

Explication sur la structure

CASSE - BRIQUE

docs :
dossier pour stocker
les ressources

display :
dossier contenant
l'engine, contient
tout ce qui est
commun à tous
les jeux

view :
dossier qui endosse
le rôle de vue si on
était dans un MVC

engine :
contient les fichiers
qui serviront de base
graphique

rules :
contient Graphical Object
qui gère les modalités
de création des objets

games :
contient les informations
du jeu et logiquement
le main.java

breakout :
contient les données
du jeu

entités
contient les entités
du jeu

rules :
contient entité.java
qui gère les modalités
de la création des
entités

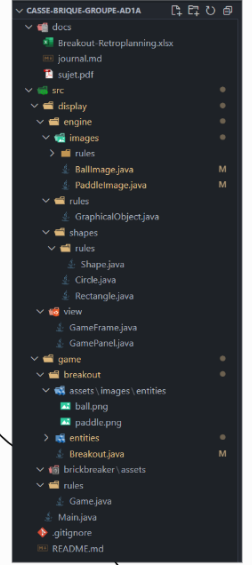
assets :
images, sons, ...

images
Permet d'avoir des
Graphical Object avec
une image

rules
contient image.java
qui s'occupe de la
manipulation des
images

shape
contient des formes
prédéfinies

rules
contient shape.java
qui gère les modalités
de création d'objets avec
une forme prédéfinies



Résumé :

La structure du projet n'est pas un MVC,

La structure est séparée en 2 dossiers, Display (qui contient l'engine) et Game (qui contient le jeu et stock les données).

On fera un rapprochement avec le MVC tout au long du projet pour des raisons de compréhension, mais la structure n'est pas un MVC.

Fonctionnement général :

Chaque sous/package contient des fichiers et un dossier rules contenant lui même un fichier qui contient les règles de créations, d'interactions etc, d'où son nom.

Les fichiers seront en général des classes-filles héritant du fichier .java qui se trouve directement dans le dossier rules (cf. package engine qui est assez explicite et facile à comprendre).

Si on veut comparer la structure avec un MVC, ça donnerait :

- Display, qui contient le dossier view qui est donc la vue et le dossier engine, qui contient la logique graphique du jeu qui fait office de contrôleur.
- Game qui contient donc les données du jeux (entités...) et sera donc comparable au modèle.