

Explication de l'architecture de Little Boat

Application

- main : Sert à créer le **contrôleur** et à *lancer l'animation*

Modèle

- InputReader : Inutilisé pour le moment (à supprimer ou à remplacer par un input dans l'interface), sert à demander au joueur la position de départ du point de contrôle (actuellement elle est définie à 0:0)
- Bezier : L'objet mathématique qui permet de conserver et de *mettre à jour* les **points de contrôle** de la voile avec *un peu de hasard* (il faudrait mettre le hasard dans une classe à part)
- Box : Boîte virtuelle qui délimite les **bordures** en dehors desquelles la voile ne peut pas aller. Elle se base sur la fonction *range()* de python, qui ne fonctionne qu'avec des entiers (j'avais oublié de t'en parlé mais c'est l'une des raisons qui m'ont poussé à continuer de travailler avec des entiers)
- Wind : Permet pour l'instant *d'augmenter* ou de *diminuer* la **force** du vent. Il conserve son **état** précédent ainsi que de des **compteurs** calqués sur le point de contrôle pour le x et le y. Plus tard il faudrait simplement qu'il possède un **vecteur** (classe à créer ?) qu'il pourrait moduler à l'aide de différentes *fonctions de comportement*

Vue

- Graph : Dessine le **repère orthogonal** dans lequel navigue le bateau. Il lui donne des dimensions (plutôt grandes pour camoufler l'utilisation d'entiers dans les calculs) et une couleur de fond. Pour finir, il supprime les bordures ainsi que les axes pour un effet plus immersif

Contrôleur

- Root : (j'aurais dû en retirer toute la partie « vue » et la renommer « Controller »)
 - (vue) Crée une **fenêtre** dans laquelle elle met un **canvas** qui contient lui-même un repère dans lequel on peut dessiner des figures géométriques, c'est le **graphe**
 - (modèle) Elle crée également le modèle : les **limites** de la voile, le **vent** et la **matrice de bézier**. Pour finir, sa fonction *animate* dit au modèle de se mettre à jour à chaque frame, et à la vue de suivre le modèle.