

# Soutenance Projet

Programmation avancée

Présenté par : Jérémy JAGUT, Samir ZENNANI, Chamir MOUSTAPHA-ABLOH et Melvin MOREAU

# Sommaire

- Travail effectué
  - Architecture général
  - Architecture des plugins
  - Plugins réalisés
- Gestion équipe
  - Outils et méthodes utilisés
  - Travail effectués par chacun
  - Problèmes rencontrés





Travail effectué

# Architecture générale

L'architecture est décomposé en 3 projets :

- Un projet parent lui même composé de deux projets.
  - Un projet "plugin" composé uniquement de plugins.
  - Un projet "app" comportant le coeur de l'application.
- 
- Un classLoader
  - On a un Outil de Reflection.



# Architecture des plugins

- 2 @interfaces:
  - Plugin.class
  - PluginInfo.class
- Un plugin est une classe avec l'annotation @Plugin
- Un plugin possède un nom, un type et un typeOf
- TypeOf :
  - Déplacement
  - Attaque
  - Graphisme
  - Core
- Méthode de plugin identifié par @PluginInfo(name)



# Fonctionnement et utilisation des plugins

- MyClassLoader charges les fichiers avec l'annotation @Plugin
- On construit une map de tous les plugins chargés.
- OutilReflection permet de construire une instance d'un plugin et d'utiliser ses méthodes.



# Plugins réalisés

- Plugins Graphique :
  - RobotColor : Permet d'affecter une couleur à un robot
- Plugins Attaque :
  - AttaqueCourte : Permet au robot d'effectuer une attaque courte
  - AttaqueLourde : Permet au robot d'effectuer une attaque lourde
  - AttaqueDistance : Permet au robot d'effectuer une attaque à distance
- Plugins Déplacement :
  - RandomMove : Permet au robot de bouger de façon aléatoire
  - AvantEtArriereMove : Permet au robot de se déplacer d'avant en arrière



# Gestion d'équipe



# Outils et méthodes utilisés

- Gestion des sources :
  - GitHub
- Gestion de build :
  - Maven
- Moyen de discussion :
  - Messenger
- Logiciel de Développement :
  - Eclipse
  - Sublime Text



# Travail effectué par chacun

- Melvin :
  - Réalisé une rapide conception du projet
  - Il a réalisé une version de template pour les plugins.
  - Développement et équilibrage de la partie plugin d'attaque de l'application.
  - Il s'est occupé des tests des plugins.
  - Il a pris en charge la rédaction du rapport, la réalisation de la diapositive et la réalisation du documentation
- Samir :
  - Chargé du developpement de la classe Grille
  - Developpement de la classe cellule
  - Calcul de la distance entre Robots pour permettre les attaques
  - Réalisation documentation JavaDoc



# Travail effectué par chacun

- Chamir :
  - Il a créé le projet Github et le projet Maven
  - Il a développé l'attaque qui permet à un robot d'absorber la vie de son adversaire.
  - Il a développé le plugin AvantEtArriereMove
  - Il a développé une méthode qui absorbe la vie d'un autre robot
  - La documentation Javadoc
- Jérémie :
  - Il a réalisé la version final du template des plugins.
  - Il a réalisé les mouvements random et l'affectation des couleurs aux robots
  - Il a fortement contribué à toute la partie affichage de l'application et au noyau de l'application.
  - Il a ajouté la barre de vie dans la fenêtre
  - Il a adapté la grille et la cellule pour l'ajouter dans la fenêtre.
  - Chargement dynamique des plugins.
  - Structuration en 3 projets (un parent et deux sous-projets)

# Problèmes rencontrés

- Organisation
- Manque de connaissances
- Manque de disponibilité
- Difficulté à exécuter le .jar



Des Questions ?

