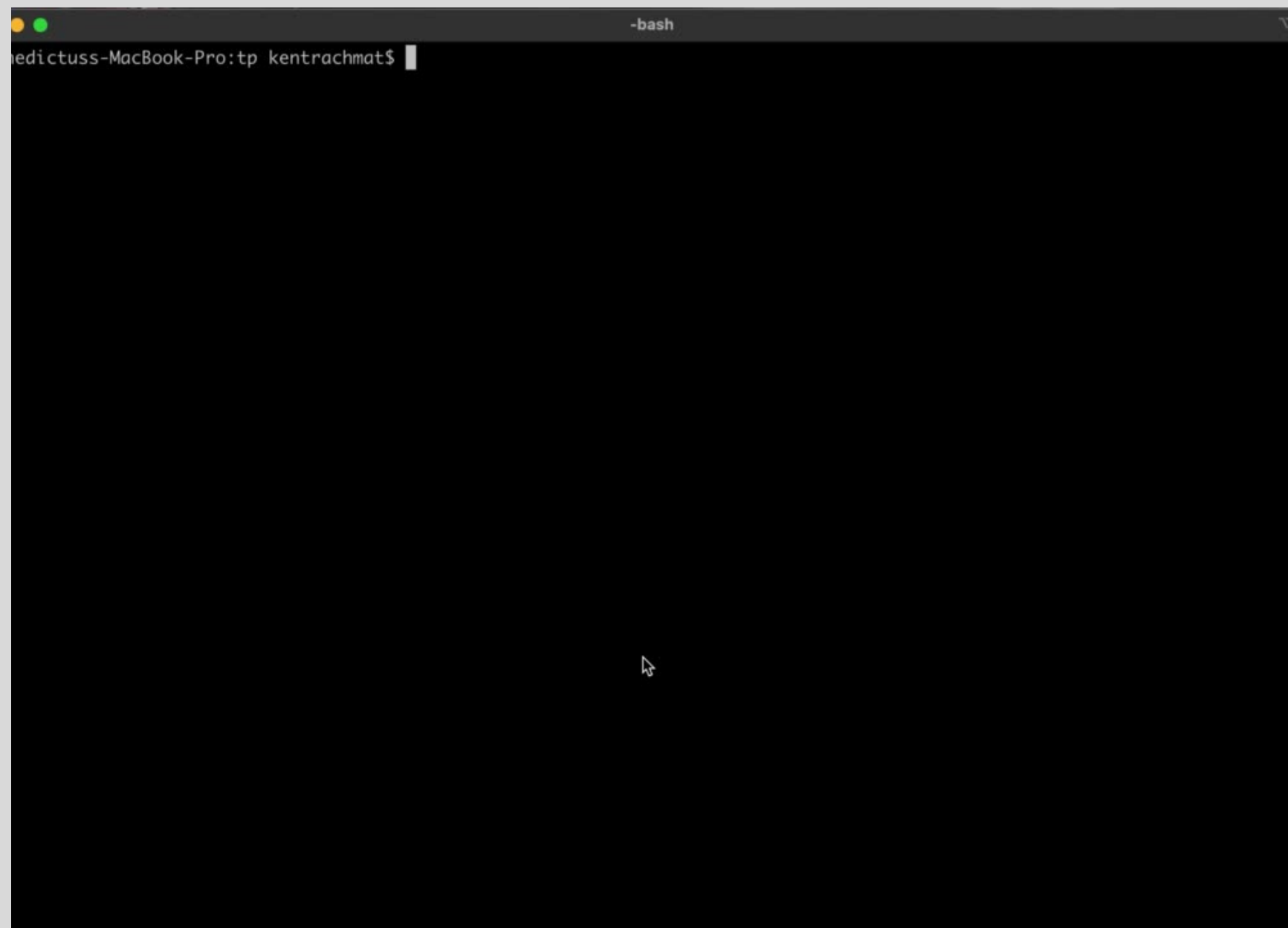




Tout à l'air parfait alors : Non, presque !

- Mieux définir la méthode equals pour les tuiles.
- Mieux faire le parallèle avec le cours d'ASD, la méthode hashCode.
- Choisir la bonne structure de données c'est bien, mais choisir l'idéale ça l'est encore plus.





Petite
partie de
jeu de
guerre.



Envie de revoir notre diaporama

THANKYOU

Avez-vous des questions ?

