



DOCUMENTATION DE PROJET 'BREAKOUT'

Stéphane Bouyiatiotis
Jérémy Corbaz
Joshua Gomes Da Costa
Nicodème Stalder
Lionel Widmer

Table des matières

But et motivations.....	2
Utilisation de l'application	3
Téléchargement et lancement.....	3
Comment utiliser l'application.....	3
Développement	4
Préparation de l'environnement IntelliJ	4
Documentation JavaDoc	5
Schéma UML de l'application	6
Ressources	7

But et motivations

Dans le cadre du module MCR (modèles de conception réutilisables – design patterns) à la HEIG-VD, nous avons dû étudier un modèle, et créer une application 'pas chiant' qui met en œuvre ce modèle. Notre groupe a décidé d'étudier le modèle Décorateur, et nous avons choisi de créer un jeu de casse-briques. L'avantage du casse-briques est que le modèle décorateur peut être utilisé à plein d'endroits (au niveau des briques, de la barre, etc.

Utilisation de l'application

Téléchargement et lancement

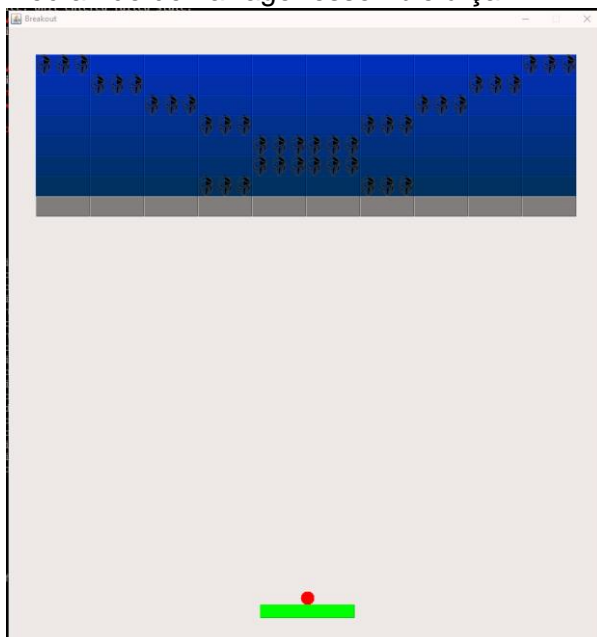
Pour utiliser l'application, il suffit d'avoir une JVM Java (version minimum 1.8). Le fichier peut être téléchargé depuis le repo GitHub et elle est également remise dans le rendu final du projet (fichier Breakout.jar).

Sur Windows, il suffit de double-cliquer sur l'application pour la lancer. Sur Linux, il faut appeler Java en lui passant le fichier jar en paramètre :

```
java -jar <PATH_TO_JAR_FILE>
```

Comment utiliser l'application

L'écran de démarrage ressemble à ça



Pour commencer à jouer, appuyer sur la touche ESPACE du clavier. Ensuite, déplacer la barre à gauche et à droite avec les touches gauche/droite du clavier.

Si la balle tombe à côté de la barre, la partie est perdue !

Développement

Préparation de l'environnement IntelliJ

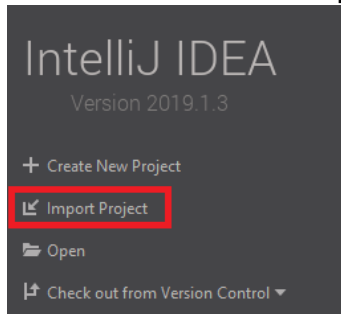
Une fois le repo GitHub (lien dans le chapitre Ressources de ce document) cloné, ou récupéré depuis les sources du projet, il faut lancer IntelliJ et importer le projet, comme ci-dessous :

- Si pas encore fait, cloner le repo :

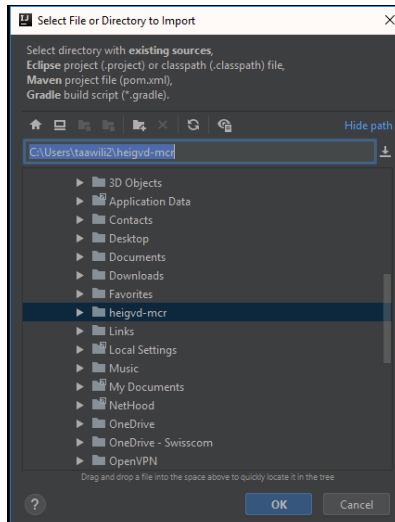
```
$ git clone https://github.com/lionelwidmer/heigvd-mcr.git
Cloning into 'heigvd-mcr'...
remote: Enumerating objects: 811, done.
remote: Counting objects: 100% (811/811), done.
remote: Compressing objects: 100% (351/351), done.
remote: Total 811 (delta 357), reused 671 (delta 227), pack-reused 0
Receiving objects: 100% (811/811), 109.09 KiB | 749.00 KiB/s, done.
Resolving deltas: 100% (357/357), done.

TAAWILI2@U309449 MINGW64 ~
```

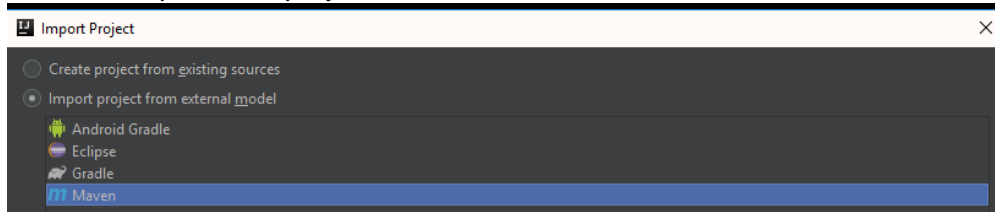
- Lancer IntelliJ et faire "Import Project"



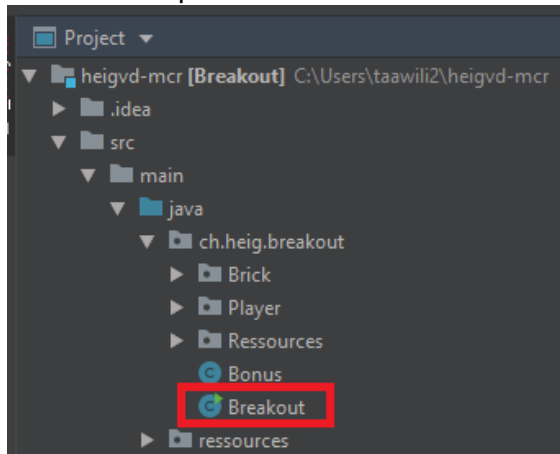
- Sélectionner le dossier du clone Git (par défaut nommé 'heigvd-mcr')



- Choisir d'importer un projet Maven



- Aux écrans suivants, tout laisser par défaut
- Une fois le projet chargé, naviguer jusqu'à la classe Breakout, faire un clic-droit dessus et cliquer sur 'Run'



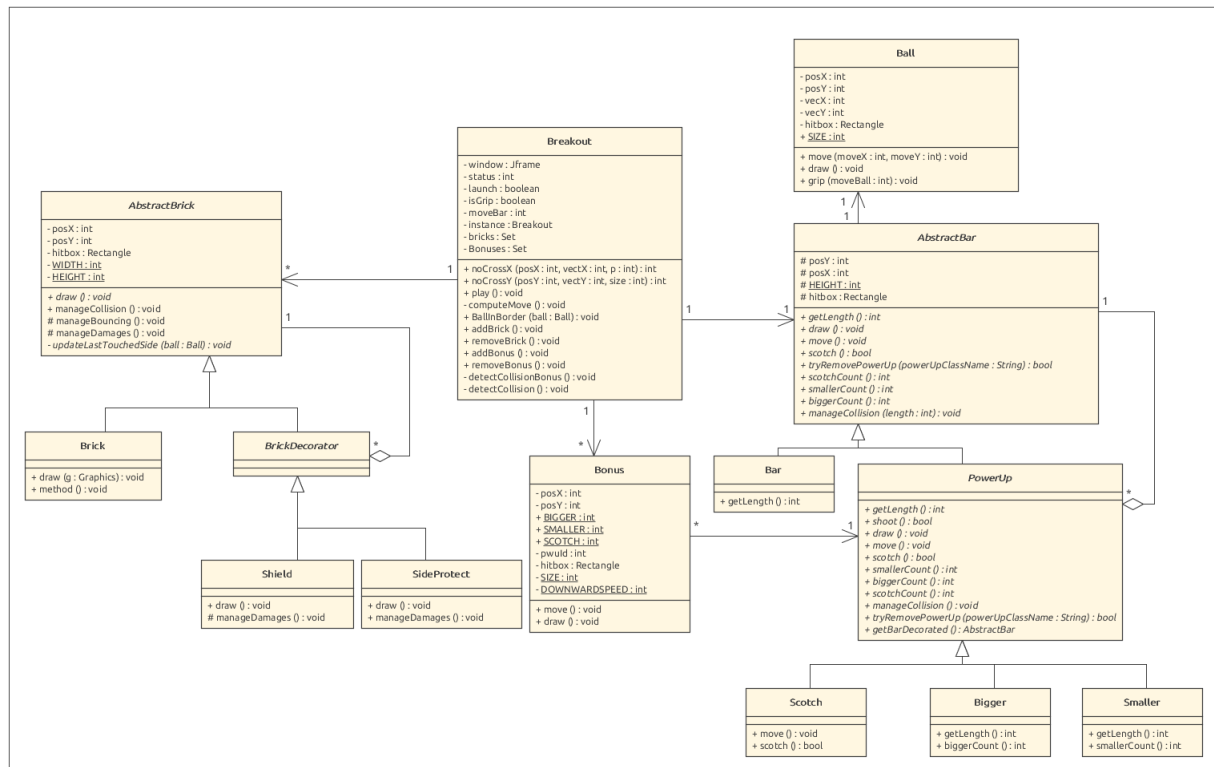
- L'application démarre et fonctionne 😊

Documentation JavaDoc

La documentation JavaDoc se trouve dans le repo GitHub, sous-répertoire docs. Cette documentation est également disponible en ligne. Le lien est disponible en bas de ce document.

Schéma UML de l'application

Le schéma ci-dessous se trouve également dans le repo GitHub, en version image et .sly (s'ouvre avec Slyum)



Ressources

Toutes les ressources nécessaires pour ce projet sont décrites ci-dessous

Quoi	Lien
Repo GitHub	https://github.com/lionelwidmer/heigvd-mcr
Documentation JavaDoc	https://lionelwidmer.github.io/heigvd-mcr/
JAR exécutable du jeu	https://github.com/lionelwidmer/heigvd-mcr/blob/master/Breakout.jar