

Introduction

Dans le jeu vidéo Arma 3, les tireurs d'élite ont parfois besoin d'éliminer un certain nombre de cibles à la suite. Ils ont pour cela besoin de noter certaines informations sur la cible (distance, différence d'élévation, vitesse, emplacement...) et doivent également effectuer certains calculs pour pouvoir ajuster un tir précis pour des distances allant généralement de 600 à 2000 mètres. Le but de l'application sera de leur faciliter ce processus en leur permettant, grâce à la reconnaissance vocale, de dicter ces informations à l'application qui effectuera elle-même les calculs, puis d'obtenir les ajustements nécessaires sans avoir à se détacher du jeu.

Table des matières

Introduction	1
Table des matières	1
Contexte	2
Arma 3	2
Elévation et distance réelle	3
Zérotagage	5
Mildots	7
Tableau d'ajustement	9
Résumé du système de tir	10
But de l'application	11
Processus d'utilisation	11
Fonctionnalités	13
Maquettes	14
Planning prévisionnel	15
Diagramme des classes	16

Contexte

L'application ayant pour but de faciliter les calculs d'un certain type de joueurs sur un jeu vidéo, il est tout d'abord nécessaire d'expliquer le fonctionnement dudit jeu vidéo afin de pouvoir déterminer l'utilité de l'application.

Arma 3

Le jeu vidéo Arma 3 est la simulation militaire la plus réaliste sortie à ce jour. Dans ce jeu il est possible de jouer en tant que sniper (ou tireur d'élite) et de réaliser des tirs allant jusqu'à 2500 mètres. Dans la réalité de tels tirs impliquent de nombreux paramètres : distance, altitude et différence d'élévation entre la cible et le tireur, longitude, hygrométrie, pression atmosphérique, direction et force du vent, effet de Coriolis...

Arma 3 étant avant tout un jeu vidéo, les développeurs ont fait le choix de simplifier le système de balistique pour permettre aux joueurs de s'amuser sans avoir à s'entraîner pendant plusieurs mois. Ainsi, les seuls paramètres pris en compte pour un tir longue distance dans Arma 3 sont la distance et la différence d'élévation (altitude) entre la cible et le tireur, ainsi que le temps de vol de la balle.

Afin de réussir son tir, le tireur a donc besoin de connaître la distance à sa cible ainsi que la différence d'élévation entre elle et lui. Il utilise pour cela un désignateur laser :

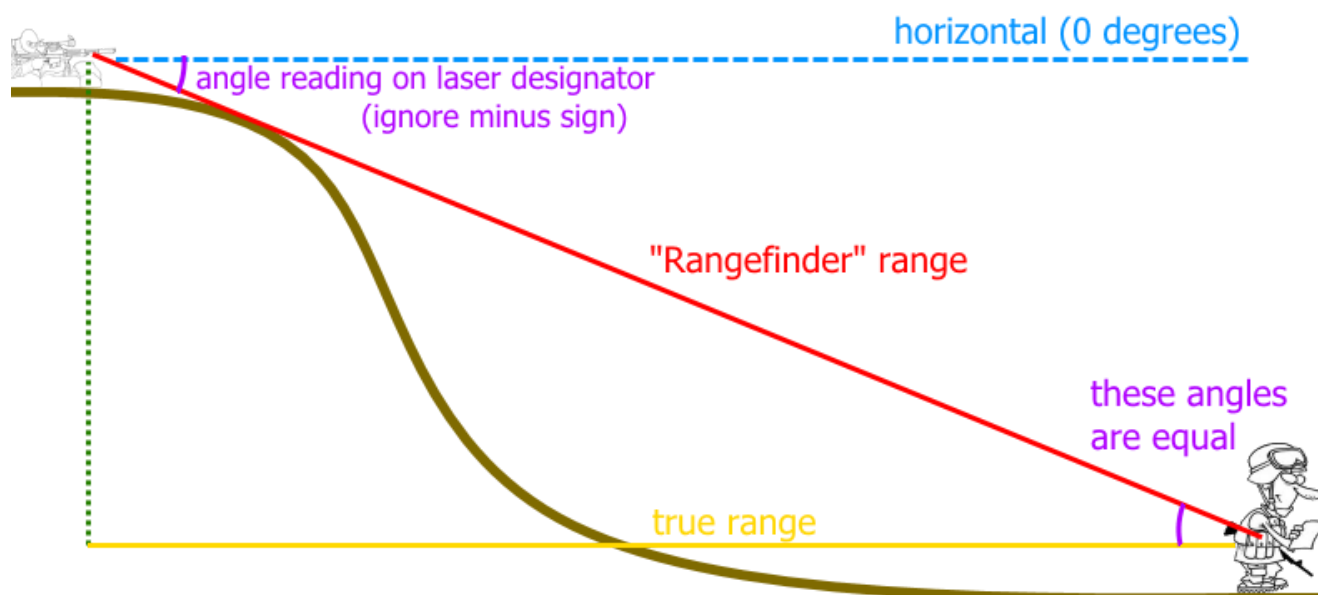


Dans la capture d'écran ci-dessus, le tireur utilise son désignateur laser pour déterminer sa distance et sa différence d'élévation avec la cible. Le rectangle vert montre les informations qui nous intéressent :

- 1222 RNG : distance en mètre à la cible pointée par la croix rouge au centre de l'écran. Ici, le soldat de gauche se trouve à une distance de 1222 mètres.
- -07.74 ELE : valeur en degrés indiquant l'angle entre la cible, le tireur et une ligne parallèle au sol passant par le tireur (voir paragraphe suivant pour les explications sur l'élévation). Le tireur doit donc prendre en compte un angle de 7,74 degrés, négatif car il se trouve plus haut que sa cible.

Elévation et distance réelle

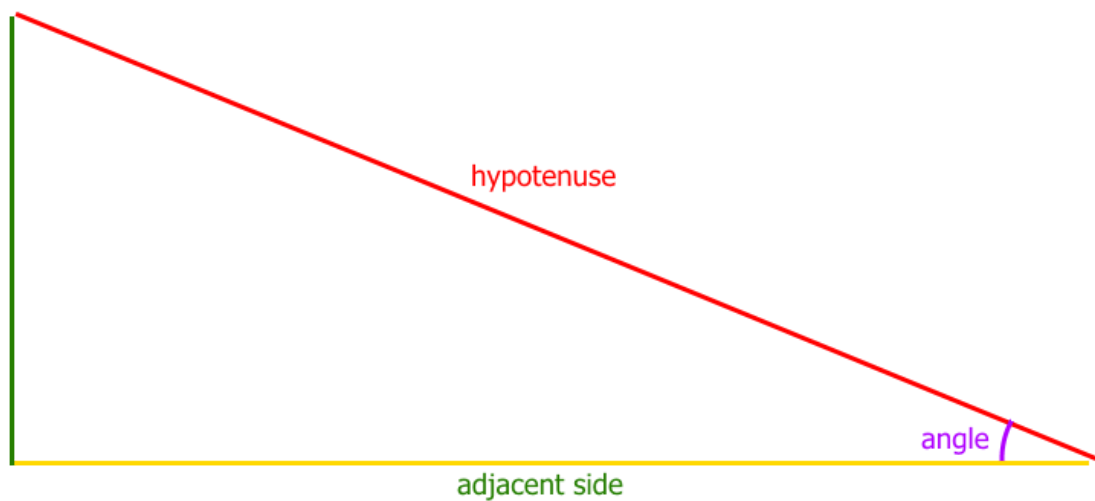
Quelques explications sur l'élévation s'imposent. L'angle en degrés appelé "élévation" permet au tireur de calculer la distance réelle entre sa cible et lui. Le schéma ci-dessous illustre ce calcul :



Le tireur se trouve en haut à gauche, la cible en bas à droite.

- **"Rangefinder" range** : la distance mesurée en mètres par le désignateur laser est un simple segment allant directement du tireur à la cible et ne tient pas compte de la différence d'élévation entre les deux
- **Horizontal (0 degrees)** : représente une droite parallèle au sol et passant par le tireur. S'il vise un point situé sur cette droite, son désignateur laser lui indiquera un **angle** de 0 degrés en élévation.
- **True range** : la distance réelle, obtenue par un calcul utilisant la **distance mesurée par le désignateur laser** et l'**angle d'élévation**
- **Ligne verte** : perpendiculaire au sol et passant par le tireur, elle indique la verticale parfaite
- **Angles** : la trigonométrie de collège nous rappelle que l'angle mesuré entre l'**horizontale**, le tireur et la cible est égal à celui mesuré entre le sol, la cible et le tireur

Un simple calcul de trigonométrie, illustré par le schéma ci-dessous, permet donc de calculer la distance réelle à la cible :



La distance réelle correspond donc au “côté adjacent” illustré sur la figure ci-dessus :
 Côté adjacent = Hypoténuse x cos(angle)

distance réelle = distance mesurée x cos(élévation)
--

Pour en revenir à notre exemple, avec une cible à 1222 mètres et une élévation de -07,74, le calcul est :

Distance réelle = distance mesurée x cos(élévation)

Distance réelle = 1222 x cos(-7.74)

Distance réelle = 1210.8 mètres

En tenant compte de l'élévation, on se rend compte que la distance passe de 1222 mètres à 1211 mètres, ce qui peut donner une différence significative pour certains tirs, raison pour laquelle l'élévation ne sera pas à négliger.

On connaît donc la véritable distance à la cible, celle que le tireur devra utiliser pour viser. Pour ce faire il devra ajuster son Zéro, ce qui s'appelle le “zéro tage”.

Zéro tage

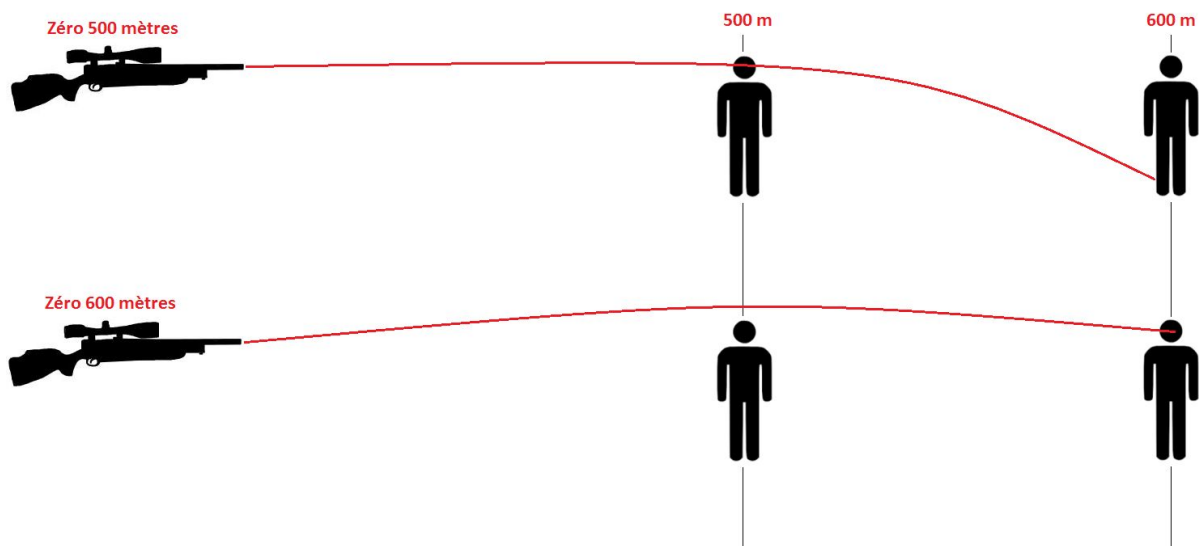
Contrairement à la croyance populaire, les armes ne tirent pas droit. Une balle décrit une trajectoire en cloche : dès sa sortie du canon elle s'élève sur une certaine distance à cause de sa rotation, puis redescend sous l'effet de la gravité.

Le zéro est une valeur exprimée en mètres. Très simplement, elle représente la distance à laquelle l'arme va tirer. Elle définit le réglage de la visée en élévation pour placer une balle dans le centre de la cible à une portée donnée. Par exemple :



Dans la capture d'écran ci-dessus, l'arme du tireur est zérotée à 500 mètres. Cela signifie que si le tireur souhaite toucher une cible à 500 mètres, il lui suffira de poser le centre de son viseur (le point rouge) sur sa cible et de tirer. En revanche, si la cible se trouve à une distance de 600 mètres, que le zéro est à 500 mètres et que le tireur vise la cible avec son point rouge, la balle tombera plus bas.

Le schéma ci-dessous résume le principe du zérotage :



Dans ce schéma, on imagine que le tireur vise à chaque fois le centre de la tête de sa cible avec son point rouge (centre du viseur).

- Si le zéro correspond à la distance à la cible : la balle atteindra l'endroit visé. Si on vise à 500 mètres avec un zéro à 500 mètres, la balle touchera notre cible.

- Si le zéro est plus petit que la distance à la cible : la balle ira plus bas que le point que l'on visait. Si on tire sur une cible à 600 mètres avec un zéro à 500 mètres, la balle le touchera dans les pieds au lieu de la tête
- Si le zéro est plus grand que la distance à la cible : la balle ira plus haut que le point que l'on visait. Si on tire sur une cible à 500 mètres avec un zéro à 600 mètres, la balle passera au-dessus de sa tête.

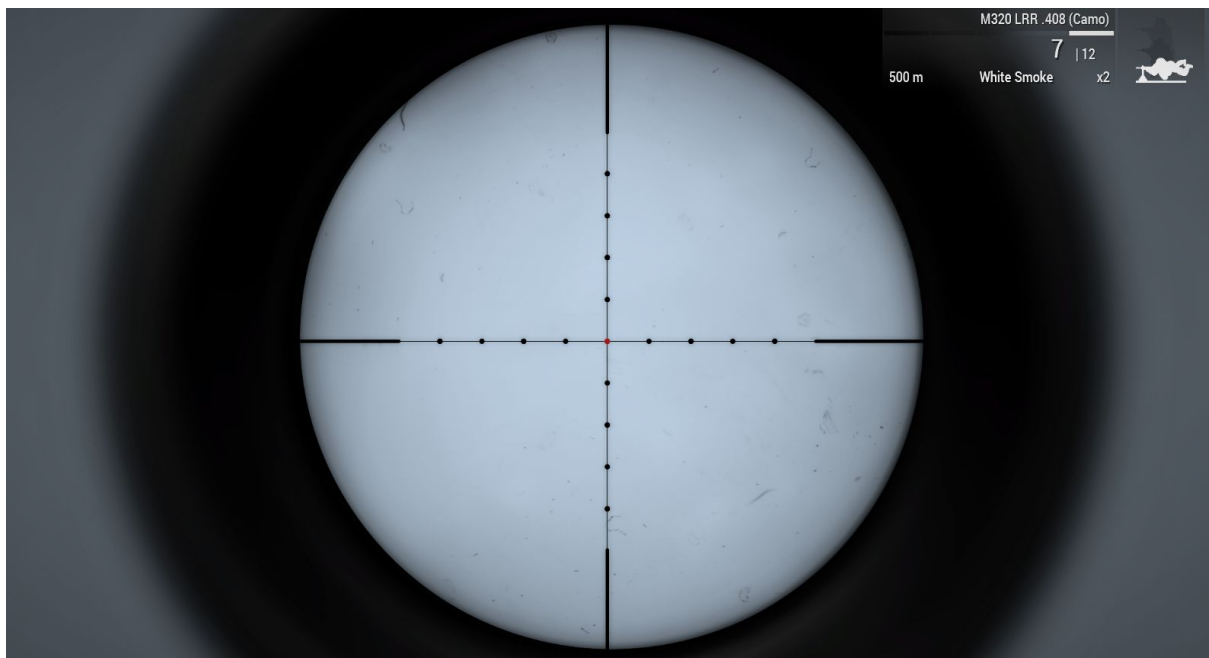
Le zéro ne pouvant s'ajuster qu'avec un pas de 100, si l'on veut toucher une cible à 550 mètres il faudra au choix :

1. Utiliser un zéro à 500 mètres et viser un petit peu au-dessus de la cible
2. Ou alors utiliser un zéro à 600 mètres et viser un petit peu en-dessous de la cible

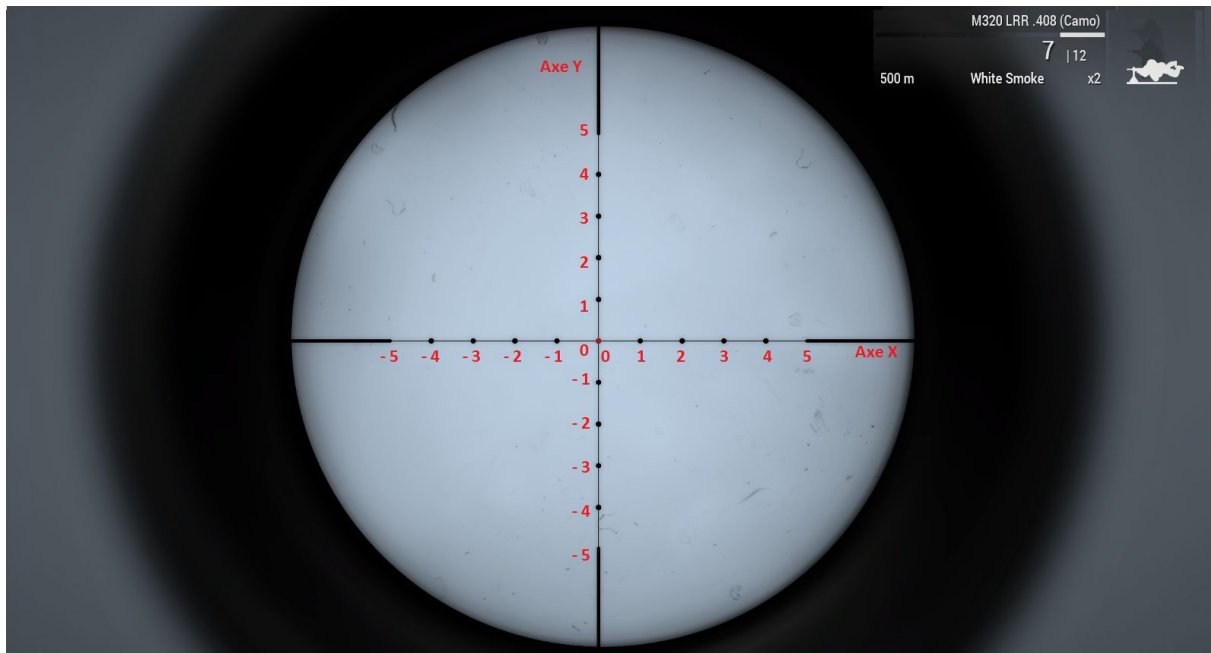
Les tirs longue distance se devant d'être très précis, il ne sera pas question de viser "un peu au-dessus" en espérant toucher sa cible. Il faudra utiliser les mildots pour déterminer un ajustement précis.

Mildots

Vous l'aurez peut-être remarqué, le viseur du fusil se comporte comme un axe orthonormé (axe vertical et axe horizontal avec des graduations identiques) :

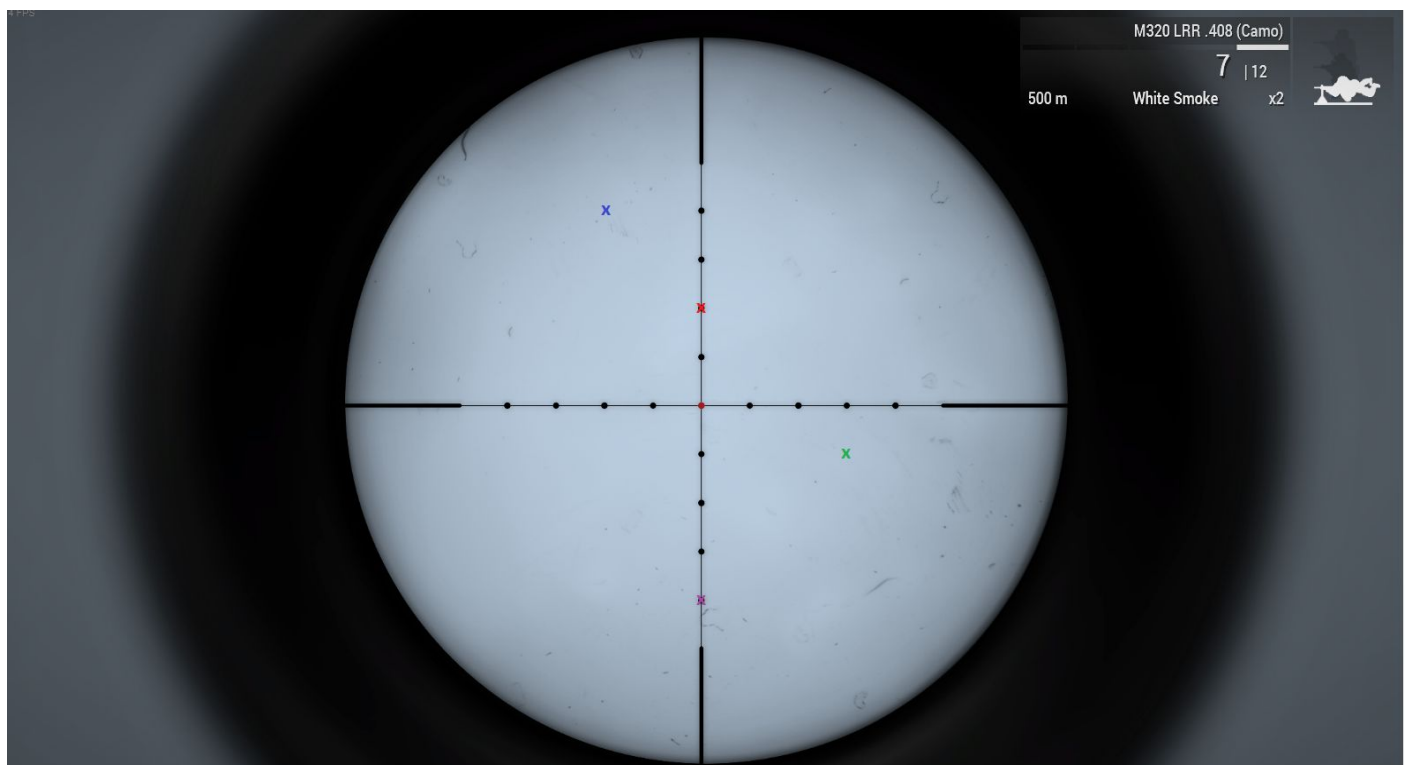


Il peut donc se représenter ainsi :



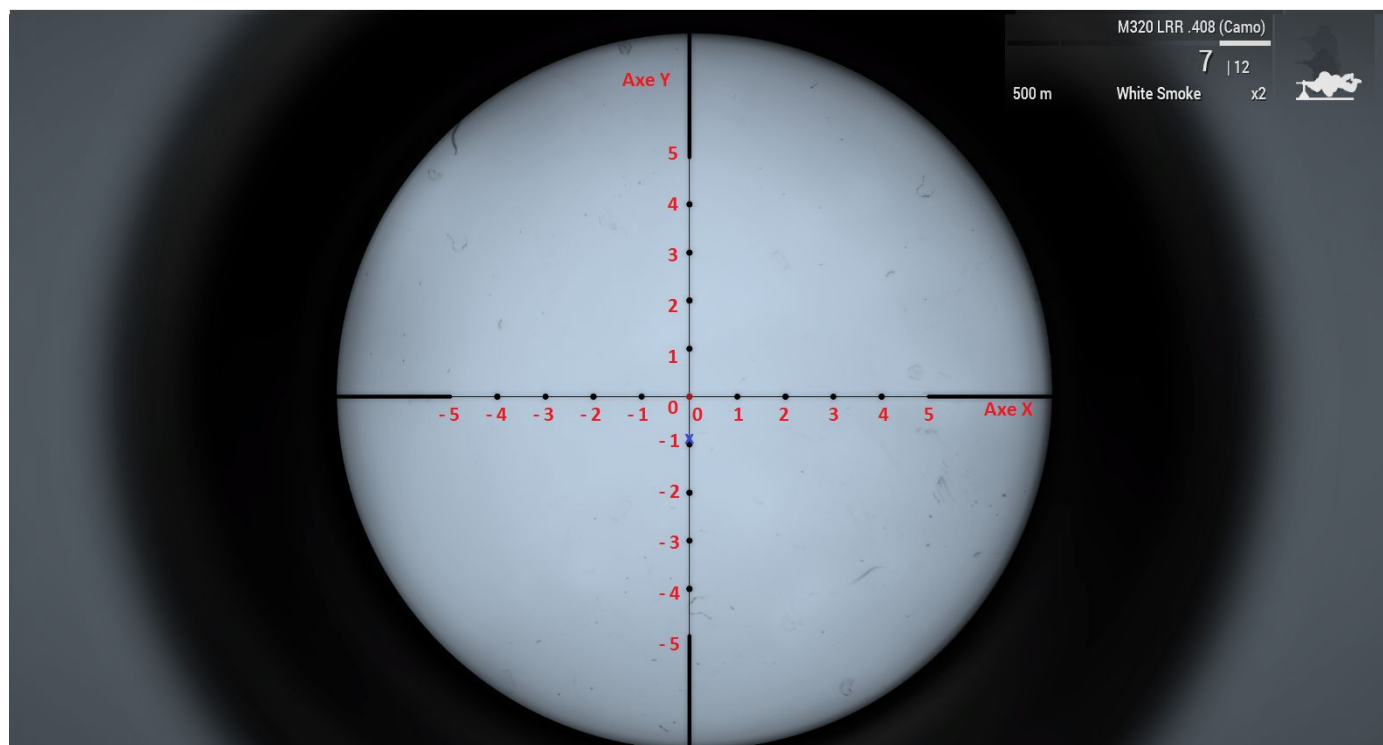
Chaque point représente 1 “mildot”. L’axe Y représente l’ajustement vertical et dépend de la distance, l’axe X représente l’ajustement horizontal et dépend de la vitesse de déplacement de la cible. Combiné à l’ajustement du zéro, cela va permettre au tireur de représenter ses ajustements pour viser une cible.

Voici par exemple quelques points de coordonnées :



(-2 ; 4) en bleu, (0 ; 2) en rouge, (0 ; -4) en violet, (3 ; -1) en vert

Ainsi, si le tireur souhaite atteindre une cible statique à 1050 mètres avec un zéro à 1000 mètres, il devra viser avec -0,9 mildots sur l'axe Y, et donc viser en utilisant un point imaginaire (**en bleu ci-dessous**) se trouvant à -0,9 sur l'axe Y :



(Avec un zéro à 1000 mètres, pour atteindre une cible à 1050 mètres on vise avec **le point bleu**, à -0,9 mildots en vertical)

Tableau d'ajustement

Le tableau suivant représente la majorité des ajustements en fonction du zéro choisi et de la distance visée :

-100 m	-75m	-50m	-25m	Ajust zero	Zero	+25m	+50 m	+75m	+100 m
0,3				-0,35	300 m	-0,4	-0,5		-1
0,5				-0,25	400 m		-0,6		-1
1				0,05	500 m		-0,7		-1
1				-0,2	600 m	-0,4	-0,8	-0,8	-1
1	0,75			-0,25	700 m		-0,8		-1
1			0,15	-0,25	800 m	-0,4	-0,8		-1
1	0,75	0,5			900 m	-0,5	-0,8		-1,5
1		0,5		-0,15	1000 m	-0,5	-0,9		-1,5
1,1		0,5		-0,25	1100 m	-0,5	-0,95		-1,6
1,15		0,45		-0,2	1200 m		-0,9		-1,7
1,5		0,5		-0,2	1300 m		-1		-1,5
1,75		0,6			1400 m		-1,1		-1,75

1,75		0,75			1500 m	-0,7	-1,1		-2
1,75	1,15	0,75			1600 m	-0,7	-1,15		-2
1,75	1,25	0,75		-0,3	1700 m		-1,2		-2,25
2		0,85			1800 m		-1,4		-2,5
2		1		-0,25	1900 m	-1	-1,45		-2,625
2,5		1		-0,25	2000 m		-1,5		-2,75
2,5		1,2			2100 m		-1,5		-3,2
					2200 m				
					2300 m				
					2400 m				-3,7

Chacune des cellules, hormis celles de la colonne "Zero", donne un ajustement vertical (axe Y) en mildots. Voici l'explication des différentes colonnes :

- Zero : le zéro choisi par le tireur, en mètres
- Ajust zero : les fusils n'étant pas parfaitement précis, il ne suffit pas de viser à 300 mètres avec un zéro à 300 pour toucher sa cible, il faut malgré tout parfois faire un petit ajustement. Ainsi, pour un zéro à 300 et une cible à 300 il faut viser avec -0,35 mildots.
- -100m : ajustement nécessaire pour tirer à 100 mètres de moins que le zéro choisi. Par exemple, avec un zéro à 300 mètres, il faut un ajustement de 0,3 (positif donc sur la partie haute de l'axe Y) pour toucher une cible à 200 mètres
- +100m : ajustement nécessaire pour tirer à 100 mètres de plus que le zéro choisi. Par exemple, avec un zéro à 300 mètres, il faut un ajustement de -1 pour toucher une cible à 400 mètres
- Les autres colonnes donnent les valeurs intermédiaires : -75m, +75m, -50m...

Tout n'est pas rempli car, Arma 3 ne fournissant pas d'explications sur ses fonctions de ballistique, il m'a fallu calculer ce tableau d'ajustement à la main, en tirant sur des cibles à différentes distances et à plusieurs reprises. Les valeurs intermédiaires peuvent généralement se calculer simplement : la cellule en rouge correspond à l'ajustement nécessaire pour une cible à 1875 mètres avec un zéro à 1900 mètres. L'ajustement n'est pas marqué, mais on peut rapidement calculer que 1875 étant à mi-chemin entre 1900 et 1850, l'ajustement pour 1875 doit se trouver à mi-chemin entre -0,25 et 1, ce qui nous donne un ajustement d'environ 0.6 pour cette case.

Résumé du système de tir

En résumé, le tireur doit obtenir la distance mesurée à sa cible et une différence d'élévation pour calculer la distance réelle entre sa cible et lui-même. Il calibre ensuite son zéro en fonction de la distance réelle et ajuste son tir à un certain nombre de mildots selon la valeur qu'il trouve dans son tableau d'ajustements.

But de l'application

Le but de l'application est donc de permettre au joueur de pouvoir sauvegarder et consulter les informations et ajustements correspondant à plusieurs cibles le plus rapidement possible.

Actuellement, lorsqu'un joueur a besoin de faire cela, il lui faut, pour chaque cible :

- Mesurer la distance et l'élévation
- Les noter
- Calculer la distance réelle
- Choisir un zéro
- Calculer l'ajustement nécessaire

Il notera ces informations pour chaque cible et devra, ensuite, réussir à les retrouver rapidement dans ses notes. Le but sera donc de faciliter et de fluidifier ce processus, de manière à ce que le joueur puisse saisir les informations sur ses cibles et soit en mesure de les retrouver le plus rapidement possible grâce à l'application qui les mettra en surbrillance.

Ainsi, le tireur n'aura plus besoin que de :

- Mesurer la distance et l'élévation
- Entrer les valeurs dans l'application
- Regarder le zéro et l'ajustement calculés par l'application

Pour cela, la saisie des informations et leur obtention se fera à l'aide de la reconnaissance vocale : l'utilisateur parlera pour donner des informations à l'application qui les affichera sur une fenêtre que l'utilisateur pourra placer sur un deuxième écran. Le processus d'utilisation imaginé est décrit dans la prochaine section.

Processus d'utilisation

L'application utilisera un ensemble de mot-clefs. Chaque parole prononcée par l'utilisateur et reconnue par l'application est notée en rouge. Elle reconnaîtra quatre processus de base :

- Création d'une nouvelle cible → On donne des informations pour créer une cible. Accessible par le mot-clef "**Nouvelle cible**".
- Modification d'une cible → On modifie les informations d'une cible. Accessible par le mot-clef "**Modifier cible**".
- Mode de tir → On demande les informations d'une cible pour obtenir les ajustements. Accessible par le mot-clef "**Engager ennemi**".
- Suppression d'une cible → On donne le nom de la cible à supprimer de l'application. Action irréversible. Accessible par le mot-clef "**Suppression cible**".

Une fois l'un de ces processus de base enclenchés, l'application entrera dans une succession de sous-processus : acquisition de la distance, modification de la vitesse, affichage en surbrillance des ajustements pour une cible... L'application sera en écoute

constante des mots-clef suivants : [Valider, Annuler]. Leur utilisation dépendra du processus en cours, mais ils permettront généralement de, respectivement, valider le processus courant et revenir au processus parent, ou annuler le processus et revenir au processus parent.

Processus : création de cible

- **Nouvelle cible** → Instanciation d'une nouvelle cible, attente de ses informations. Par défaut, ses valeurs sont : {Distance : 0 mètres, Elévation : 0, Vitesse : statique, Commentaire : chaîne vide}
- **Cible** → Attente du nom de la cible (pour simplifier il ne s'agira que d'entiers positifs)
- **1** → Mise à jour du nom de la cible qui devient "Cible 1", attente des autres informations.
- **Distance** → Attente de la distance : un entier positif suivi du mot-clef "**mètres**".
- **1222 mètres** → Mise à jour de la distance qui vaut 1222 mètres.
- **Elévation** → Attente de l'élévation : un flottant compris entre -90 et 90.
- **-9,44** → Mise à jour de l'élévation qui vaut -9,44.
- **Vitesse** → Attente de la vitesse : une chaîne de caractère parmi la liste [**statique**, **recherche**, **patrouille**, **course**]. Tout autre mot entendu donnera une vitesse statique.
- **Recherche** → Mise à jour de la vitesse qui devient "recherche".
- **Commentaire** → Attente du commentaire : une chaîne de caractère se finissant par le mot-clefs "**Terminé**".
- **TP en haut de la tour sud, terminé** → Mise à jour du commentaire qui devient "TP en haut de la tour sud".
- **Valider** → Si l'on est pas en écoute d'une information, l'application valide la cible et se remet en attente d'un des processus de base.
- **Annuler** → Si l'on est en écoute d'une information, l'application en annule la saisie et attend un mot-clef. Si l'on est pas en écoute d'une information, l'application supprime la cible en cours et revient à la base.

Processus : modification de cible

- **Modifier cible** → Attente du nom d'une cible : "**Cible**" suivi d'un entier positif.
- **Cible 1** → Entrée dans la modification de la cible 1. Attente des mots-clefs "**Distance**", "**Elévation**", "**Vitesse**" ou "**Commentaire**".
- A la prononciation de l'un de ces mot-clef, l'application attend la nouvelle valeur. Le mot-clef "**annuler**" permet d'annuler la nouvelle saisie et de revenir à l'ancienne valeur.
- **Valider** → Sort du processus de modification.

Processus : mode de tir

- **Engager ennemi** → Attente du nom d'une cible : "**Cible**" suivi d'un entier positif.
- **Cible 1** → Affiche en surbrillance les informations et ajustements pour la cible 1.
- **Cible 2** → Affiche en surbrillance les informations et ajustements pour la cible 2.

- **Elimination confirmée** → Marque la cible 2 (dernière cible demandée) comme éliminée.
- **Terminer** → Quitte le mode de tir.

Processus : suppression de cible

- **Supprimer cible** → Attente du nom d'une cible : "**Cible**" suivi d'un entier positif.
- **Cible 1** → Supprime définitivement la cible 1.

Fonctionnalités

Il devra être possible de :

- Créer une nouvelle cible
 - Lui attribuer un nom
 - Lui attribuer une distance
 - Lui attribuer une élévation
 - Lui attribuer une vitesse : statique, recherche, patrouille, course
 - Lui attribuer un commentaire (indication textuelle permettant à l'utilisateur de reconnaître la cible d'un simple coup d'oeil)
- Obtenir les informations sur une cible :
 - Nom
 - Distance
 - Elévation
 - Commentaire
- Obtenir les ajustements nécessaires pour une cible :
 - Zéro à choisir
 - Ajustement en mildots à la verticale et l'horizontale dépendant de la distance, élévation et vitesse de la cible
- Marquer une cible comme éliminée : l'application la conservera en mémoire afin que les informations soient toujours accessibles si une autre cible se déplaçait au même endroit
- Modifier les informations d'une cible

La reconnaissance vocale sera étudiée lors de la première phase de développement. Si le système se révèle trop peu performant ou trop difficile à mettre en place, la partie reconnaissance vocale sera supprimée et l'utilisateur devra se contenter d'entrer les informations à l'aide de champs texte. Une section sera alors ajoutée à ce document pour revenir sur la conception et les processus.

L'application devra constamment afficher son processus courant : "Création de cible", "Saisie de distance : cible 2", "Mode de tir", "Mode de tir : cible 1", "Suppression de cible"...

Une fonctionnalité supplémentaire pourra éventuellement être d'afficher constamment ce que l'application reconnaît et pourrait servir au debug.

Maquettes

Ecran général de l'application :

Affichage du processus courant : "Création de cible", "Mode de tir", "Cible 1 : Saisie de la distance"...									
<div><div><div>Cible 1</div><div>Distance : <input type="text" value="1222 mètres"/> Elévation : <input type="text" value="- 9,44"/></div><div>Vitesse : <input type="text" value="Statique"/> Commentaire : <input type="text" value="TP au pied de la tour sud"/></div></div><div><div>Zéro</div><table border="1"><thead><tr><th>1200 m</th><th>1300 m</th></tr></thead><tbody><tr><td>X : 0 Y : - 0,1</td><td>X : 0 Y : + 1,4</td></tr></tbody></table></div></div>						1200 m	1300 m	X : 0 Y : - 0,1	X : 0 Y : + 1,4
1200 m	1300 m								
X : 0 Y : - 0,1	X : 0 Y : + 1,4								
<div><div><div>Cible 2</div><div>Distance : <input type="text" value="1222 mètres"/> Elévation : <input type="text" value="- 9,44"/></div><div>Vitesse : <input type="text" value="Statique"/> Commentaire : <input type="text" value="TP au pied de la tour sud"/></div></div><div><div>Zéro</div><table border="1"><thead><tr><th>1200 m</th><th>1300 m</th></tr></thead><tbody><tr><td>X : 0 Y : - 0,1</td><td>X : 0 Y : + 1,4</td></tr></tbody></table></div></div>						1200 m	1300 m	X : 0 Y : - 0,1	X : 0 Y : + 1,4
1200 m	1300 m								
X : 0 Y : - 0,1	X : 0 Y : + 1,4								
<div><div><div>Cible 7</div><div>Distance : <input type="text" value="1222 mètres"/> Elévation : <input type="text" value="- 9,44"/></div><div>Vitesse : <input type="text" value="Statique"/> Commentaire : <input type="text" value="TP au pied de la tour sud"/></div></div><div><div>Zéro</div><table border="1"><thead><tr><th>1200 m</th><th>1300 m</th></tr></thead><tbody><tr><td>X : 0 Y : - 0,1</td><td>X : 0 Y : + 1,4</td></tr></tbody></table></div></div>						1200 m	1300 m	X : 0 Y : - 0,1	X : 0 Y : + 1,4
1200 m	1300 m								
X : 0 Y : - 0,1	X : 0 Y : + 1,4								
Optionnel : affichage constant des mots que l'application reconnaît en temps réel									

Elle se composera d'un seul écran.

Une bande horizontale, au sommet, affichera le processus courant.

Une bande horizontale en bas sera éventuellement mise en place pour afficher le texte écouté et reconnu en temps réel par l'application.

Les cibles sont représentées par une liste de rectangles. Chacun contient :

- Le nom de la cible au sommet
- Ses informations (distance, élévation, vitesse et commentaire). Elles seront affichées dans des inputs pour signifier à l'utilisateur qu'une modification est possible
- Le zérotage, un rectangle séparé en deux parties : une partie pour le zéro "bas" et une partie pour le zéro "haut" (pour une distance de 1222m le zéro bas est 1200m et le zéro haut est 1300m) avec pour chacune l'ajustement en X et en Y.

Dans le cas où l'utilisateur ajouterait trop de cible pour que l'application les affiche toutes en même temps, le scroll se comportera de la manière suivante :

- A l'ajout d'une cible l'application scrollera au niveau de cette cible
- En mode de tir l'application scrollera au niveau de la cible annoncée

Si je me rend compte à l'usage qu'il est difficile d'utiliser ce système de scrolling pour l'utilisateur, l'affichage sera modifié de la manière suivante :

- Chaque cible n'aura que son nom, sa distance, sa vitesse et son commentaire d'affichés
- En mode de tir, lorsque l'utilisateur prononcera le nom d'une cible l'application affichera tous les détails de cette cible

Ajout de nouvelle cible :

Affichage du processus courant : "Création de cible", "Mode de tir", "Cible 1 : Saisie de la distance"...

Nouvelle cible : Cible 1

Distance : Elévation :

Vitesse : Commentaire :

Optionnel : affichage constant des mots que l'application reconnaît en temps réel

Le nom de la cible a été donné (cible 1), et l'utilisateur vient de prononcer le mot-clef "Distance" : l'application attend une valeur entière suivie de "mètres".

Mode de tir :

Affichage du processus courant : "Création de cible", "Mode de tir", "Cible 1 : Saisie de la distance"...

Cible 1 **Zéro**

Distance : Elévation :

Vitesse : Commentaire :

1200 m	1300 m
X : 0 Y : - 0,1	X : 0 Y : + 1,4

Cible 2 **Zéro**

Distance : Elévation :

Vitesse : Commentaire :

1200 m	1300 m
X : 0 Y : - 0,1	X : 0 Y : + 1,4

Cible 7 **Zéro**

Distance : Elévation :

Vitesse : Commentaire :

1200 m	1300 m
X : 0 Y : - 0,1	X : 0 Y : + 1,4

Optionnel : affichage constant des mots que l'application reconnaît en temps réel

L'utilisateur a prononcé "Engager ennemi" suivi de "cible 2" : l'application lui affiche les ajustements nécessaires pour la cible 2.

Planning prévisionnel

Le planning est représenté par un diagramme de Gantt :

Tâche	6-12 nov	13-19 nov	20-26 nov	27 nov-3	4-10 dec	11-17 dec	18-24 dec	25-31 dec	1-7 jan	8-15 jan
-------	-------------	--------------	--------------	-------------	-------------	--------------	--------------	--------------	------------	-------------

				dec						
Tests de la reconnaissance vocale										
Implémentation des classes										
Processus d'ajout										
Processus de tir										
Processus de modification										
Processus de suppression										
Réalisation de l'interface										

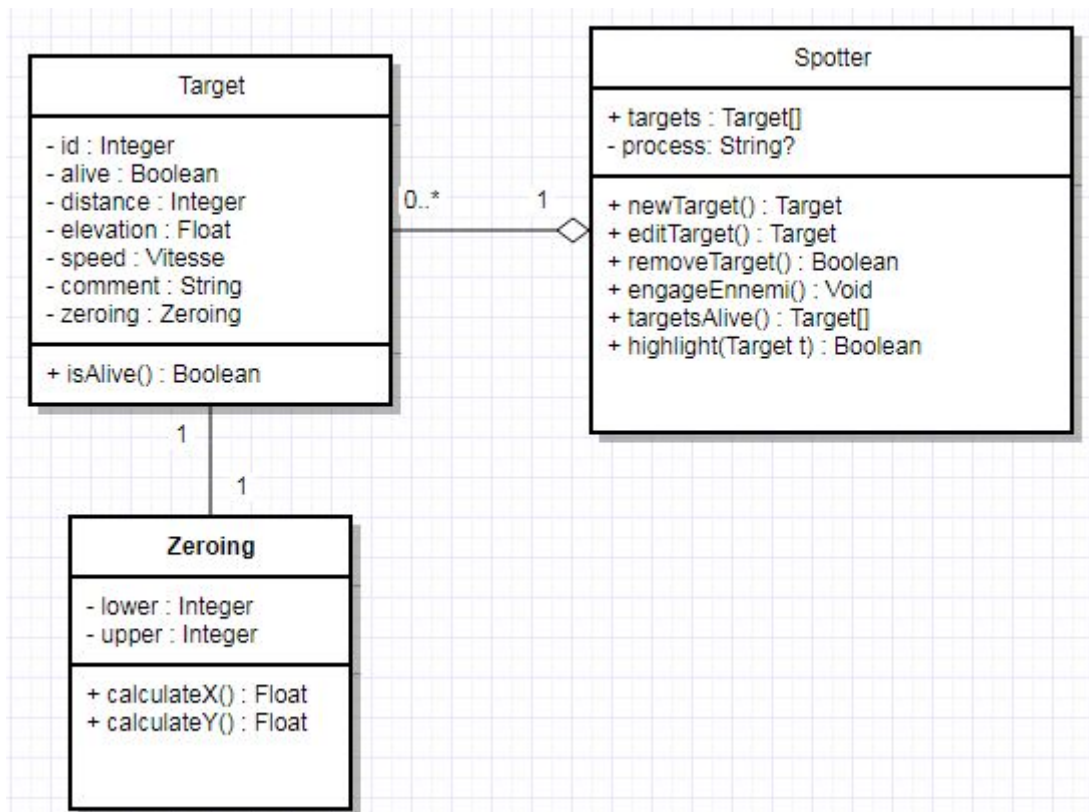
Les deux premières semaines de développement seront consacrées à l'analyse et les tests de la reconnaissance vocale en .net, ainsi que la recherche et l'analyse éventuelles de librairies permettant d'améliorer cette fonctionnalité.

L'interface sera travaillée tout au long : j'aurai très probablement des modifications à apporter au fur-et-à-mesure, n'ayant pas de connaissance en interfaces .Net.

Le processus d'ajout sera sans aucun doute celui qui prendra le plus de temps : c'est avec lui que j'utiliserai réellement mes classes et la reconnaissance vocale, et ce sera le premier processus. Les suivants commenceront une fois qu'il sera terminé, et leurs durées de réalisation dépendront sans aucun doute de leur similarité avec le processus de création de cible.

Il est possible que je rencontre un problème sur le focus : la reconnaissance vocale ne fonctionnera peut-être pas si le focus n'est pas sur la fenêtre de l'application (ce qui peut être gênant car en situation le focus sera sur la fenêtre du jeu vidéo). Si ce problème se présente, il ne sera abordé qu'en toute fin du projet une fois tout le reste fonctionnel.

Diagramme des classes



On considère que les getters, setters et constructeurs seront implémentés, inutile de les faire apparaître.

La classe Spotter sera la classe contenant les cibles et gérant les processus. Il y aura sans aucun doute besoin d'une classe gérant l'interface graphique mais, n'ayant encore jamais travaillé là-dessus, je ne sais pas comment la modéliser et verrai donc en temps utile.

Le Spotter contient un ensemble de cibles. Son attribut processus permet de savoir quel est le processus courant.

Une Target possède un Zeroing. Un Zeroing a un zéro bas et un zéro haut (lower et upper), dépendant de la distance avec laquelle il est instancié : si on instancie un ajustement avec une distance (passée en paramètre du constructeur) de 1250 mètres, le zéro bas vaudra 1200 et le zéro haut 1300. Ainsi le tireur pourra choisir le zéro qui lui convient.

Les méthodes du Zeroing permettent de calculer les ajustements en x et y pour chaque zéro.

Afin d'effectuer les calculs d'ajustements, l'application aura besoin de connaître le tableau montré dans la partie "Tableau d'ajustement". L'idéal serait de pouvoir parser ce type de tableaux depuis un csv ou un xls, mais cela risque de prendre trop de temps pour une fonction non indispensable.

Les valeurs d'ajustement seront donc stockées en dur dans un objet de l'application, probablement sous la forme d'un tableau associatif. Calculer les ajustements reviendra à comparer la distance et le zéro choisis à ceux stockés en dur.