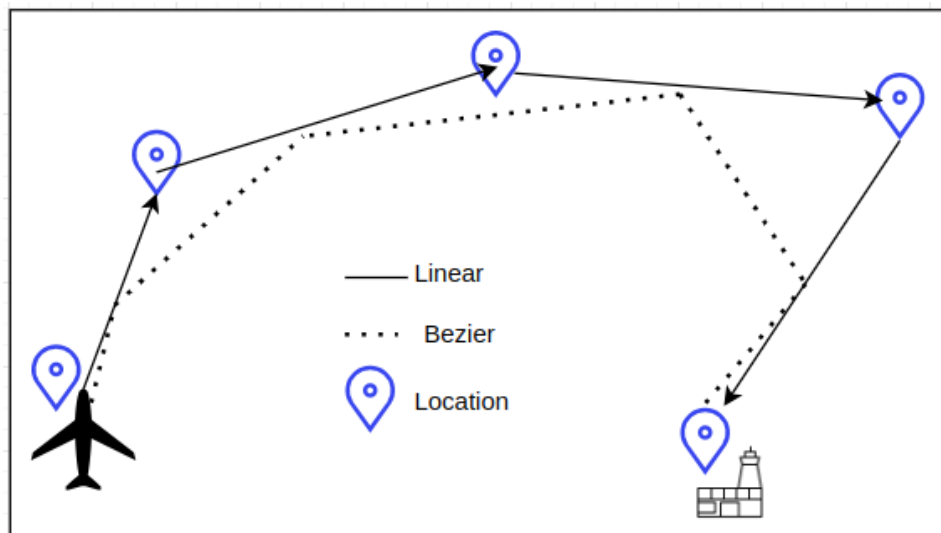
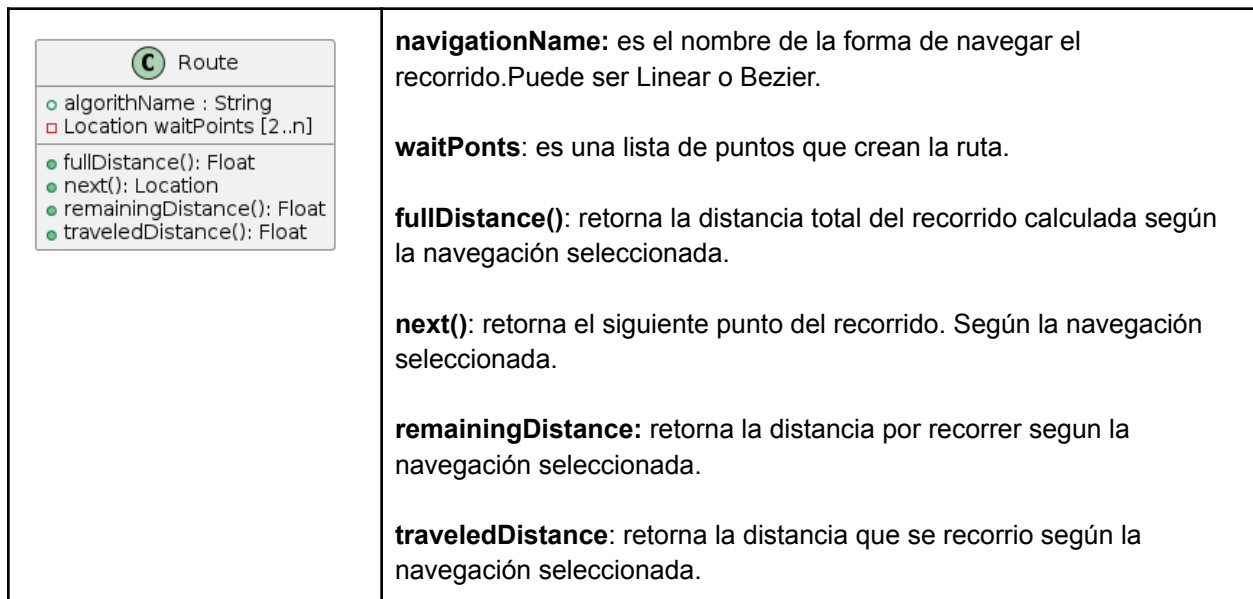


Trabajo Final: Route

Plazo de desarrollo máximo: 2 meses

Considere la clase Route que representa una ruta que esta compuesta por al menos dos coordenadas. Las coordenadas son instancias de la clase Location. En el siguiente diagrama de clases se presenta una versión simplificada de Route que implementa las operaciones:



La ruta se puede “navegar” de manera Linear o Bezier. En la manera “Linear” la ruta navegada pasa exactamente sobre los puntos que denotan la ruta (waitPoints). En la manera “Bezier” la ruta navegada se aproxima a los puntos intermedios de la siguiente manera.

Bezier (0) genera un conjunto de WaitPoints que corresponden a puntos (Locations) que están a mitad de camino entre los Waitpoints de la ruta original.

Bezier(1) genera un conjunto de WaitPoints que corresponden a puntos (Locations) que están a mitad de camino entre los waitpoints de Bezier(0).

...

Bezier(n) genera un conjunto de WaitPoints que corresponden a puntos (Locations) que están a mitad de camino entre los waitpoints de Bezier(n-1).

Bezier se configura con la cantidad de iteraciones a correr

En el futuro se van a agregar otros modos de navegación como por ejemplo navegación Dummy que solo considera el punto de partida y el de llegada.

1. Implemente test cases para Navegaciones presentadas (Linear, Bezier y Dummy)
2. Se requiere implementar implementar una solución:
 - a. Orientada a la inclusión de nuevas navegaciones dos versiones de Navegación:
 - b. Se require dos versiones de Bezier:
 - i. BezierIterativa que implementa el pseudocódigo presentado antes
 - ii. PartialBezier que es capaz de retornar el siguiente punto sin tener que calcular todos los puntos de la iteracion anterior.
3. Explique los patrones que utilizaría para solucionar el problema presentado
4. Implemente en Java o Smalltalk