

Commentaire diagramme de classe.

Ce document est un commentaire du diagramme de classe que vous pouvez retrouver dans le même répertoire.

On ne détaillera pas particulièrement le package *view* qui est pauvre en code et traitement par définition.

Tout d'abord nous voudrions souligner le package *paint* qui contient les classes permettant de dessiner sur l'écran. *PaintView* est la classe qui va gérer les interactions avec l'utilisateur, nous l'avons copiée depuis un tutoriel et n'avons changé que très peu de code. Nous l'avons commenté le plus précisément possible pour pouvoir la comprendre. Ce que l'on peut en dire c'est qu'elle stock les mouvements de l'utilisateur sur l'écran en tant que *FingerPath*, puis que nous pouvons récupérer un rendu de la *PaintView* sous forme d'image Bitmap.

Le package métier comporte la classe *Room*, elle se construit à partir des informations stockées sur Firebase. Elle représente tout simplement la session de jeu et contient la liste des joueurs. La classe *Player* n'est ni plus ni moins que la représentation d'un joueur pour notre application, à savoir un ID unique et un pseudonyme.

Il y a eu des petites incertitudes au niveau du code, on a parfois préféré créer des Rooms directement dans les vues, tandis que parfois elles étaient créées depuis une méthode de *Player*.

C'est un point faible et nous souhaiterions que la seule classe accessible à l'instanciation depuis une *view* soit la classe *Player*.

Enfin on a la classe *LoadImage*, qui sert à afficher une image à partir d'une URL. Nous nous sommes basés sur un tutoriel pour faire cette classe.

Les controllers *Firebase** sont les classes contenant des méthodes statiques qui permettent la liaison avec notre application Firebase.

Toutes les interfaces *Listener* englobent les outils qui permettent la notification d'événements au niveau des vues. C'est la meilleure solution rapport efficacité/temps de travail que nous ayons trouvé.