
RAPPORT PROJET ALGO1 : LES BOITES

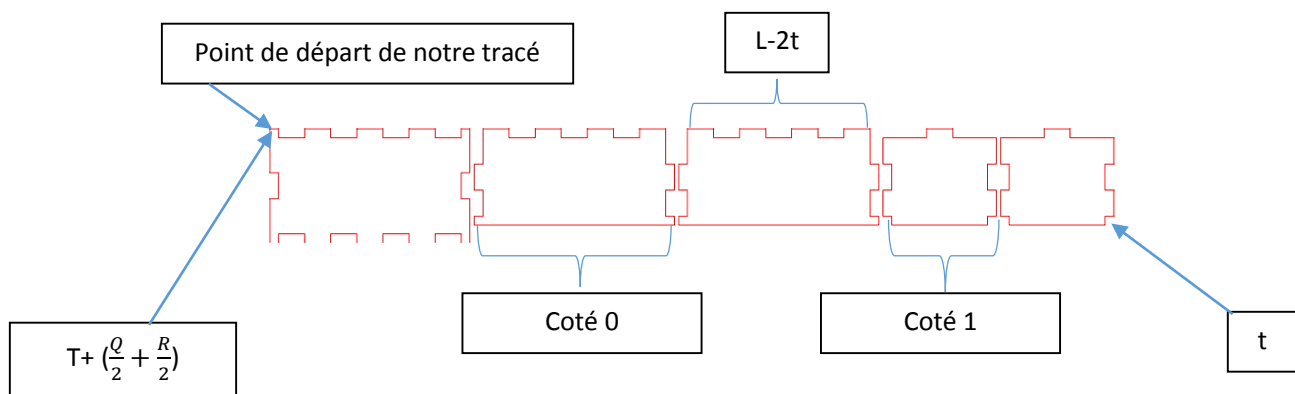
Bastien Roos

Paul Bouton

I-Réflexion théorique

Lors de ce projet, nous avons réfléchi à la meilleure manière de découper le problème : il fallait récupérer les grandeurs données en arguments dans le terminal et tracer les quinze faces de la boîte. Ainsi, Nous avons créé deux paquets : `funcboites` et `recup_donnees`. Afin de dessiner un fichier SVG, nous avons choisi d'utiliser les paquets SVG et dessin utilisés en TP (après en avoir eu l'autorisation). Dans le paquetage SVG, nous utilisons uniquement la fonction qui trace une droite. Le paquetage dessin contient 7 procédures : initialisation, fin, BC, LC, AV, TD, TG. Nous avons modifié ces procédures afin que les figures soient tracées dans le fichier SVG créé par notre programme.

Nous avons alors réfléchi à la réalisation du paquetage `funcboites`. Ce dernier devait tracer chaque face et chaque fond des demi-boîtes. Les trois demi-boîtes étant identiques, nous avons décidé d'utiliser une procédure traçant une seule demi-boîte, procédure que nous avons appelée trois fois. Afin de tracer les différents côtés, nous avons décomposé le problème en plusieurs parties :



On utilise trois procédures : Une première pour tracer le fond, une autre pour tracer les côtés et une dernière pour tracer les créneaux (utilisée dans les deux précédentes). La seconde nous permet de tracer les deux types de cotés (largeur et longueur). De plus, nous avons adopté une convention: On commence toujours par tracer un créneau commençant par une encoche à partir du point de départ puis on trace un créneau commençant par un plat sur le côté suivant et ainsi de suite. Effectivement, lors du tracé de créneau, on peut choisir plusieurs configurations de départ (encoche, queue ou plat suivis d'une encoche ou d'une queue). Cela nous permet de prédire la configuration des créneaux inférieurs (ceux en contact avec le fond) des côtés. Une procédure auxiliaire calcule le nombre d'encoches et de queues à tracer sur chaque côté et nous permet d'en déduire la longueur à laisser « plate » de chaque côté du créneau.

II- Mise en pratique

Afin de clarifier au maximum notre code, nous avons utilisé un maximum de packages (4 au total). En plus des packages dessin et SVG, un package funcboites permet de tracer les demi-boîtes et un package recup_donnees permet de lire la ligne de commande. Ainsi, notre fichier boites.adb ne contient qu'une quarantaine de lignes de code : Il teste la validité des données rentrées et appelle trois fois la fonction trace_boite. En effet, pour que la boîte soit conforme il faut que $h-2t > b$ (condition donnée) mais aussi que l'on puisse tracer au moins une queue ou une encoche par côté ($Q \leq b-2t$, $Q \leq W-4t$, $Q \leq L-4t$). Le package recup_donnees nous permet également de créer un fichier SVG dans lequel les patrons seront tracés. Ainsi, nous avons été obligés de modifier les procédures des Packages SVG et dessin afin que les tracés soient enregistrés dans le fichier SVG.

La convention « coté 0 coté 1 » est paramétrée par un entier (coté) valant 0 ou 1. Ce dernier nous permet, en intervenant dans les calculs (calcul des angles, calcul du créneau) de tracer le côté souhaité.

Enfin, nous avons utilisé des variables globales dans les différents packages afin de mémoriser les données importantes pour le tracer (nombre de créneaux (Nl et Nw), Ajout de $\frac{Q}{2}$ (Pl et Pw) ou $\frac{R}{2}$ aux extrémités du créneau.).

III-Problèmes rencontrés

Premièrement, le typage très strict d'ADA nous a contraint à modifier souvent les types des données. En effet, lors du calcul du nombre d'encoches et de queues à tracer sur un côté, nous devons utiliser des entiers afin de réaliser une division euclidienne. Cependant, pour avoir un tracé exact (Nous avons eu recours à des divisions), il a fallu effectuer des calculs avec des flottants. Nous avons remarqué que nous utilisions plus de flottants que d'entier dans nos différentes procédures, c'est pourquoi les données sont par défaut définies comme des flottants.

Avec notre méthode, nous évitons d'avoir des créneaux adjacents qui se touchent au niveau de la première encoche. En effet, si nous traçons un créneau commençant par une encoche sur la longueur, le créneau situé sur la largeur commencera par une queue. Cependant, dans le cas où $W-2t < 3Q$, alors $Nw=1$ et donc la largeur devient plate (un seul créneau commençant par un plat). Nous avons cependant considéré que ce tracé était correct et qu'il revenait à l'utilisateur de modifier ses paramètres afin d'obtenir une boîte plus stable.

IV- Conclusion

Lors de ce projet nous avons commencé par analyser théoriquement le problème ce qui nous a permis d'identifier les fonctions nécessaires à sa résolution. Ainsi, le codage a été facilité, l'objectif à atteindre ayant été fixé. En fin, afin de s'assurer du fonctionnement de notre programme, nous avons réalisé de nombreux tests notamment dans les cas limites (absence de données, données nul, format fichier incorrect,...).