

Manuel utilisateur deuxième itération

Leo Flamencourt, Téo Pisenti, Frederic Camail, Cylia Yangour, Alexis Courboin, Emile Devos

Avril 2023

Compilation

Compilation avec Maven

Le git Hudson ne permet pas l'envoi de fichiers de plus de 1mo sur le serveur. Il nous est donc impossible de fournir un .jar qui contient toutes les dépendances nécessaires au lancement du programme. Il faudra donc que vous recompiliez vous même le programme. Vous pouvez compiler le programme à l'aide de Maven, en faisant `mvn assembly:assembly` depuis la racine du git. Cela donnera un .jar avec toutes les dépendances dans le dossier target.

Importation dans Eclipse

Sinon vous pouvez importer le projet dans Eclipse en faisant “file > import project > Maven > import existing maven project” et en choisissant le dossier cloné avec le git.

Lancement du jeu

Le lancement du jeu est un peu particulier car le jeu dépend de LWJGL qui lui-même dépend de bibliothèques partagées natives à votre système d'exploitation. Si vous êtes sur windows ou MacOS, **vous aurez besoin de récupérer les “.dll” (windows) ou les “.dylib” (macos) dans le dossier nommé “lanceur” à la racine du git et les mettre dans le dossier courant.**

Si vous avez créé le .jar avec Maven

Si vous avez créé le .jar avec maven comme indiqué précédemment, vous pouvez lancer le projet comme suit : !! ATTENTION !! Veuillez lancer ces commandes dans la racine du git

```
java -cp target/projet-long-1.0-SNAPSHOT-jar-with-dependencies.jar  
Jouer pseudo1 pseudo2 pseudo3 pseudo4
```

Note pour MacOS

Je n'ai pas eu l'occasion de tester le logiciel sur MacOS, mais d'après la documentation de LWJGL, il est obligatoire de passer l'argument “-XstartOnFirstThread” à la jvm **sur Mac** pour pouvoir lancer le jeu. La commande devient donc :

```
java -XstartOnFirstThread -cp target/projet-long-1.0-SNAPSHOT-jar-with-dependencies.jar  
Jouer pseudo1 pseudo2 pseudo3 pseudo4
```

Distribuer le logiciel

Vous pouvez mettre le .jar “projet-long-1.0-SNAPSHOT-jar-with-dependencies.jar” dans le dossier target, généré avec maven dans le dossier “lanceur” sous le nom “monopoly.jar”. Cela vous permettra de lancer le jeu directement avec le script “run.sh” (sous Linux) ou “run.bat” (sous Windows). Le lanceur a été testé avec succès sur Linux, et Windows 10. Sur MacOS vous aurez probablement à modifier le script en suivant la section précédente “Note pour MacOS”.

Pour pouvoir utiliser le “lanceur”, il faudra que l'utilisateur aie installé java sur sa machine (accès à la commande java dans le terminal). Le script demandera à l'utilisateur de renseigner les noms des 4 joueurs.

Si vous avez importé le projet dans Eclipse

Vous devriez pouvoir lancer le jeu sans problème sur Linux en faisant : Clic droit sur la classe “Jouer” du “default package” > Run > Edit Run Configuration > Java Application

Dans le menu qui suit, modifier les arguments de la ligne de commande et mettant le pseudo des 4 joueurs (séparés par un espace).

Cela est suffisant pour lancer sur Linux.

Si vous êtes sur MacOS (non testé), essayez de rajouter “-XstartOnFirstThread” dans la section “JVM Arguments” (voir “Note pour MacOS”)

Sur Windows, faites la même chose, mais rajoutez les “.dll” du dossier “lanceur” à la racine du projet pour que le programme puisse trouver les bibliothèques partagées à son lancement. (Sur MacOS vous aurez peut-être à faire la même chose avec les “.dylib”)

Déroulement d'une partie

- Une partie se déroule jusqu'à ce qu'un joueur n'ait plus d'argent. Le classement définitif est alors donné dans le scoreboard en haut à droite de l'écran, le vainqueur est le joueur qui a le plus d'argent
- Le joueur dont c'est actuellement le tour est affiché dans le scoreboard avec une couleur différente
- Un tour se déroule comme suit :
 - Le joueur clique sur le dé
 - Le pion avance automatiquement sur le plateau
 - Une popup s'ouvre demandant une réponse au joueur
 - Le joueur peut décider de gérer les propriétés qu'il possède via le menu propriétés (non implémenté)

- Quand il a fini, le joueur clique sur “fin du tour” pour passer au tour suivant

Informations supplémentaires

- J’ai testé actuellement le jeu avec succès sur Linux et Windows 10 en générant un .jar avec maven et en distribuant le jeu via le dossier “lanceur”.
- Le jeu a été testé sur les ordinateurs de l’école avec succès en compilant et lançant avec Eclipse
- Certains vieux ordinateurs avec de vieux OS peuvent refuser de lancer le jeu car ils ne supportent pas Opengl 3.3 (vu sur des vieux ordinateurs de plus de 10 ans et sur de très rares ordinateurs du bâtiment C)
- Ce document est amené à être modifié dans le futur si je parviens à mettre la main sur un Mac
- Je prévois (Téo) de publier ce projet sur mon github en public pour en continuer l’amélioration, aussi ce document pourra être modifié d’ici la fin de semaine si de nouveaux problèmes sont détectés, mais les futures versions du code ne seront pas publiées sur hudson pour ne pas interférer avec la notation

Pour tout problème supplémentaire, ne pas hésiter à contacter Téo sur son adresse mail “teo.pisenti@free.fr”