

IA
TP1. Configuration pour les TPs d'IA-MDP

Laëtitia Maignon

Pour les TPs, vous devez récupérer et compléter le code fourni sur
<https://github.com/laetitiamatignon/tp-ia-mdp>.

Votre version modifiée de ce code sera déposée sur github en **DEPOT PRIVE** pour que l'enseignante puisse suivre et noter votre TP. Pensez donc bien à ajouter l'enseignante (username `laetitiamatignon`) comme collaboratrice de votre dépôt.

Les TPs se font en monôme.

Le dépôt <https://github.com/laetitiamatignon/tp-ia-mdp> contient :

- `./lib/IAMDPLib.jar` : bibliothèque pour affichage graphique et mdp
- `./javadoc` : javadoc de la bibliothèque IAMDP Lib
- dans le dossier `src`, des packages à compléter
- dans le dossier `simu`, des packages pour tester vos algorithmes dans différents environnements.

1 Installation de JavaFX

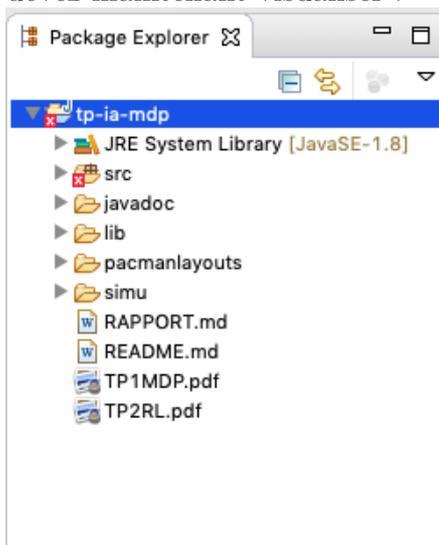
Le code et les bibliothèques fournies dépendent de JavaFX. Vous avez deux options pour l'installer :

- **la plus simple** est d'utiliser le JDK8 qui intègre JavaFX. Une version pour chaque OS est disponible ici : <https://perso.liris.cnrs.fr/laetitia.matignon/index/5ASMA/install>. Vous pouvez l'installer, et configurer votre projet pour qu'il l'utilise (cf. tutoriel ici pour **Eclipse** : <http://blog.paumard.org/tutoriaux/eclipse-projet/> ici pour **IntelliJ** : <https://www.jetbrains.com/help/idea/sdk.html>).
- utiliser une version plus récente du JDK (13 ou 14) et suivre le tutoriel JavaFX <https://openjfx.io/openjfx-docs/>.

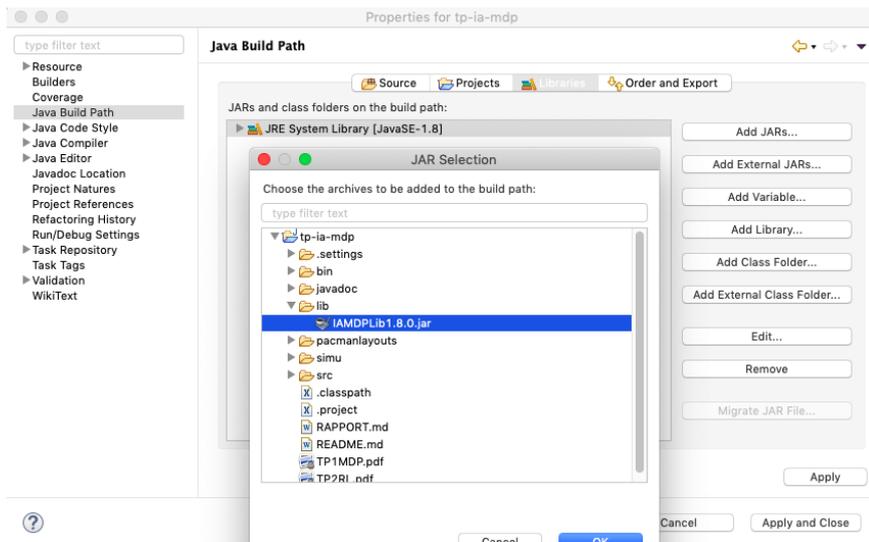
2 Mise en place sous Eclipse

2.1 Configuration du projet local sous Eclipse

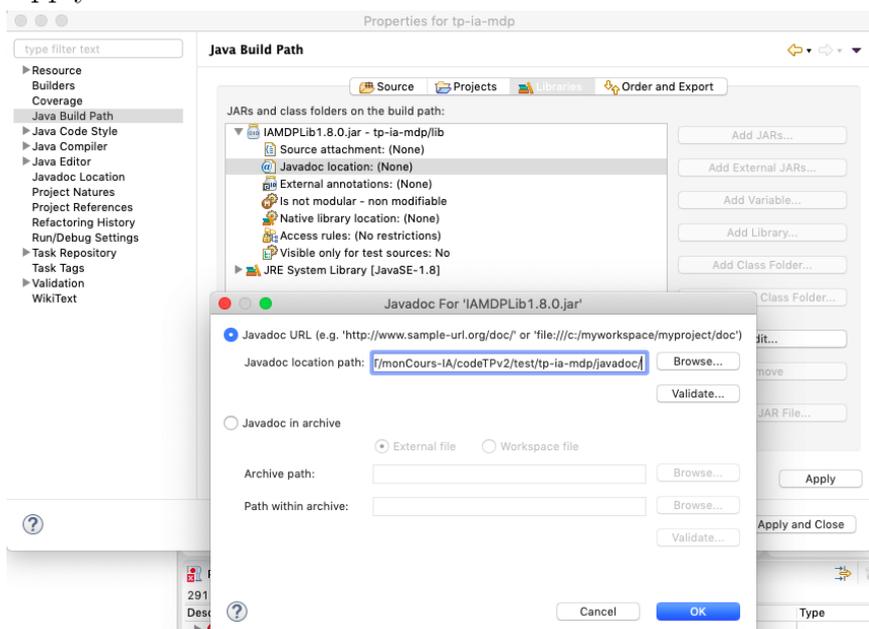
1. télécharger le projet sur <https://github.com/laetitiamatignon/tp-ia-mdp> : bouton vert "clone or download", puis download ZIP.
2. lancer eclipse et choisir votre workspace
3. créer un nouveau projet Java vide : File-> New->Java Project
4. donner un nom à votre projet dans Project Name (TP-IA-MDP par exemple), choisissez votre JRE (java8 si c'est celui que vous avez installé pour avoir JavaFX), cocher : "create separate folders for sources and class files" puis Next
5. Cliquer sur Finish
6. copiez-collez dans le dossier du projet qui vous venez de créer (TP-IA-MDP) **tous les dossiers de l'archive** que vous téléchargé (vous allez ainsi écraser le dossier src). Vous devez ainsi avoir dans le dossier TP-IA-MDP les éléments suivants : dossiers javadoc, lib, simu, src, et les fichiers RAPPORT.MD et README.MD.
7. dans eclipse, faites un clic-droit sur votre projet TP-IA-MDP et Refresh. Vous devez maintenant visualiser :



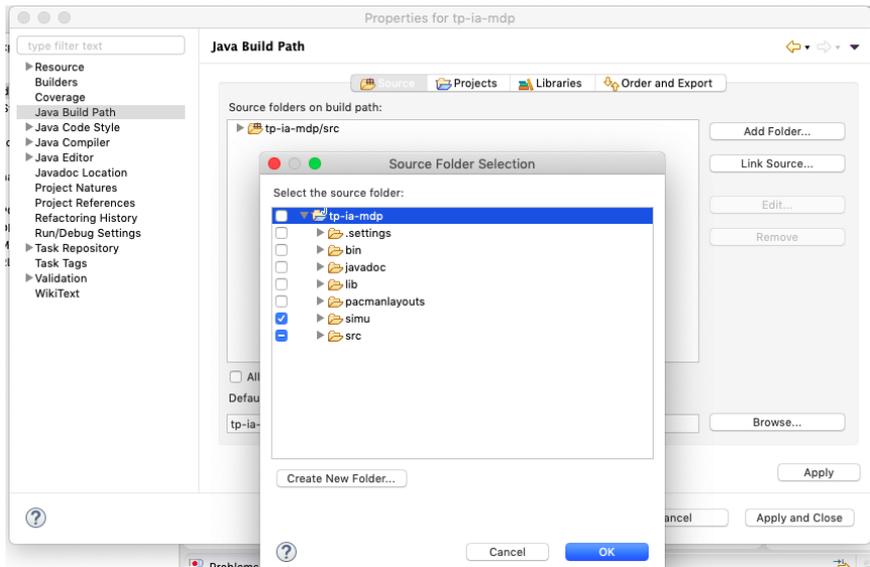
8. dans eclipse, faites un clic-droit sur votre projet TP-IA-MDP, puis Properties. Choisir l'onglet à gauche Java Build Path, puis l'onglet Libraries, et cliquer sur Add Jars. Choisir dans le menu déroulant votre projet TP-IA-MDP->lib->IAMDPLib1.8.0.jar, puis OK et apply and close.



9. dans eclipse, faites un clic-droit sur votre projet TP-IA-MDP, puis Properties. Choisir l'onglet à gauche Java Build Path, puis l'onglet Libraries. Dérouler l'élément IAMDPlib1.8.0.jar, double clic sur Javadoc Location, et dans Javadoc location Path entrer le chemin :TP-IA-MDP /javadoc/ Cliquez sur Open, OK, Apply and close.



10. Ajouter le dossier `simu` à votre projet comme un dossier source de votre projet : dans eclipse, faites un clic-droit sur votre projet TP-IA-MDP, puis Properties, , puis l'onglet *Source*, *Add Folder*, cocher le dossier `simu` OK, Apply and close.



11. Pour tester, faites un clic-droit sur le fichier qui est dans `simu/simuProjetPacman/PacmanGame.java`, RunAs->JavaApplications.

Votre projet fonctionne maintenant localement. La deuxième étape est d'ajouter ce projet dans GitHub pour permettre à l'enseignante de suivre votre travail et récupérer votre TP à la fin des séances.

2.2 Dépôt de votre projet dans GitHub (en PRIVE)

Suivre le tutoriel <https://www.youtube.com/watch?v=LPT7v69guVY> en :

- Step 2 : **mettant votre repository en Private** ;
- Step 5 : ne pas créer de nouveau projet comme dans le tutoriel mais utiliser le projet créé précédemment.

Puis dans votre projet sous github, aller dans Settings->Manage Access et "Invite a collaborator" pour ajouter l'enseignante (username `laetitiamatignon`).

2.3 Test d'une modification

Modifier le fichier `RAPPORT.md` (compte-rendu) en y ajoutant votre nom et prénom. Envoyer cette modification sur github (clic-droit sur votre projet, team, commit ; puis déplacer le fichier `RAPPORT.md` qui apparaît dans "Unstaged Changes" vers "Staged Changes", puis cliquez sur "Commit and push").

La modification devrait maintenant être reportée et visible dans votre projet sur Github.

3 Mise en place sous IntelliJ

3.1 Configuration du projet local sous IntelliJ

1. télécharger le projet sur <https://github.com/laetitiamatignon/tp-ia-mdp> : bouton vert "clone or download", puis download ZIP.
2. copier-coller le dossier téléchargé dans le dossier de votre espace de travail IntelliJ.
3. lancer IntelliJ et import project -> choisir le dossier récupéré précédemment.
4. Cocher "Create project from existing sources"
5. Ne pas modifier le nom du projet et chemin
6. Bien sélectionner les deux chemins `/simu` et `/src` proposés comme fichiers sources,
7. Cocher IAMDPLib1.8.0 comme librairie
8. Sélectionner votre SDK (par exemple 1.8 si vous avez installé Java8 pour avoir JavaFX)
9. Pour tester, faites un clic-droit sur le fichier qui est dans `simu/simuProjetPacman/PacmanGame.java`, Run 'PacmanGame.main()".

Si besoin (le test ne marche pas ...), les éléments de configuration du projet se trouvent dans Clic droit sur votre projet -> Open Module Settings :

- Dans l'onglet à gauche "Projet", vous devez avoir le bon SDK choisi (si c'est 1.8, choisir le "project Langage Level" à "8 : Lambdas, annotation, ..."
- Dans l'onglet à gauche "Module" et onglet au centre "Sources", vous pouvez spécifier (clic droit dessus) que les dossiers `simu` et `src` doivent être marqué comme Sources
- Dans l'onglet à gauche "Module" et onglet au centre "Dependencies", vous pouvez ajoutez (clic sur le plus en bas, addJARs) la librairie `IAMDPLib1.8.0.jar` (qui est dans le dossier `lib`)
- Dans l'onglet à gauche "SDKs", vous pouvez changer le SDK utilisé

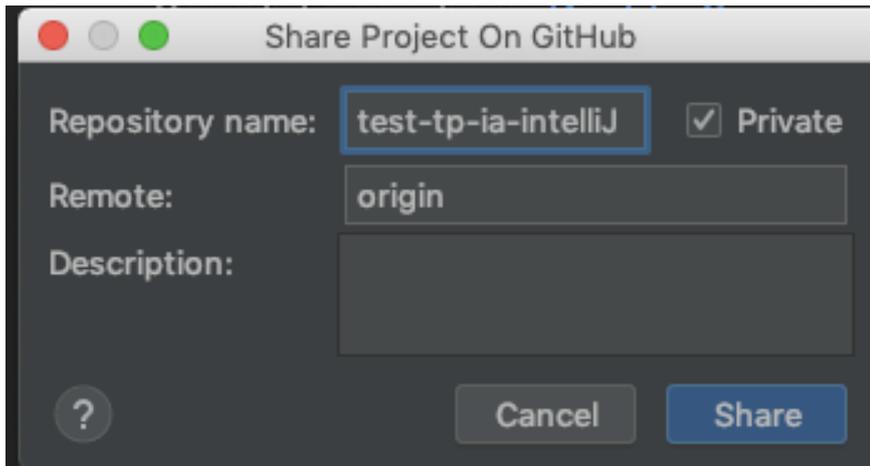
Votre projet fonctionne maintenant localement. La deuxième étape est d'ajouter ce projet dans GitHub pour permettre à l'enseignante de suivre votre travail et récupérer votre TP à la fin des séances.

3.2 Dépôt de votre projet dans GitHub (en PRIVE)

Suivre le tutoriel

<https://www.jetbrains.com/help/idea/manage-projects-hosted-on-github.html>
section "Share a project on GitHub."

- Bien préciser lorsque vous partagez le projet que vous voulez le partager en Private



- Ajouter tous les fichiers au commit initial

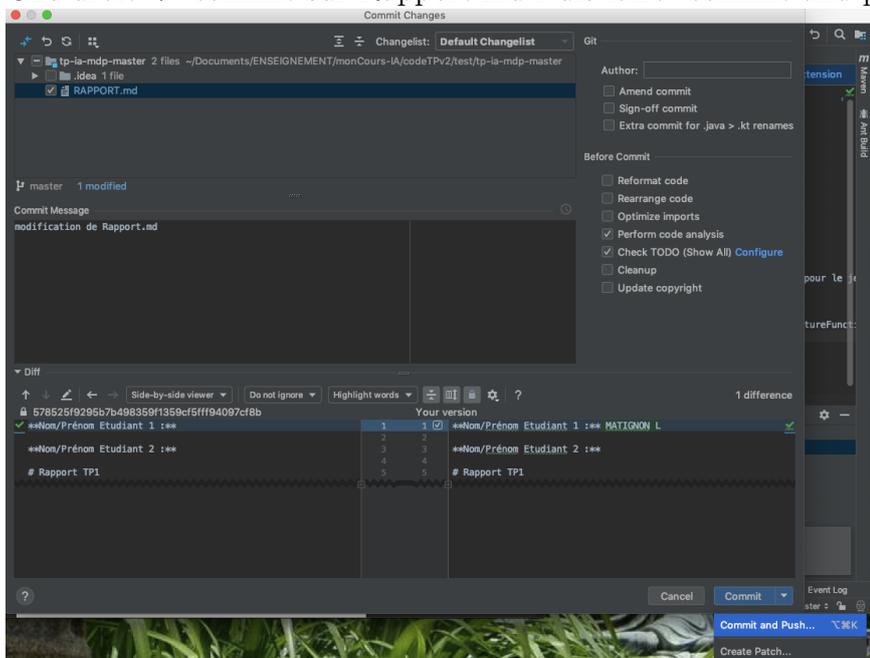
3.3 Test d'une modification

Modifier le fichier RAPPORT.md (compte-rendu) en y ajoutant votre nom et prénom.
Envoyer cette modification sur github :

- View -> Tool Window -> Version Control
- On voit alors les fichiers modifiés.



- Clic droit -> commit sur Rapport.md Puis faire "commit and push" et Push.



La modification devrait maintenant être reportée dans votre projet sur Github.

Vous avez un tutoriel sur l'utilisation de Git dans IntelliJ ici :
<https://www.jetbrains.com/help/idea/commit-and-push-changes.html>