PROJET INFORMATIQUE

MODULE CNAM NFP136

Année 2008

Réalisation d'un simulateur de vol aux instruments.

IFR JavaSim

GENETE Mathieu

mathieu.genete@gmail.com

Table des matières

Simulateur de vol aux instruments

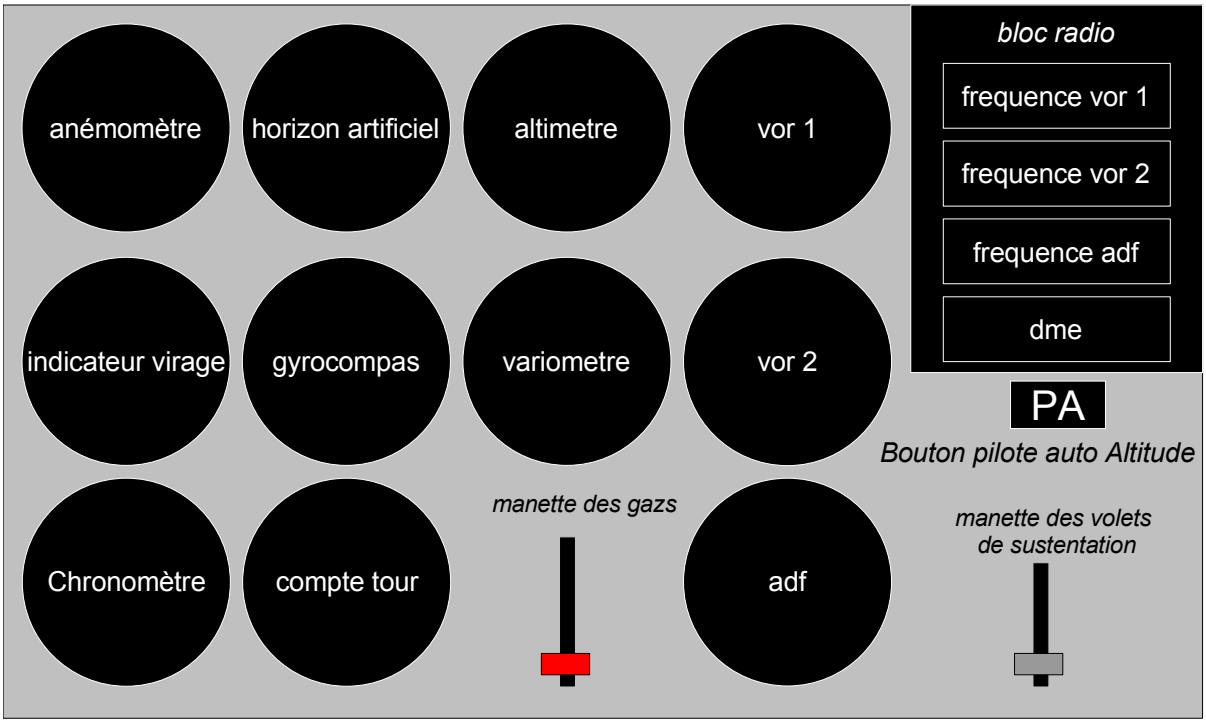
I) Description du projet:

Réalisation d'un simulateur de vol aux instruments simplifié, qui simule le fonctionnement d'un avion de tourisme avec une propulsion à hélice. L'objectif de ce logiciel est de faire découvrir le vol aux instrument aux débutants. Le modèle de vol et la motorisation sont simplifié, Il n'y a aucune infrastructure aéroportuaire, l'avion sera directement en vol ou au sol lors du démarrage du logiciel. Il est possible de réaliser décollages et atterrissages. Un fichier texte facilement modifiable dans un éditeur de texte, permettra de définir la taille de la zone de vol longueur et largeur, la position, le type et les paramètres des balises. On trouvera également dans ce fichier, les données relatives à l'avion (position, altitude,cap,vitesse,...). Ce fichier d'initialisation sera choisi au départ par l'utilisateur parmi d'autres fichiers d'initialisation présents dans le répertoire de l'application. Le projet doit être compatible Linux, Windows et Mac.

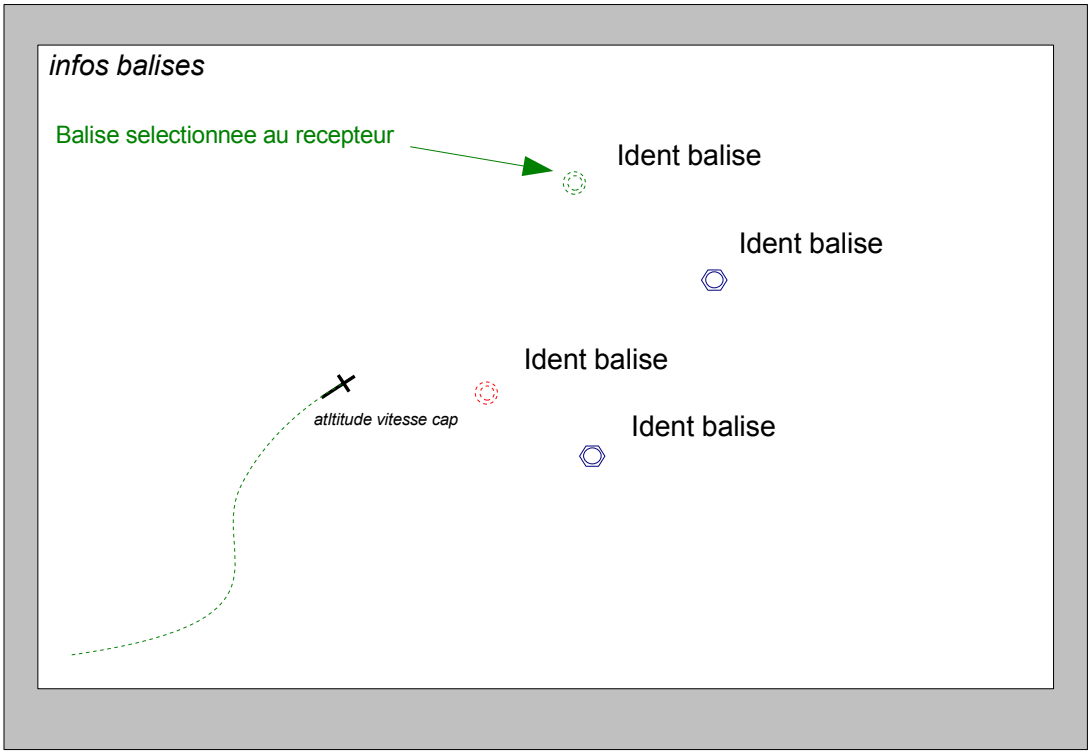
Le logiciel se lance et charge le fichier de configuration. Deux formulaire apparaissent:

- le premier contient les instruments de navigation soit: un anémomètre, horizon artificiel, altimètre, indicateur de virage, gyrocompas, indicateur de vitesse verticale, compte tour, un chronomètre. Il contient également les instruments lié à la radio navigation : deux vor, un adf ainsi qu'un bloc radio permettant d'affecter les fréquences, disposant d'un DME (mesure de distance par rapport a une balise équipée). On trouve également une manette des gazs pour la conduite du moteur; ainsi qu'une manette de contrôle des volets de sustentation.
  Enfin, un bouton permet d'activer ou non le pilote automatique pour conserver l'altitude actuelle.
  Tous ces instruments affichent leurs paramètres respectifs en temps réel.

Maquette de l'interface graphique souhaitée



- Le deuxième formulaire contient une carte graphique de la zone d'exercice. La carte affiche en temps réel l'avion, la vitesse, l'altitude et le cap de ce dernier ainsi que le trajet effectué (ligne verte). Les différentes balises avec leurs nom et fréquences. Les balises sélectionnées sur le récepteur de l'avion deviennent de couleur verte sur la carte. Il est également possible d'afficher la radiale sélectionnée sur un vor.
  La carte est interactive et permet de modifier en temps réel à l'aide de la souris, la position, l'altitude et le cap de l'avion.



Le contrôle de l'appareil s'effectue à l'aide du clavier les flèches permettent de diriger l'appareil dans le plan vertical et horizontal.

Les touches A et Q du clavier permettent respectivement d'augmenter et diminuer la puissance du moteur.

Les touches F et G respectivement rentrée et sortie des volets de sustentation.

La touche B le freins des roues du train d'atterrissage.

La touche P active ou désactive le pilote automatique pour l'altitude.

La touche R affiche ou efface les radiales des balises actives.

La touche S augmente la vitesse de simulation.

La souris peut être aussi utilisée pour paramétrer la manette des gazs, la manette des volets, le chronomètre ou certains instruments IFR (vor et adf).

Plus précisément, la position des ailerons (flèche gauche et droite) affecte le taux du virage, la position de la gouverne de profondeur (flèche haut et bas) affecte la vitesse verticale (augmentation ou diminution de l'altitude), la puissance du moteur (touche A et Q) affecte la vitesse de l'avion. La gouverne de direction n'est pas simulée.

Une description détaillée du modèle de vol est donnée ci-dessous. Certaines forces ont volontairement été omises (forces latérales, ...) pour simplifier la modélisation.

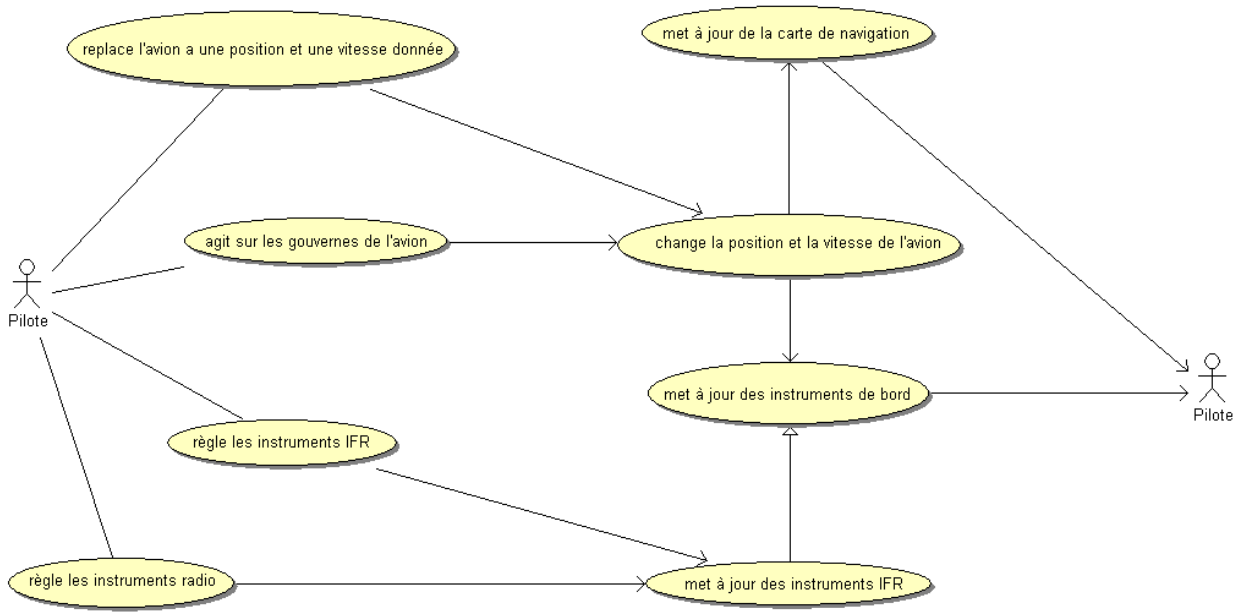
II) Environnement de développement et outils:

Le projet est entièrement réalisé en Java (JDK1.5.0) à l'aide de l'environnement de développement intégré ECLIPSE 3.2.0.

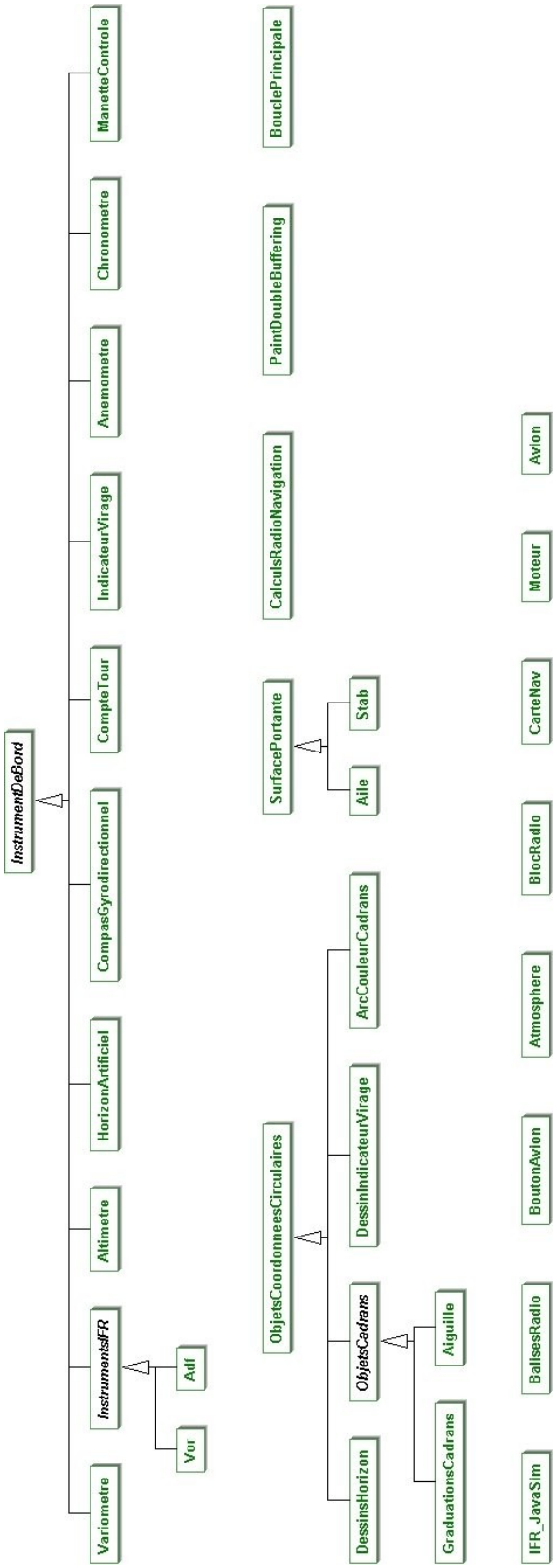
L'analyse UML est réalisée à l'aide du logiciel BOUML 4.2-1 et eUML2 Free édition 3.0.1 (plugin ECLIPSE). La documentation est réalisée à l'aide de la suite OpenOffice 2.4.

Le projet est réalisé à 50% sous le système windows XP et 50% sous la distribution linux OpenSUSE 10.2.

III) Diagramme des cas d'utilisation:



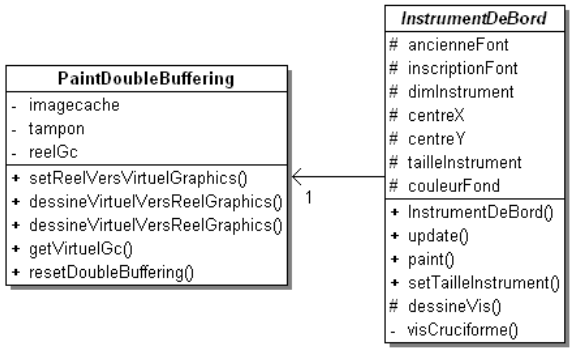
IV) Diagrammes de classe:



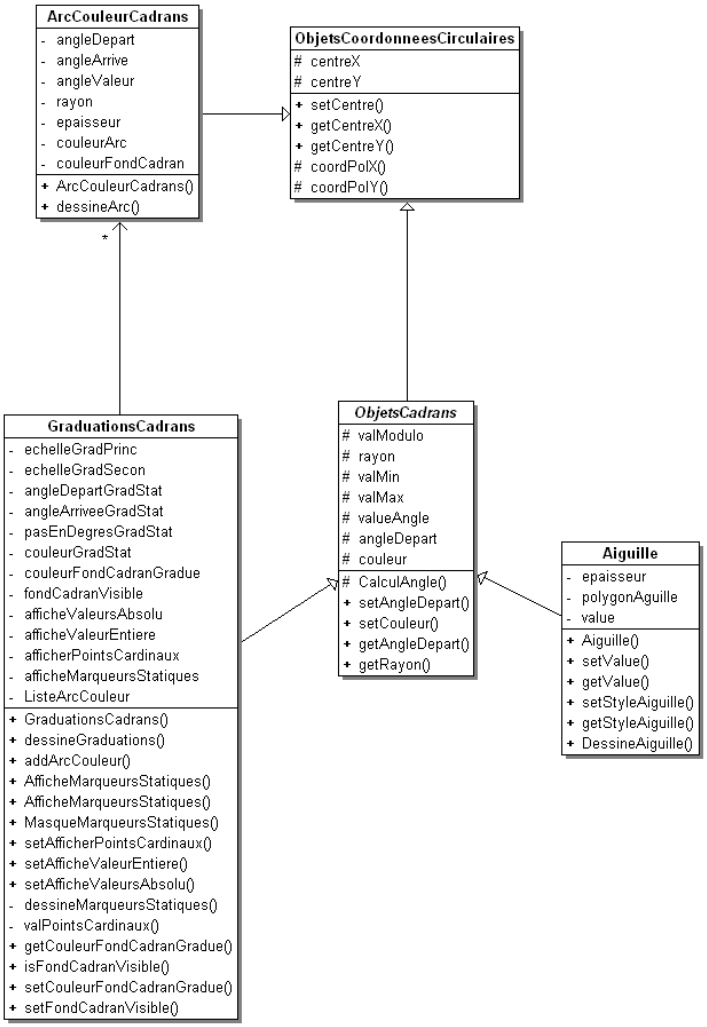
IV.1) Instruments de bord:

Schéma de la classe InstrumentDeBord, qui décrit les propriétés communes à tous les instruments.

Pour plus de détail, consulter la documentation des classes (Javadoc). Elle dérive de CANVAS.

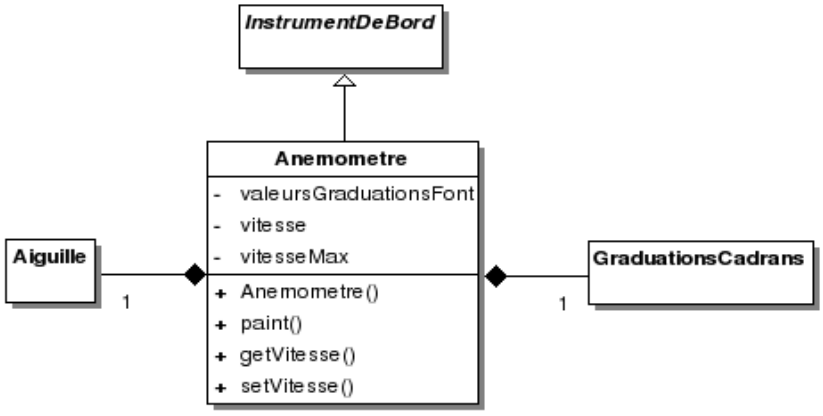


La plupart des instruments de bord contiennent des graduations des une ou des aiguilles. Il est donc nécessaire d'avoir à disposition une classe décrivant les aiguilles et les graduations.



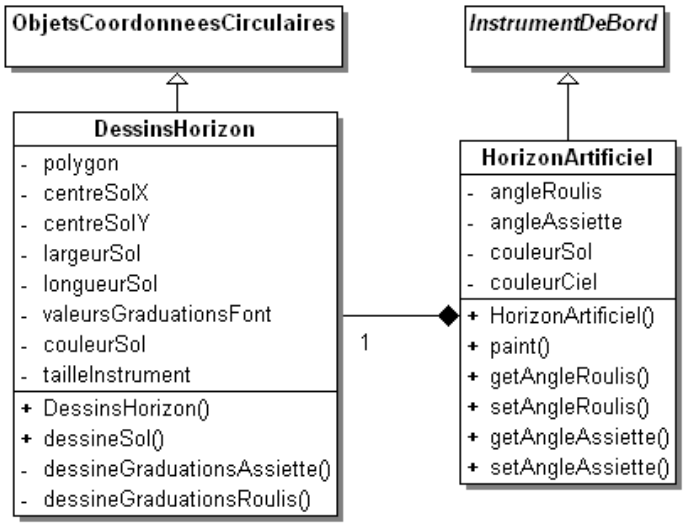
IV.1.1) Anémomètre:

L'anémomètre affiche la vitesse de l'avion en nœuds (kt). L'instrument est gradué, et dispose d'une aiguille. Il contient également plusieurs arcs colorés pour délimiter des plages de vitesse



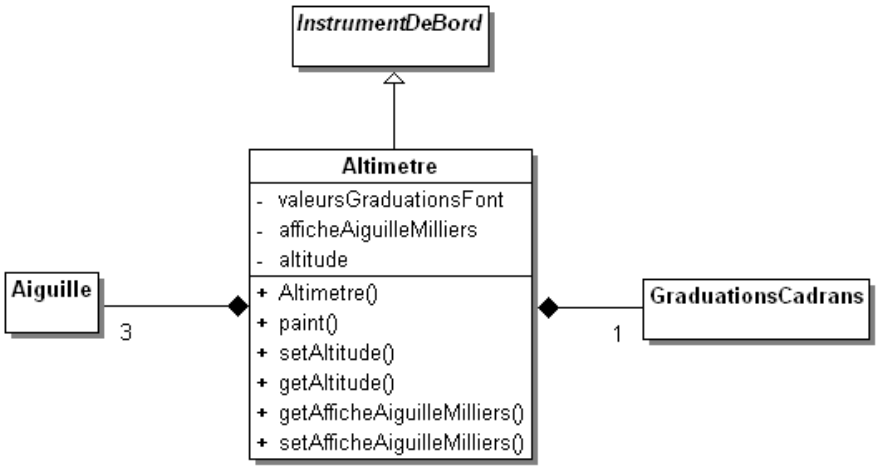
IV.1.2) Horizon artificiel:

La conception de l'horizon artificiel est différent, il ne possède pas d'aiguille, ni de cadran gradué. Cependant, il doit utiliser les coordonnées circulaire. D'où la structure suivante:



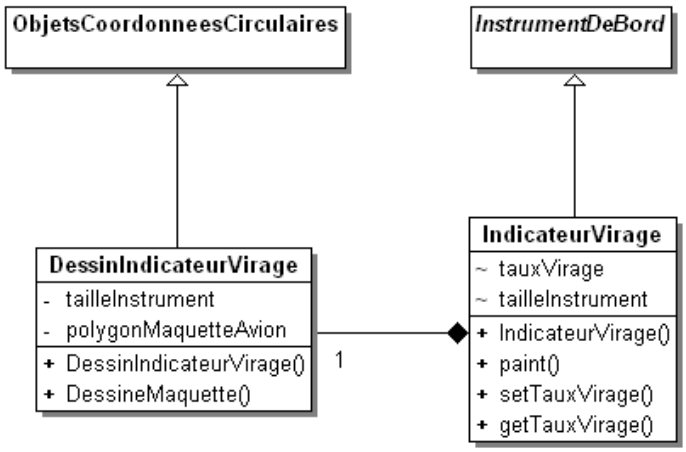
IV.1.3) Altimètre:

L'altimètre affiche l'altitude de l'avion. C'est un instrument gradué qui est composé de 3 aiguilles une pour les 100 aines de pieds, une pour les 1000 iers de pieds et une pour les 10 000 iers de pieds qui est affichable à volonté.



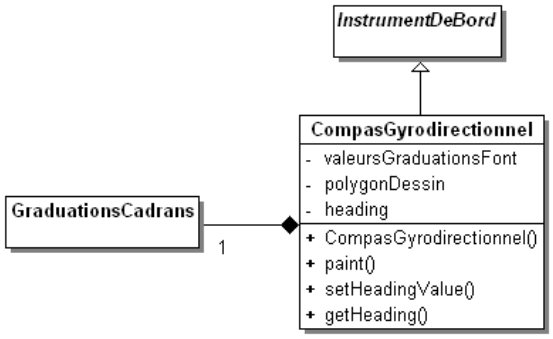
IV.1.4) Indicateur de virage:

L'indicateur de virage donne le sens et le taux du virage. Il est composé d'une maquette de l'avion, vu de l'arrière, qui peu pivoter. Lorsque les ailes de la maquette de l'instrument sont au niveau de la graduation L ou R, alors l'avion effectue un virage standard : 180°/minute



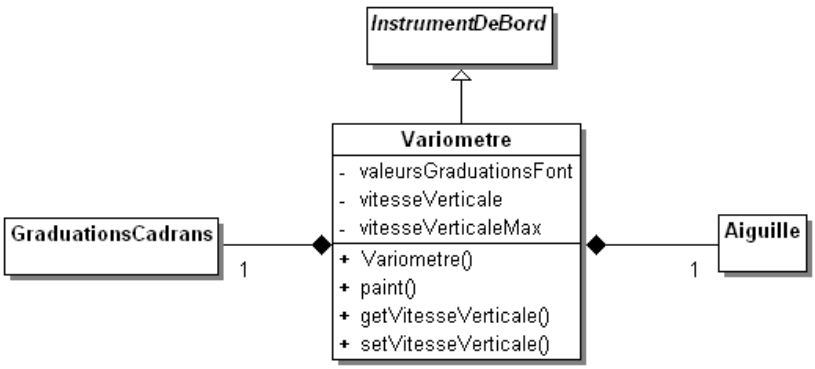
IV.1.5) Conservateur de cap:

Le conservateur de cap indique le cap de l'avion en degrés. Il est composé d'une maquette statique de l'avion au centre et d'une rose des cap graduée qui pivote.



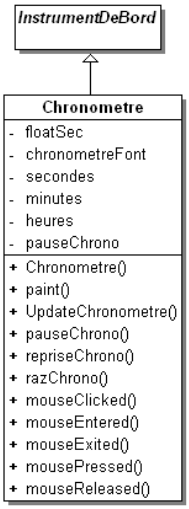
IV.1.6) Variomètre:

Le variomètre indique la vitesse verticale de l'avion en pieds/minute. Il est composé d'un cadrant gradué et d'une aiguille.



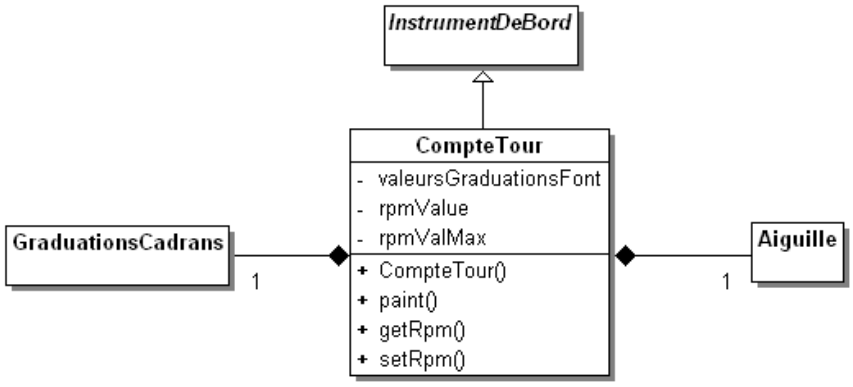
IV.1.7) Chronomètre:

Le chronomètre indique le temps écoulé depuis le dernier déclenchement. Il peut être mis en pause et redémarré (clique gauche souris), remis à zéro (clique droit souris). Il affiche le temps sous forme digitale.



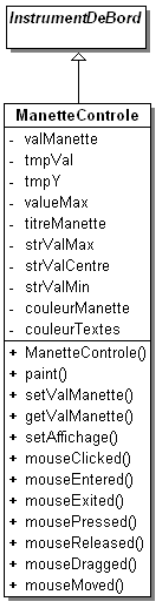
IV.1.8) Compte tours:

Le compte tour indique la vitesse de rotation de l'hélice en tours minute. C'est un instrument gradué, avec une aiguille. Il contient plusieurs arcs colorés pour délimiter des domaines d'utilisation.



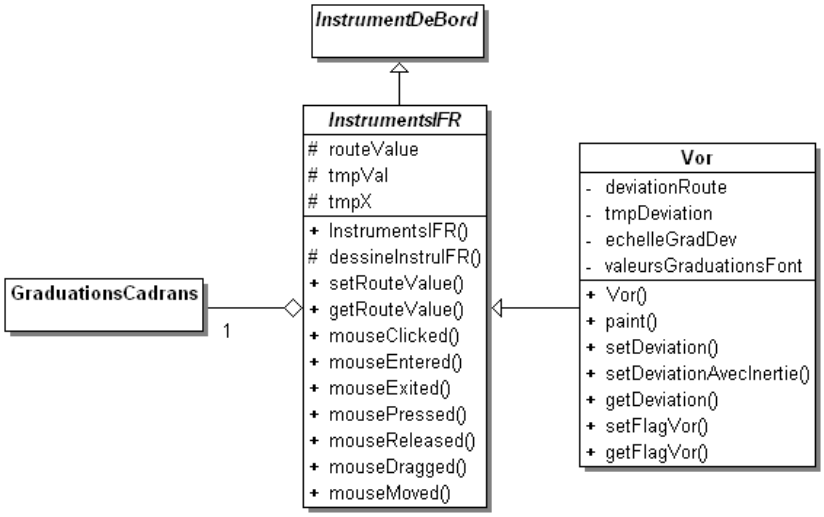
IV.1.9) Manette des gazs et manette des volets de sustentation:

Il s'agit d'un petit curseur, qui peut être déplacé à la souris. L'objet renvoie une valeur comprise entre 0 et 1000 pour le maximum.



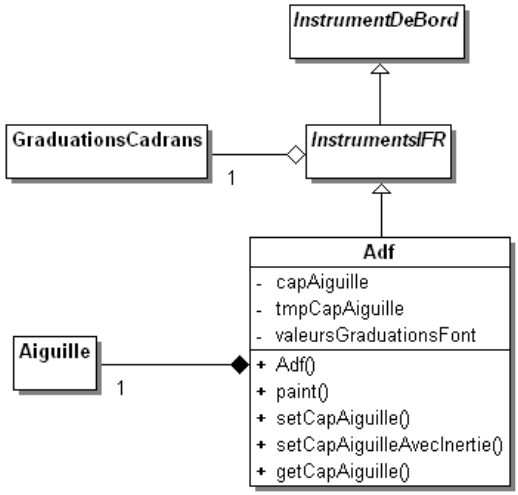
IV.1.10) VOR ou VHF Omnidirectional Range (instrument):

L'instrument VOR est un instrument IFR (instrument flight rules), composé d'une rose des caps amovible grâce à la souris par un cliquer déplacer; d'un indicateur de déviation par rapport à la route (barre verticale) et d'un indicateur de statuts (OFF, TO, FROM)



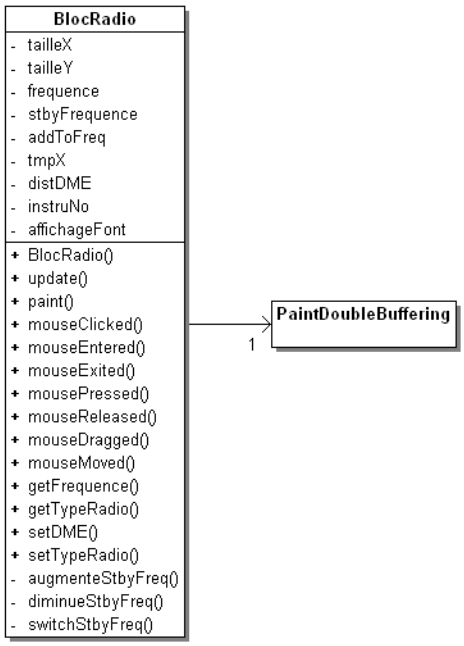
IV.1.11) ADF ou Automatic Direction Finder (instrument):

L'instrument ADF est un instrument IFR, composé d'une rose des caps amovible grâce à la souris par un cliquer déplacer et d'une aiguille donnant la direction de la balise.



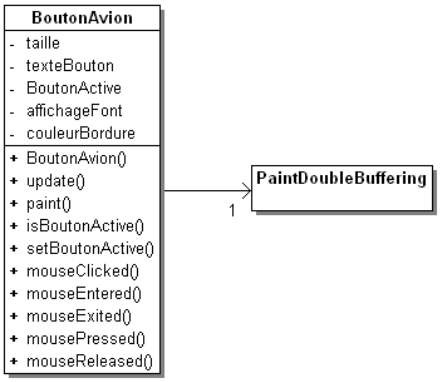
IV.2) Le bloc radio:

Le bloc radio permet de créer des récepteurs pour la radionavigation: VOR, ADF ou DME (Distance Measuring Equipment). A droite, on trouve la fréquence en stand-by, modifiable à l'aide des boutons gauche et droit de la souris. A gauche, on trouve la fréquence active. Un cliquer déplacer à l'aide de la souris permet de permuter les deux fréquences. Dans le cas du DME, il n'y a pas de fréquence stand-by.



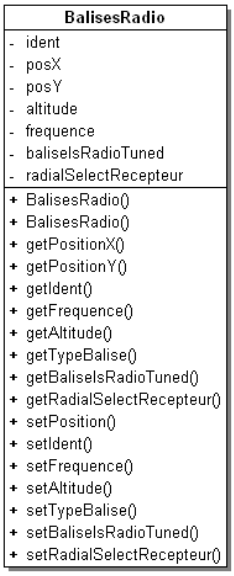
IV.3) Bouton du pilote automatique:

Il s'agit d'un bouton ON ou OFF, qui peut être personnalisé.



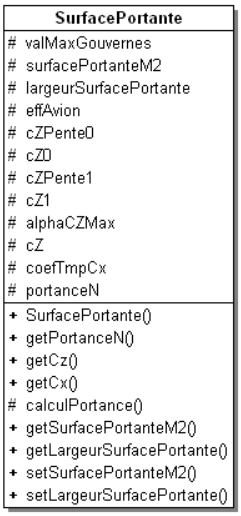
IV.4) Balise Radio:

Définit une balise radio de type VOR, VOR DME ou NDB. La balise dispose d'un identifiant, d'une fréquence, d'une position horizontale et verticale (altitude)



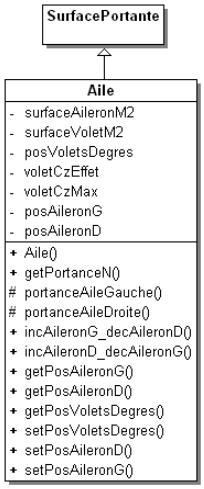
IV.5) Surface portante:

La surface portante décrit les propriétés de l'aile et du stabilisateur de l'avion. On retrouve le calcul de la portance.



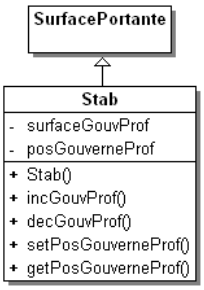
IV.5.1) Aile:

L'aile de l'avion est une surface portante, comportant 2 ailerons et des volets de sustentation.



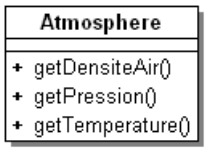
IV.5.2) Stabilisateur de profondeur:

Le stabilisateur de profondeur est également une surface portante de comportant qu'un seul aileron.



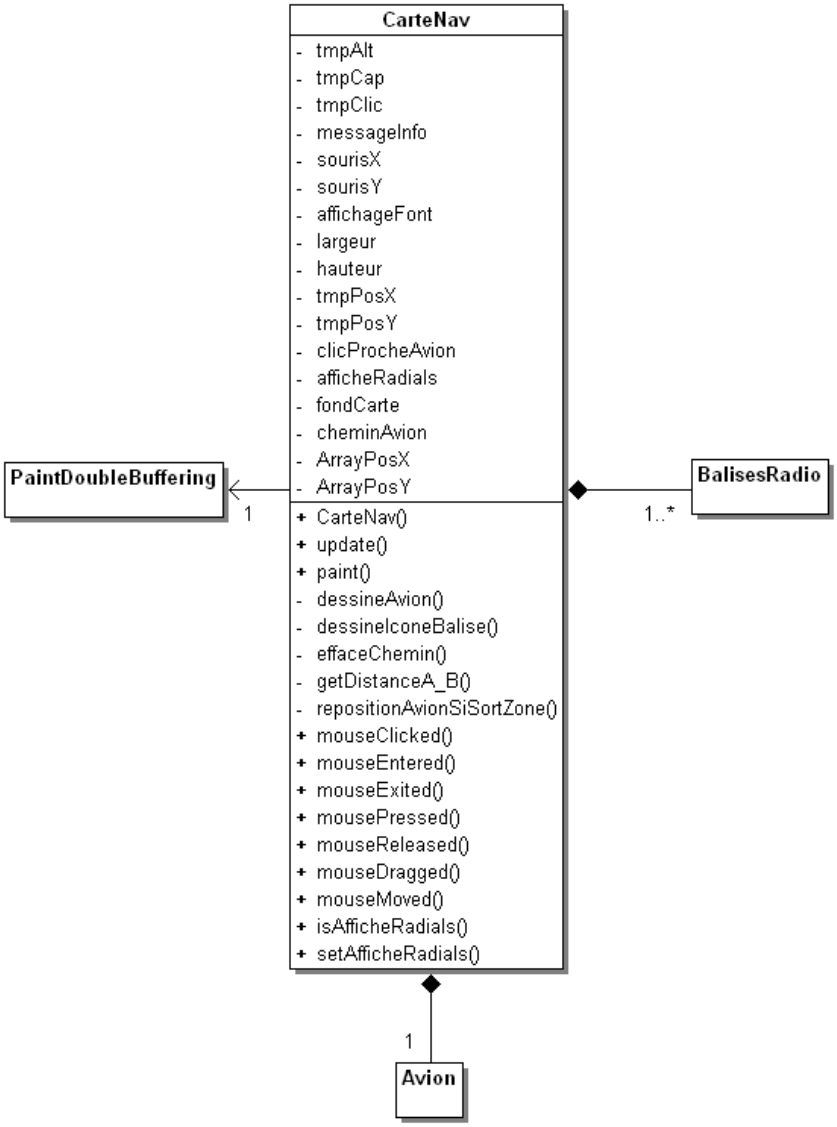
IV.6) Atmosphère:

Décrit un modèle simplifié d'atmosphère, retournant la densité, la température et la pression pour une altitude donnée.



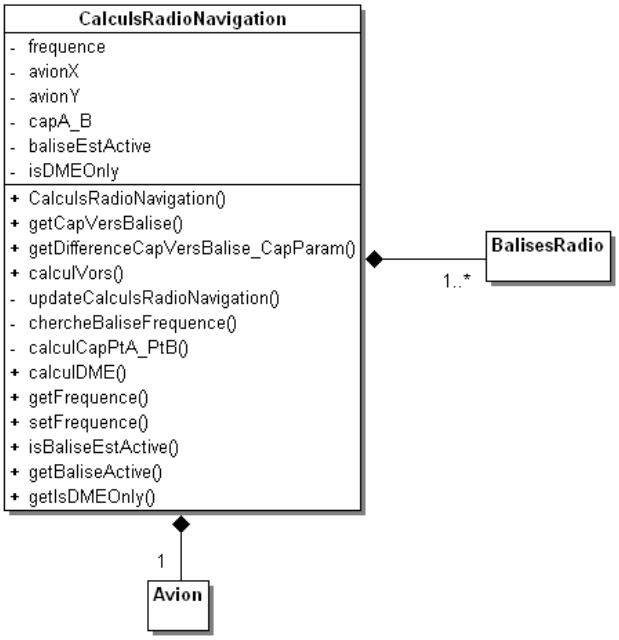
IV.7) Carte de navigation:

La carte de navigation permet de visualiser en temps réel la position de l'avion, de son trajet, des balises et des aides à la radionavigation. La carte dérive de CANVAS.



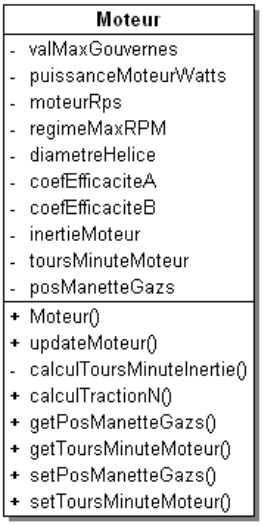
IV.8) Calculs pour la radionavigation:

Les calculs concernant les moyens de radionavigation sont regroupé dans cette classe.



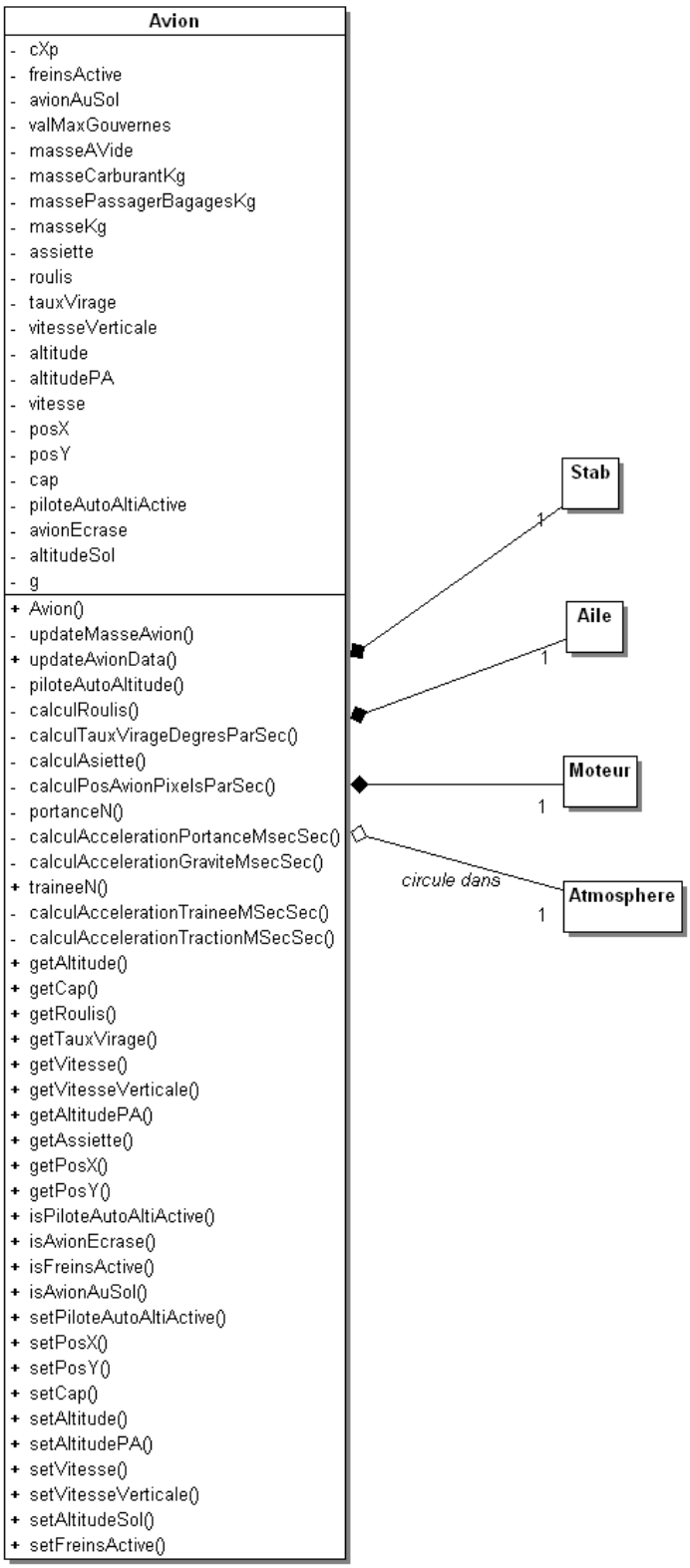
IV.9) Moteur:

La motorisation de l'avion est définit dans cette classe. On retrouve également le calcul de la traction.



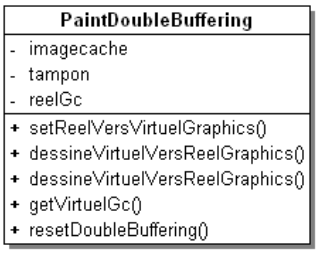
IV.10) Avion:

L'avion est définit dans la classe du même nom. On retrouve également le calcul de la trainée et du poids.



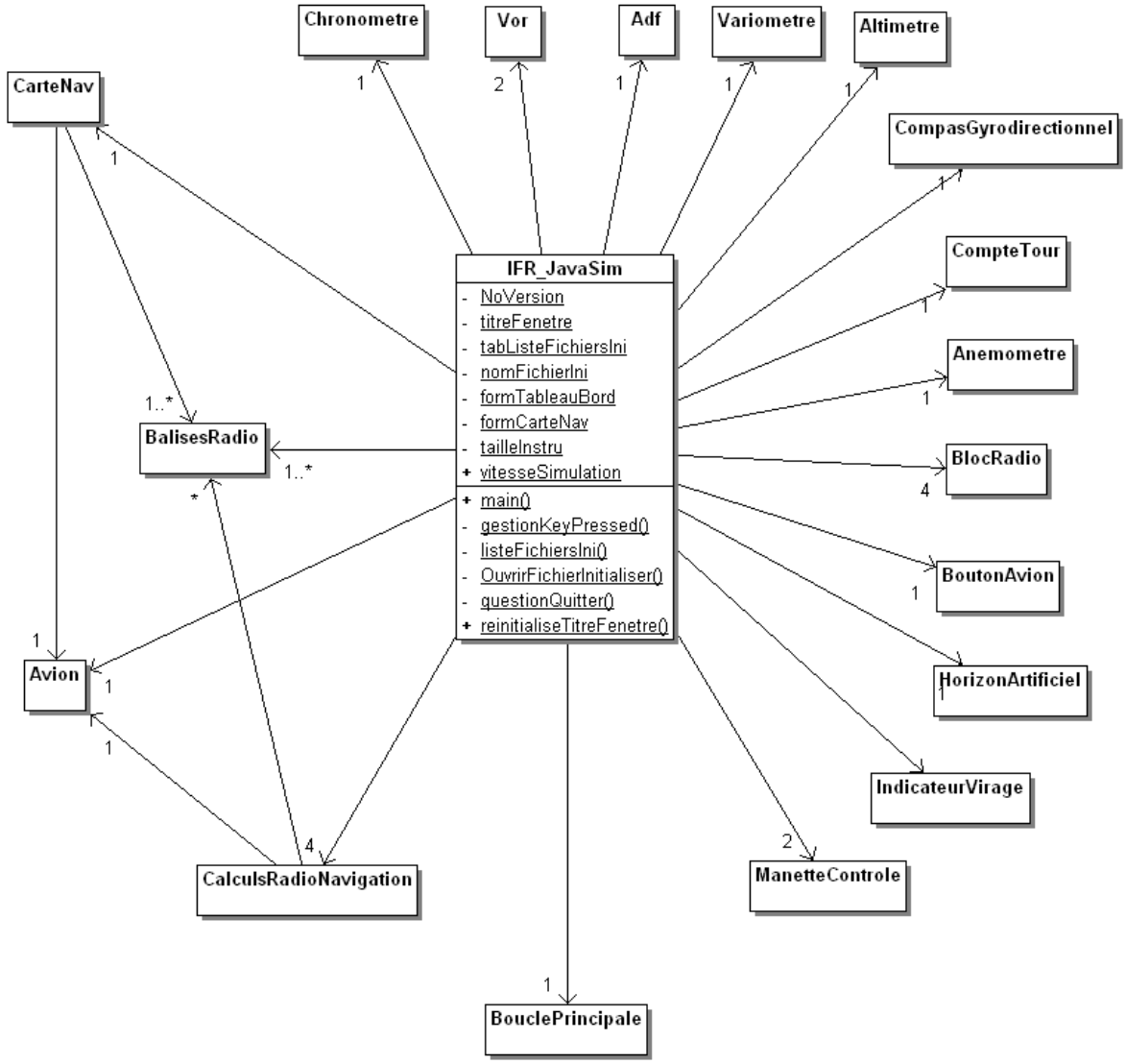
IV.11) Double bufferging pour l'affichage:

Lorsqu'on rafraichit un dessin en java, on observe un phénomène de clignotement. Tous les objets sont affichés les uns après les autres. La classe PaintDoubleBuffering permet d'éviter ce phénomène, en dessinant l'instrument dans une image « virtuelle », affichée ensuite sur le formulaire.



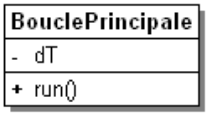
IV.12) Programme principale:

La fonction main() se trouve dans le fichier IFRJavaSim.java. Le programme charge le fichier sélectionné par l'utilisateur et initialise ses composants. La boucle principale est alors démarrée.



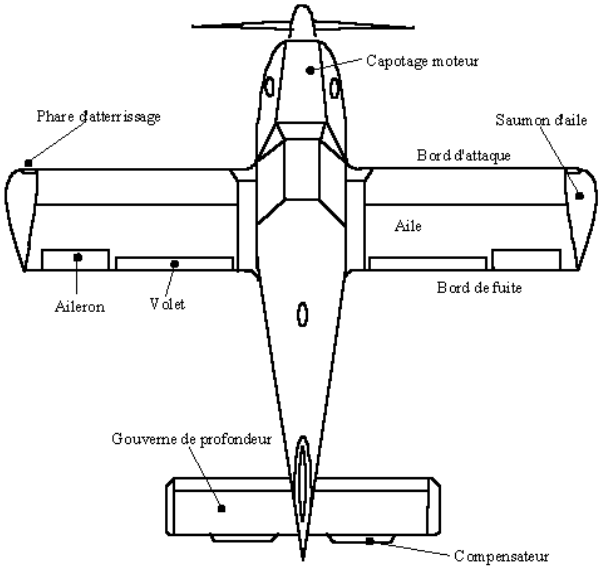
IV.13) Boucle principale:

La boucle principale dérive de la classe Thread. Elle dispose d'une fonction run() constituée d'une boucle infinie réalisée à cadence régulière de 50ms (dT). Cette boucle permet de mettre à jour tous les éléments du programme, en temps réel : avion, instruments de bord, carte de navigation.

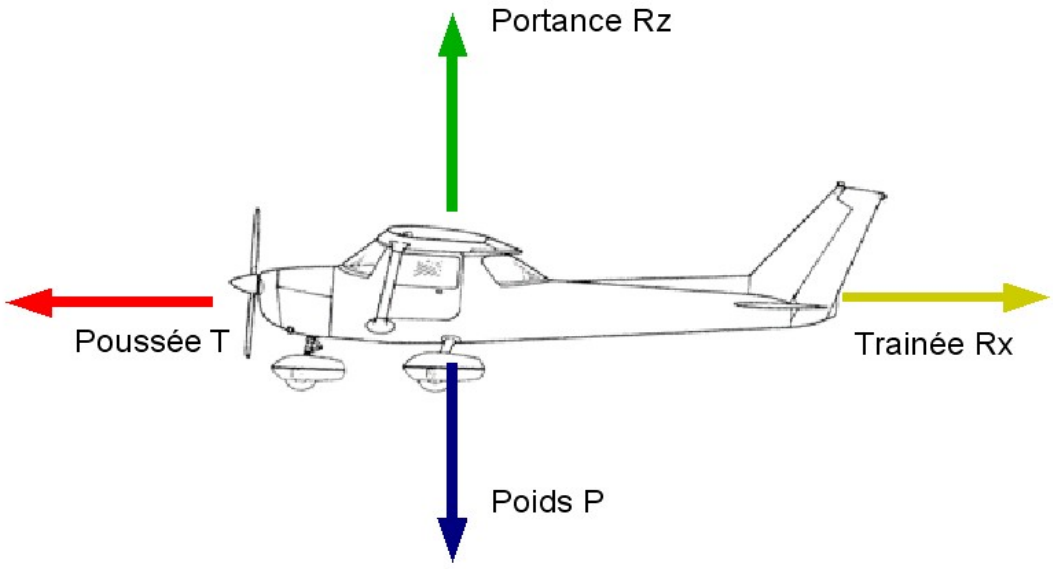


V) Modèle de vol simplifié:

Schéma d'un avion:



Schémas des forces:



Navigation sur le plan horizontal:

La position des ailerons est définit de la façon suivante: une position basse et haute, avec une multitude de positions intermédiaires. Chaque aileron a une position symétriquement opposé à l'autre.

Dans le programme, la position d'un aileron varie de -1000 à +1000

lors de l'appuie sur la flèche gauche: aileron gauche += 1 et aileron droit -=1

lors de l'appuie sur la flèche droite: aileron gauche -= 1 et aileron droit +=1

L'inclinaison de l'avion est définit de la façon suivante:

$$Inclinaison\,de\,l\,'\,avion\,en\,degrés = 180 \times \frac{(valAileronGauche - valAileronDroit)}{2000}$$

L'inclinaison ainsi calculée permet de déterminer le taux du virage en degrés par secondes. Ce taux dépends également de la vitesse de l'avion V:

$$Taux\,du\,virage\,en\,degrés / seconde = -inclinaison \times \frac{3}{(V \times 0{,}15)}$$

ATTENTION ici à la division par 0

La simplification permet de manœuvrer plus facilement l'appareil au clavier.

Navigation sur le plan vertical :

La position de la gouverne de profondeur est définie de la façon suivante : comme les ailerons, la gouverne a une position haute et basse avec une multitude de positions intermédiaires.

Dans le programme, la position de la gouverne de profondeur varie de -1000 à 1000.

Lors de l'appuie sur la flèche du haut : gouverne de profondeur -= 1

Lors de l'appuie sur la flèche du bas : gouverne de profondeur += 1

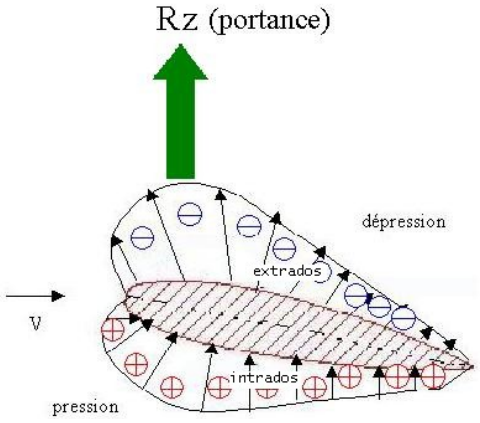
On détermine donc l'assiette en degrés de l'avion par cette formule:

$$Assiette = 90 \times \frac{valGouverneProfondeur}{1000}$$

La simplification permet de manœuvrer plus facilement l'appareil au clavier.

V.1) Calcul de la portance résultante Rz'

Les filets d'air parcourant la courbure supérieure de l'aile (extrados) créent une dépression qui aspire l'aile vers le haut. Les filets d'air parcourant la courbure inférieure de l'aile (intrados) créent un surpression qui pousse l'aile vers le haut. 70 % de la portance est fournie par la dépression de l'extrados.



$$Rz = Cz \cdot \rho \cdot S \cdot \left(\frac{V^2}{2} \right)$$

Rz, Rz' : portance en Newton

Cz: coefficient de portance dépendant du profil de l'aile. Dans notre cas, le profile est du type NACA 2412. (voir ci-dessous pour plus de détail)

ρ: masse volumique de l'aire en kg/m^3 $\rho = 3{,}407 \cdot 10^{-9} \cdot A^2 - 1{,}150 \cdot 10^{-4} \cdot A + 1{,}225$ avec A l'altitude en mètres (cf classe Atmosphere)

S: surface portante en m^2. Dans le cas du simulateur, l'aile fait 16,2 m^2

V: la vitesse de l'avion en m/s

D'après cette formule, on calcul Rz' qui dépends de l'inclinaison en roulis de l'avion et de l'assiette.

$$Rz' = Rz \cdot \cos\left(\alpha \cdot \frac{\Pi}{180} \right) \cdot \cos\left(\beta \cdot \frac{\Pi}{180} \right)$$

avec α l'inclinaison en roulis (degrés) et β l'assiette en degrés

Une fois la portance définie, on peut calculer l'accélération verticale de l'avion Az en m/s^2:

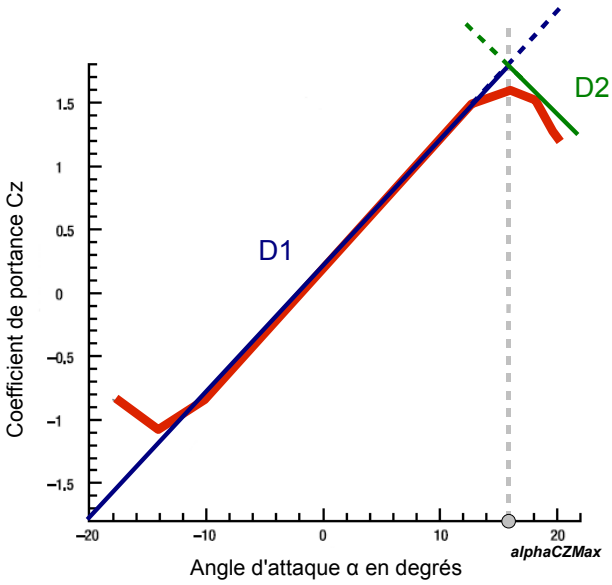
$$Az = \frac{Rz'}{m}$$

avec m : masse de l'avion en kg

(Pour obtenir l'accélération verticale en ft/min/s : $Az\,(ft/min/s) = Az \times 3{,}281 \times 60$)

Coefficient de portance Cz:

Le coefficient de portance dans le modèle vol du simulateur n'est valide que pour des assiettes entre -16 et +20 degrés. α est l'angle d'attaque de l'aile, confondu ici avec l'assiette de l'avion (en degrés).



Coefficients de portance pour le profil d'aile NACA 2412

Coefficient de portance pour le profil NACA2412 : courbe rouge.
Dans le simulateur, cette courbe est approximée à l'aide de deux droites D1 et D2 (ci-dessus).
Avec:

$$D_1 : C_z(\alpha) = cZPente0 \cdot \alpha + C_{z0}$$

$$D_2 : C_z(\alpha) = cZPente1 \cdot \alpha + C_{z1}$$

quand α < ou = à alphaCZMax on utilise D1

quand α > alphaCZMax on utilisa D2

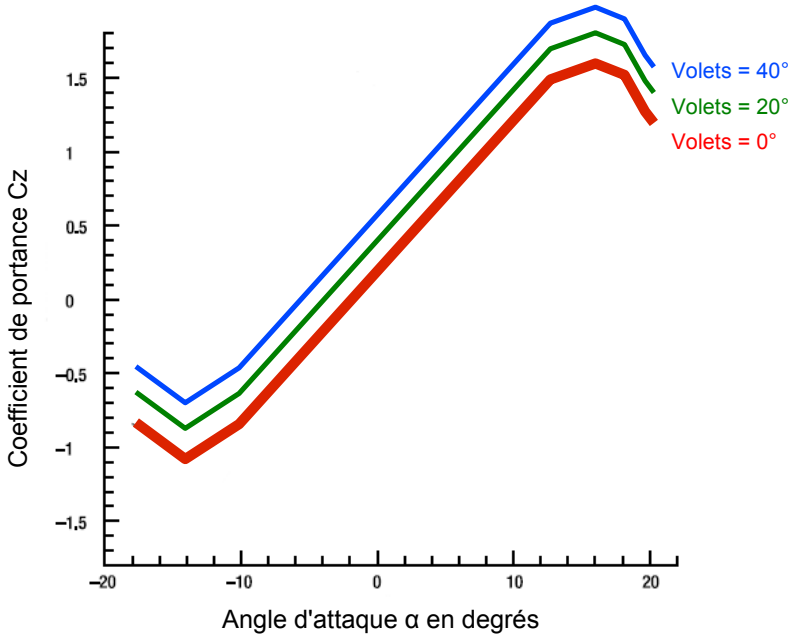
Dans le programme:
alphaCZMax = 16

cZpente0 = 0,0889
Cz0 = 0,178

cZpente1 = -0,1
cZ1 = 3,2

Effets des volets de sustentation sur le coefficient de portance Cz:

Les volets de sustentation permettent d'augmenter la portance de l'aile. Il entrainent également une augmentation du coefficient de trainée. Dans le logiciel, les volets positionné a 40° augmentent de 0,3 le coefficient de portance.



Effet des volets sur les coefficients de portance

V.2) Calcul du Poids P:

$$P = m \cdot g$$

P : poids en Newton

m: masse de l'avion en kg

g: accélération de la pesanteur = 9,81 m/s^2

L'accélération verticale, du poids Ap est opposé à Az : $Ap = \frac{P}{m} = g = 9{,}81\, m/s^2$

Ap est constant.

V.3) Calcul de la Trainée Rx:

$$Rx = Cx \cdot \rho \cdot S \cdot \left(\frac{V^2}{2} \right)$$

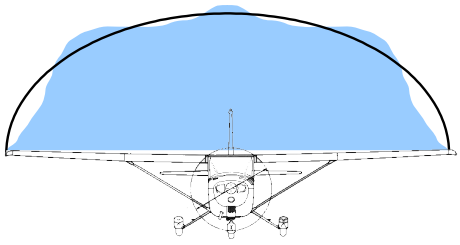
Les coefficients sont identiques à ceux de la portance, sauf Cx qui correspond au coefficient de trainée.

Calcul de l'allongement de l'aile λ : $\lambda = \frac{largeur\ de\ l'\ aile^2}{Surface_{aile}}$ largeur de l'aile en m et surface de l'aile en m^2.

Calcul de l'allongement effectif de l'aile λe :

$$\lambda_e = \lambda \cdot eff$$

avec *eff* l'efficacité de l'aile (0 = non efficace; 1 = efficacité maximum). Lorsque la répartition de la portance suit un profile elliptique (schéma ci-dessous) $\lambda_e = \lambda$ donc $eff = 1$



Le profile (en bleu) ne suit pas parfaitement l'ellipse (noir). Dans le simulateur $eff = 0{,}77$

Le calcul du coefficient Cx est le suivant : $C_x = C_{x0} + \frac{C_z^2}{\pi \cdot \lambda_e}$ avec C_{x0} le coefficient de trainée parasite.

Une fois la trainée définie, on peut calculer l'accélération de la trainée de l'avion Ax en m/s^2:

$$Ax = \frac{Rx}{m}$$

avec m : masse de l'avion en kg (dans le cas du simulateur, m = 900 kg)

V.4) Calcul de la poussée ou traction FT, conduite du moteur:

Le calcul de la traction, tient compte de l'altitude. Cet effet est définit par la formule suivante:

$$\sigma = \frac{\rho}{\rho_0}$$

avec σ le rapport entre la densité de l'air à l'altitude de l'avion ρ et la densité de l'air au niveau de la mer $\rho_0 = 1{,}225$.

On détermine à partir de σ le facteur de perte de puissance Φ :

$$\Phi = \frac{\sigma - C}{1 - C}$$

C correspond à un facteur mécanique de perte de puissance, indépendant de l'altitude C=0,12.

On détermine ensuite le paramètre de fonctionnement de l'hélice γ qui correspond au rapport de la vitesse propre de l'avion et de la vitesse périphérique en extrémité de pale.

$$\gamma = \frac{V}{N \cdot D}$$

V : vitesse propre de l'avion en m/s
N : vitesse de rotation de l'hélice en tours/seconde
D : diamètre de l'hélice en m

La puissance de traction de l'hélice est définie par:

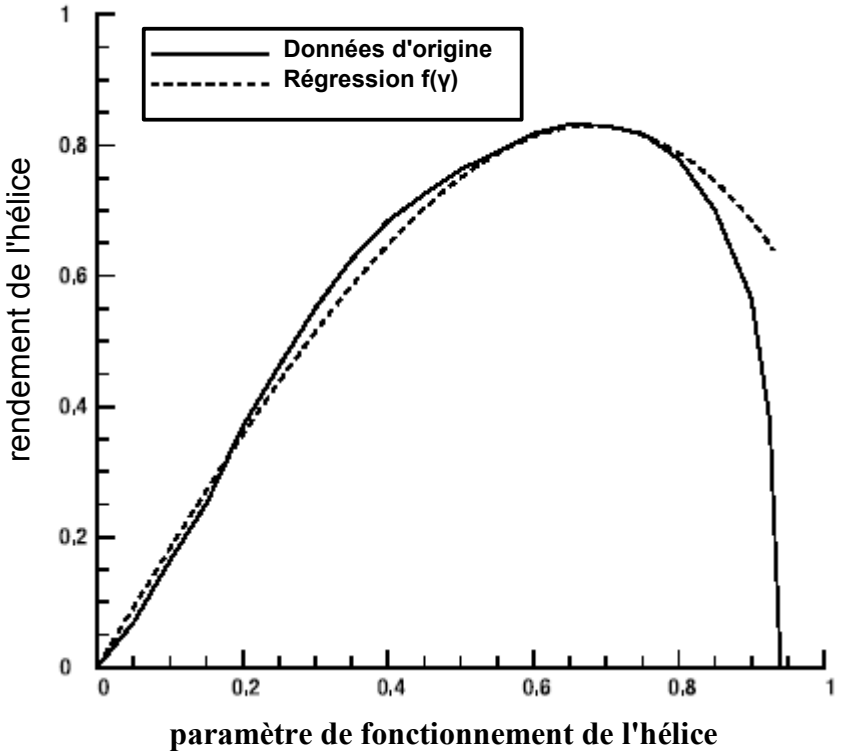
$$P_T = \eta_P \cdot P_E$$

avec P_E la puissance du moteur en watts et η_P le coefficient de rendement de l'hélice.

On a également $P_T = F_T \cdot V$

avec F_T la traction en N et V la vitesse propre de l'avion en m/s

voici ci-dessous le rendement de l'hélice en fonction de γ



On peut modéliser cette courbe à l'aide d'une équation du type $\rho_p = a \cdot \gamma + b \cdot \gamma^3$ avec a = 1,83 et b = -1,32 (courbe en pointillé sur le schéma précédent).

On a : $P_T = F_T \cdot V$ on veut calculer la traction F_T

$$F_T = \frac{P_T}{V}$$

$$P_T = \eta_P \cdot P_E \quad \text{d'où} \quad F_T = \frac{\eta_P \cdot P_E}{V}$$

$$\gamma = \frac{V}{N \cdot D} \quad \text{on en déduit} \quad V = \gamma \cdot N \cdot D$$

$$\text{d'où} \quad F_T = \frac{\eta_P \cdot P_E}{\gamma \cdot N \cdot D}$$

$$\eta_P = a \cdot \gamma + b \cdot \gamma^3$$

$$F_T = \frac{P_E}{\gamma \cdot N \cdot D} \cdot (a \cdot \gamma + b \cdot \gamma^3) = \frac{P_E}{N \cdot D} \cdot (a + b \cdot \gamma^2)$$

Pour incorporer l'effet de l'altitude, on multiplie simplement l'équation par le facteur de perte de puissance Φ :

$$F_T = \frac{\Phi \cdot P_E}{N \cdot D} \cdot \left(a + b \cdot \frac{V^2}{N^2 \cdot D^2} \right)$$

enfin, FacteurManette correspond à la position de la manette des gazs:

FacteurManette = 0 : manette en position basse
FacteurManette = 1 : manette en position haute

On ajoute ce facteur à l'équation précédente:

$$F_T = FacteurManette \cdot \frac{\Phi \cdot P_E}{N \cdot D} \cdot \left(a + b \cdot \frac{V^2}{N^2 \cdot D^2} \right)$$

calcul de la traction résultante en fonction de l'assiette de l'avion:

$$F_{T'} = F_T - (m \cdot g) \cdot \sin\left(\alpha \cdot \frac{\Pi}{180} \right)$$

α correspond à l'assiette de l'avion en degrés
Une fois la poussée définie, on peut calculer l'accélération de l'avion At en m/s^2:

$$At = \frac{F_{T'}}{m}$$

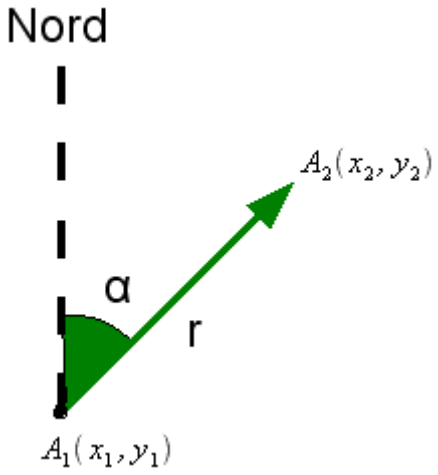
avec m : masse de l'avion en kg

Le régime moteur est proportionnel à la position de la manette des gazs. La position de la manette des gazs varie de 0 à 1000 :

$$RPM = \frac{regimeMaxRPM}{valMaxGouvernes} \times positionManetteDesGazs$$

V.5) Calcul de la position de l'avion sur la carte:

1 pixel = 0,1 mile nautique



$$x_2 = x_1 + r \cdot \cos\left[(\alpha - 90)\, mod\ 360 \cdot \frac{\Pi}{180}\right]$$

$$y_2 = y_1 + r \cdot \sin\left[(\alpha - 90)\, mod\ 360 \cdot \frac{\Pi}{180}\right]$$

α est le cap suivi par l'avion. r est proportionnel à la vitesse de l'avion:

$$V_r = V \cdot \frac{10}{3600}$$

avec Vr en pixels/seconde

V en kt

10 représente le nombre de pixels / miles nautiques

VI) Calculs liés à la radionavigation:

VI.1) Calcul d'un cap d'un point A vers un point B:

Les caps sont en degrés.

On a: $A(x_A, y_A)$ et $B(x_B, y_B)$

$$\Delta_x = x_B - x_A$$

$$\Delta_y = y_B - y_A$$

Soit z un facteur de signe:

Si $(\Delta_x < 0)$ OU $(\Delta_y < 0)$ faire
    z = -1
Sinon
    z = 1
FinSi

Soit r la distance A-B :

$$r = z \cdot \sqrt{\Delta_x{}^2 + \Delta_y{}^2}$$

Soit α_n les différentes valeurs du cap:

$$\alpha_1 = \left[ASin\left(\frac{y_A - y_B}{r}\right) \cdot \frac{180}{\pi} + 90 \right] mod\ 360$$

$$\alpha_2 = \left[ACos\left(\frac{x_A - x_B}{r}\right) \cdot \frac{180}{\pi} + 90 \right] mod\ 360$$

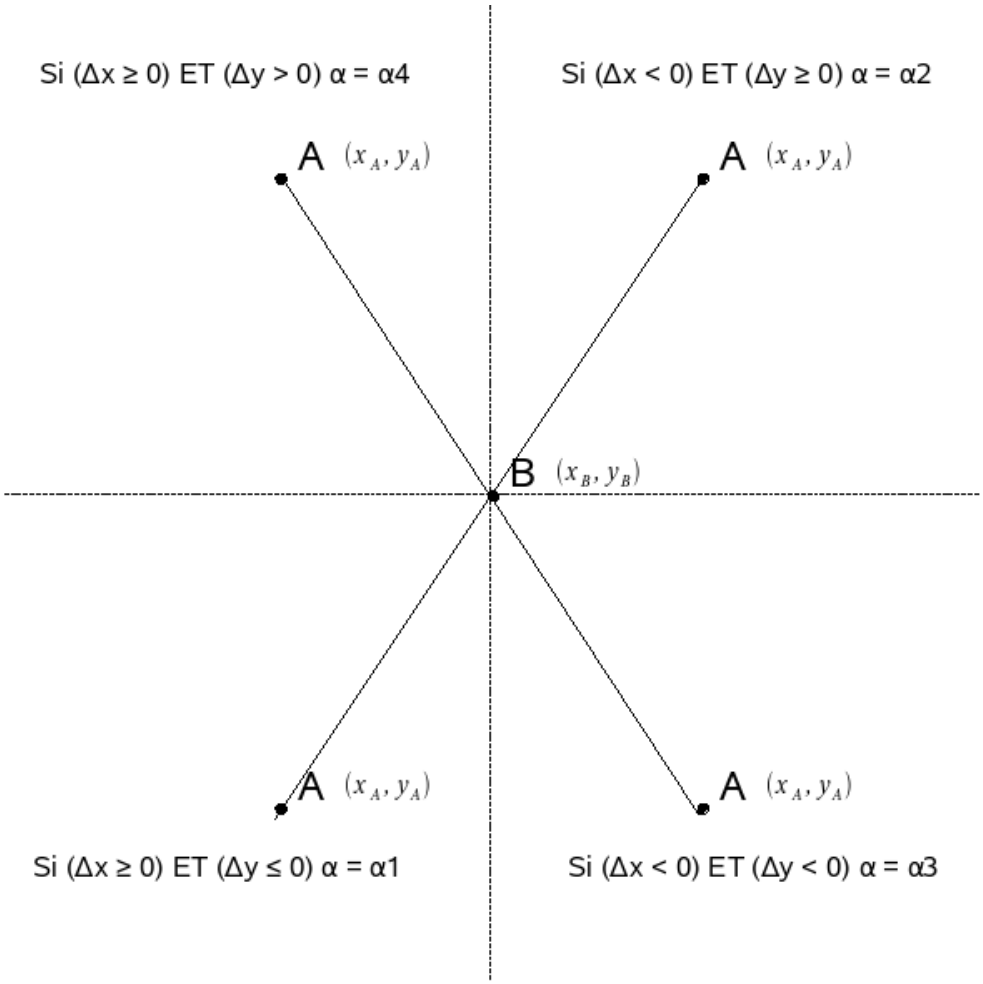
$$\alpha_3 = 360 - \alpha_1$$

$$\alpha_4 = 360 - \alpha_2$$

Rappel sur la conversion d'angle degrés-radians:

$$\alpha_{deg} = \alpha_{rad} \cdot \frac{180}{\pi} \qquad \alpha_{rad} = \alpha_{deg} \cdot \frac{\pi}{180}$$

Calcul d'α en fonction de la position de A:



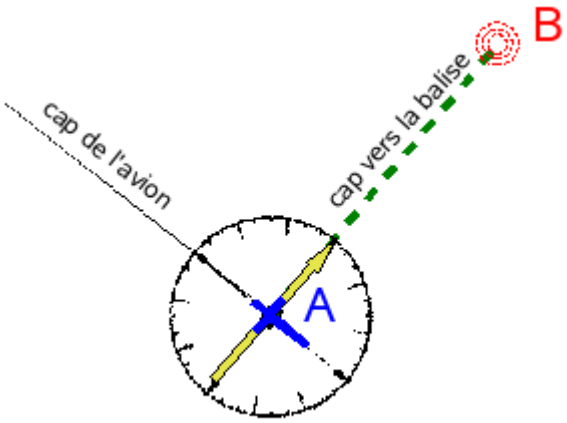
VI.2) Calculs pour l'ADF:

Db : direction de la balise par rapport à l'avion (suivant un cap)

Ca : cap de l'avion

Cp : cap de la balise (par rapport à la position de l'avion)

$$D_b = (C_p - C_a) \, mod \, 360$$



VI.3) Calculs pour les VORS

α : déviation

R : radiale sélectionnée sur le VOR

Cp : cap de la balise (par rapport à la position de l'avion)

$$\alpha = (C_p - R) \, mod \, 360 \quad \textbf{- 1}$$

Si α >270 ou α<90 alors l'avion est dans la zone TO

Si α <270 ou α>90 alors l'avion est dans la zone FROM dans ce cas, la formule est la suivante : $\alpha = 180 - (C_p - R) \, mod \, 360$ - 2

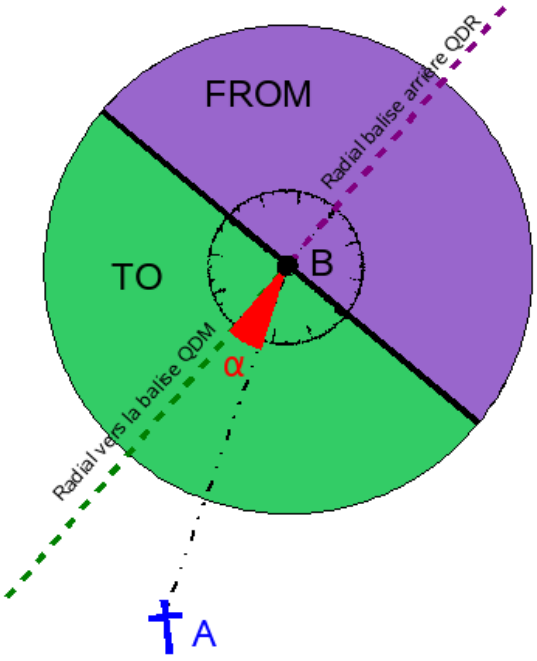
Pour éviter les erreurs de calcul, on ajoute 90° aux calculs précédents ce qui donne:

1 - $\alpha = (C_p - (R + 90)) \, mod \, 360 - 90$

2 - $\alpha = 270 - (C_p - (R + 90)) \, mod \, 360$

En effet: si Cp = 10 et R = 355 alors α = -345

En décalant de 90° on obtient Cp = 10 R =85 et α = -15.



VII) Tests du programme:

Lors de l'implémentation, le bon fonctionnement de chaque instrument a été testé. L'implémentation terminée, le programme a été testé sur différents ordinateurs et systèmes d'exploitation:

Test sur PC:

Système d'exploitation Windows XP SP2, processeur 2,5GHz, RAM:512Mo

Système d'exploitation Windows XP, processeur 1,7GHz, RAM:1 Go

Système d'exploitation Linux (OpenSuSE 10.2), processeur 1,7GHz, RAM:1 Go

Système d'exploitation Windows ME processeur 600 GHz, RAM:64Mo

Test sur MacOS X.

Le programme a été testé avec des fichiers ini au mauvais format ou non présents dans le répertoire de l'application.

Conclusion: dans la plupart des cas, le logiciel fonctionne correctement.

Seul le test sur Windows ME n'a pas été concluant. Le système de mesure du temps mis par la boucle principale ne fonctionne pas. La fonction *System.currentTimeMillis()* ne retourne aucune valeur. Il a donc fallut adapter le programme pour ce type d'erreur.

VIII) Conclusion:

VIII.1) Générale:

Le programme dans sa version actuelle comporte toutes les exigences de départ. Le simulateur est entièrement fonctionnel, permet de créer ses propres zones de vol et d'effectuer des exercices.

Passionné d'aviation (pilotage),j'ai bien évidemment pris beaucoup de plaisir à réaliser ce programme. C'était avant tout un défit personnel, car je partais sans aucune connaissance en aérodynamique et mécanique du vol. Il a donc fallut me documenter sur le sujet.

VIII.2) Problèmes rencontrés:

Dans un premier temps, lors du test des instruments de bord, j'ai rencontré un problème d'affichage: clignotement de l'affichage. En effet, les dessins (ligne, cercle,...) sont affichés les uns après les autres sur le contexte graphique; d'où cet effet de clignotement. La solution proposée a donc été de créer une classe qui permet de gérer l'affichage en double buffering: le dessin est réalisé dans une image en mémoire puis cette image est affichée à l'écran.

J'ai rencontré également quelques difficultés pour la régulation temporelle de la boucle principale. Cette régulation est importante, car tous les calculs d'accélérations et autre sont basé dessus. La boucle est régulée à l'aide de la méthode sleep() à un interval de 50ms. Malheureusement, la méthode sleep() arrête l'exécution pour un minimum de 50ms, rien n'empêche un arrêt supérieur à 50ms. Pour tenir compte de cela, j'ai mis en place dans la boucle principale un système de mesure qui permet de connaître le temps en ms réellement mis par la boucle. C'est ce temps qui est utilisé pour les calculs. Sous Windows ME (cf. VII p37), le système de mesure est inopérant.

VIII.3) Évolutions:

Actuellement, le simulateur est basé sur un modèle de vol simplifié qui mériterai d'être amélioré. La vitesse verticale de l'avion est beaucoup trop sensible pour une légère modification de l'assiette. L'implémentation des moments de roulis et de tangage implique la prise en charge du joystick par le logiciel (pilotage trop difficile au clavier), peut être à l'aide de la librairie Jinput.

La simulation de l'atmosphère mérite aussi amélioration, afin de gérer le vent, important dans le vol sans visibilité

Il faudrait également ajouter des infrastructures aéroportuaires : pistes et balises pour l'atterrissage aux instruments.

Enfin, même s'il s'agit d'un simulateur de vol aux instruments, il serai envisageable d'ajouter un visuel 3D (API Java3D).

IX) Bibliographie:

- **Programmer en Java**, *Claude Delannoy*, EYROLLES
- **Le livre de Java premier langage**, *Anne Tasso*, EYROLLES
- **Aide-mémoire de Java**, *Vincent Granet, Jean-Pierre Regourd*, DUNOD
- **UML2 pratique de la modélisation**, *Benoit Charroux, Aomar Osmani, Yann Thierry-Mieg*, PEARSON Education
- **Manuel du pilote d'avion 5ème édition**, CEPADUES
- **La mécanique du vol de l'avion léger**, *Serge Bonnet, Jacques Verrière*, CEPADUES
- **Physics for Game Programmers**, *Grant Palmer*, APRESS
- http://www.aviationpassion.org/
- http://www.aeroclub-montpellier.com/Documents/docNavigation.pdf