

# Pictionary<sup>TM</sup> BOT

India Cabo &  
Clémence Carcanade

## Table des matières

Introduction.....	2
Présentation du système .....	3
Spécifications fonctionnelles .....	4
Description du comportement du robot.....	4
Configuration des GPIO .....	4
Structure du programme .....	5
Répartition des fichiers .....	5
Usage des registres.....	5
Diagramme de classe .....	6
Réflexion algorithmique.....	7
Choix de l'axe de rotation .....	7
Simplification géométrique des formes .....	7
Organigramme d'exécution .....	8
Difficultés rencontrées .....	9
Résultats .....	10
Pistes d'amélioration .....	11
Conclusion.....	12
Annexes.....	13
Formes géométriques mystères .....	13
Résultats des formes mystères.....	13
Organigramme d'exécution complet.....	14
Combinaison des formes géométriques.....	15
Modèle 3D du support du feutre .....	16
Programme complet.....	17

## Introduction

Dans le cadre du module « Architecture » de la première année du cycle ingénieur de la filière *Informatique et Applications* à l'ESIEE Paris, nous avons réalisé un projet basé sur l'utilisation d'une carte EVALBOT, afin de mettre en pratique la programmation en langage assembleur.

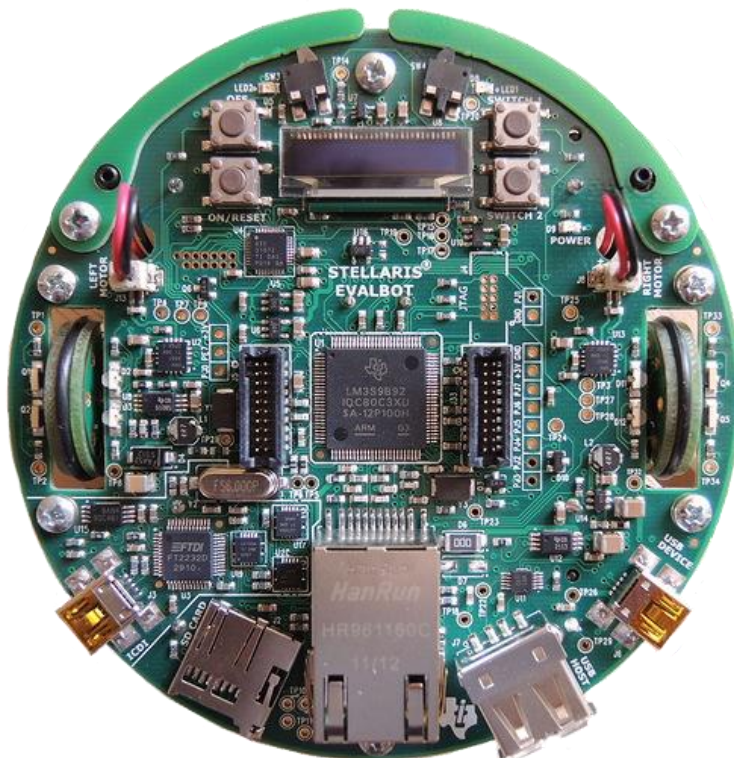
Dans ce contexte, nous avons conçu et développé un « *Pictionary Bot* », un robot capable de dessiner automatiquement des formes géométriques simples sur une feuille, en fonction des commandes de l'utilisateur.

## Présentation du système

Nous disposons d'une carte EVALBOT programmable en langage assembleur, présentant les caractéristiques matérielles suivantes :

- Un microcontrôleur *Stellaris® LM3S9B92* basé sur l'architecture ARM Cortex-M3, utilisé comme unité de calcul principale.
- Deux moteurs entraînant les roues gauche et droite.
- Deux capteurs de contact, appelés bumpers.
- Deux boutons poussoirs, appelés switches.
- Quatre LEDs, deux situées à l'avant et deux à l'arrière.
- Un haut-parleur
- Un écran OLED monochrome de résolution 96 x 16 pixels.
- Un port MicroSD, USB et Ethernet (RJ45).
- Une alimentation assurée soit par trois piles AA (1,5 V), soit via le port USB de l'interface ICDI.

Le cahier des charges du projet impose l'utilisation d'au minimum un moteur, un switch, un bumper et une LED.

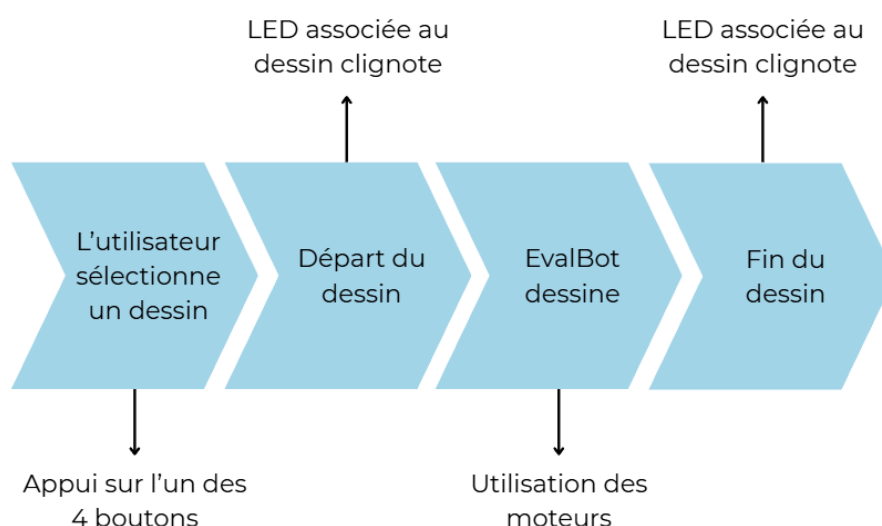


## Spécifications fonctionnelles

### Description du comportement du robot

Le « *Pictionary Bot* » est un robot de dessin interactif destiné à proposer une activité ludique à l'utilisateur. Il s'inspire du jeu « *Pictionary* », dans lequel un joueur doit faire deviner des mots ou expressions aux autres à l'aide de dessins. Dans notre cas, l'utilisateur doit deviner les formes dessinées par le robot.

Le robot, équipé d'un feutre fixé à sa structure, est en mesure de dessiner quatre formes différentes, sélectionnées à l'aide des éléments de commande disponibles, à savoir les deux switches et les deux bumpers. Les LEDS sont utilisées pour indiquer l'avancement du robot. Le scénario de fonctionnement global est présenté sous forme de schéma ci-dessous :



### Configuration des GPIO

Périphérique	Port et Broche	E/S	Fonction
Switch Haut	B - 0	Entrée	Lance la forme 1 → <b>Carré</b>   LED arrière gauche associée
Switch Bas	B - 1	Entrée	Lance la forme 2 → <b>Mystère</b>   LED arrière droite associée
Bumper Droit	E - 0	Entrée	Lance la forme 3 → <b>Etoile</b>   LED avant droite associée
Bumper Gauche	E - 1	Entrée	Lance la forme 4 → <b>Mystère</b>   LED avant gauche associée
LEDS	F - 2, 3, 4, 5	Sortie	Clignote individuellement 3 fois pour annoncer le départ et la fin du dessin Reste allumée pour signaler que le dessin est en cours de réalisation
Moteurs	D - 0, 1, 2, 5	Sortie	Avancent, reculent et tournent

## Structure du programme

### Répartition des fichiers

Par soucis de lisibilité et de qualité de code, le programme a été structuré en plusieurs fichiers distincts. Chaque périphérique utilisé (switchs, bumpers, LEDS et moteurs) dispose de son propre fichier, contenant les routines d'initialisation et les fonctions utilitaires associées.

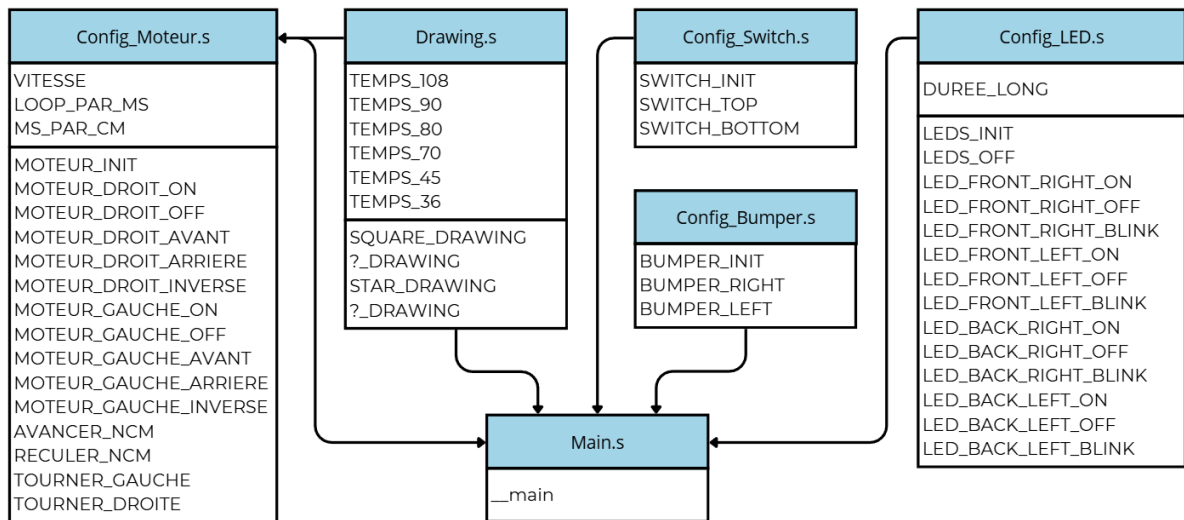
Le fichier *Drawing.s* regroupe les algorithmes responsables du tracé des quatre formes, tandis que le fichier *Main.s* constitue cœur du programme en assurant la liaison entre les périphériques et les fonctions de dessin.

### Usage des registres

Afin d'assurer une organisation claire du programme, une répartition précise des registres du microcontrôleur a été définie. La répartition adoptée est la suivante :

Registres	Usage
r0	Contient les variables d'angles et de distances
r1, r2, r3	Utilisés pour les calculs, et le stockage de variable
r4	Contient le nombre de répétitions dans les formes
r5	Stocke l'état du switch haut
r6	Stocke l'état du switch bas
r7	Stocke l'état du bumper droit
r8	Stocke l'état du bumper gauche
r9, r10	Sauvegarde le registre lr (registre de lien)

## Diagramme de classe



## Réflexion algorithmique

### Choix de l'axe de rotation

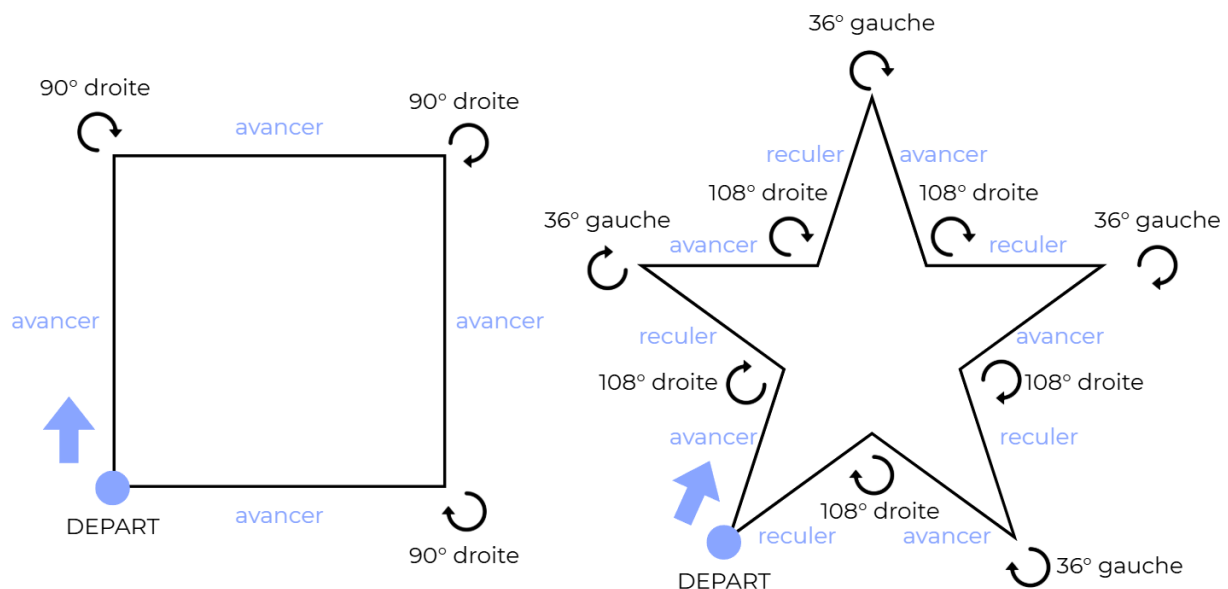
La problématique principale de notre projet concerne la précision du tracé des dessins. Il a donc été nécessaire d'adapter le code au besoin pratique, et de réfléchir à une solution pour accueillir un feutre et garantir un rendu exploitable. En particulier, l'intégration d'un feutre directement au centre du robot s'est révélée impossible.

Nous avons donc choisi de décaler l'axe de rotation du robot afin qu'il ne tourne plus autour de son centre géométrique, mais autour de la roue droite. Cette configuration permet de placer le feutre sur le côté droit du robot, à proximité de l'axe de rotation, ce qui limite les erreurs de trajectoire lors des rotations. En conséquence, les fonctions de rotation du programme n'utilisent que le moteur gauche.

À titre de précision, une rotation vers la droite est obtenue par la rotation avant du moteur gauche, tandis qu'une rotation vers la gauche est réalisée par le recul de ce même moteur.

### Simplification géométrique des formes

Les formes à dessiner ont été simplifiées sous la forme d'une succession de segments de droites et de rotations angulaires. Le schéma des formes géométriques ainsi obtenues est présenté ci-dessous :

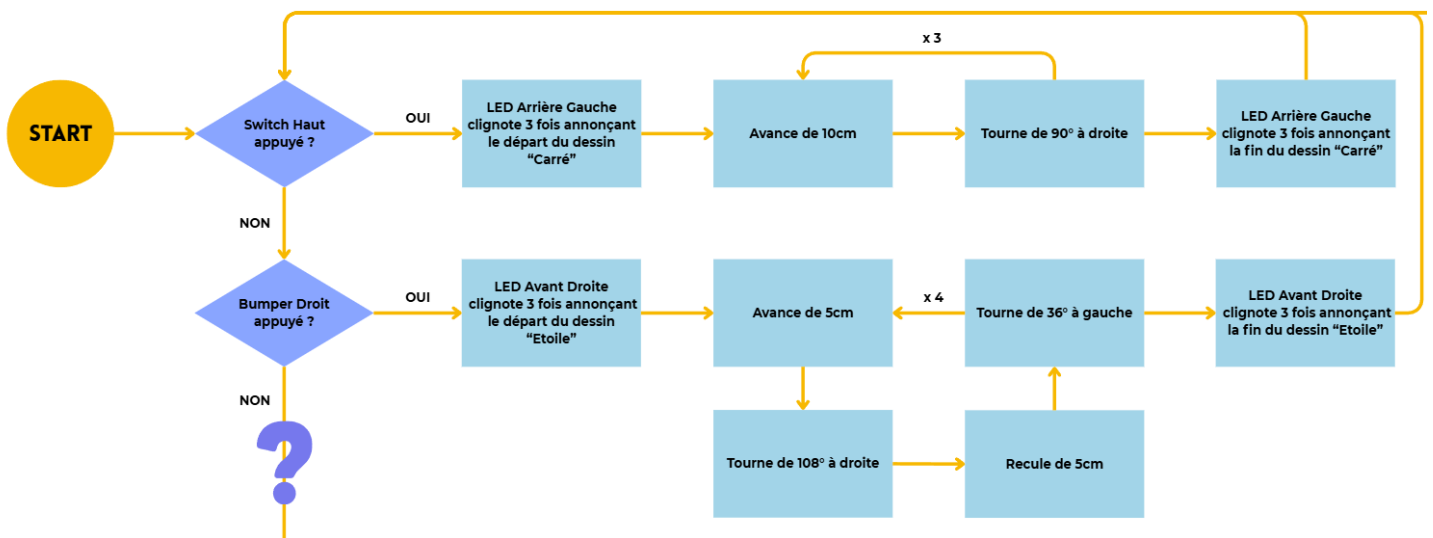


Les tracés des deux formes dites « mystères » sont présentés en annexe. Il est préférable d'avoir assisté à la démonstration du robot avant de les consulter, afin de mieux apprécier l'aspect ludique du Pictionary Bot.



Deux outils ont été utilisés pour concevoir ces formes. Le logiciel GeoGebra a permis de déterminer précisément les angles et les longueurs des segments, tandis que Scratch a servi de support pour la mise au point de l'algorithme de déplacement, qui a ensuite été transcrit en langage assembleur.

## Organigramme d'exécution



## Difficultés rencontrées

Le langage assembleur ne permet pas d'imbriquer facilement des appels de fonctions sans précautions particulières, sous peine de perdre l'adresse de retour et de provoquer des boucles infinies. Il a donc été nécessaire de sauvegarder le registre de lien (LR) dans un autre registre avant chaque appel de fonction, afin de garantir un retour correct à l'exécution principale.

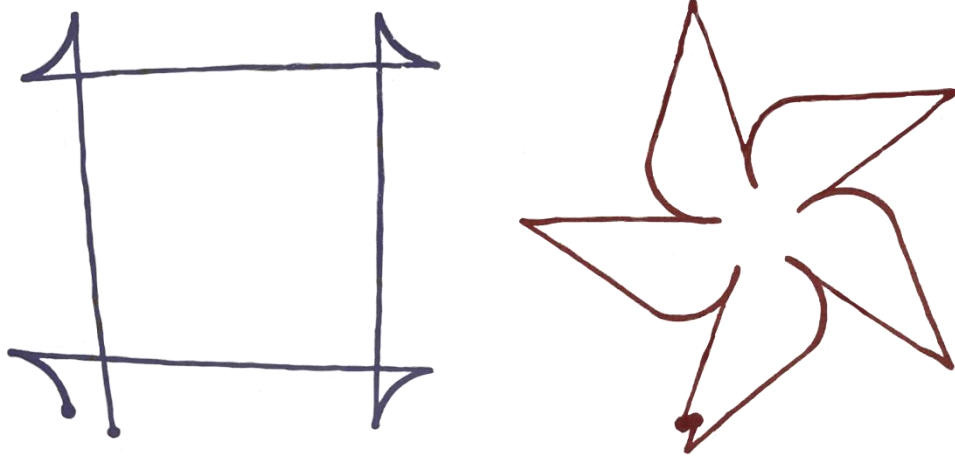
Les principales difficultés rencontrées n'ont toutefois pas été uniquement d'ordre logiciel, mais également d'ordre pratique. En théorie, les formes géométriques à tracer sont simples, mais en pratique la précision du robot est limitée. Celui-ci n'avance pas parfaitement en ligne droite, ce qui génère un décalage angulaire progressif, constituant une erreur accumulée au fil des déplacements. Cette erreur doit être compensée par des ajustements sur les rotations.

Ainsi, un angle droit utilisé pour le tracé d'un carré de grande taille ne correspond pas exactement à un angle droit dans le cas des formes plus petites et plus détaillées. De plus, la précision des angles dépend de nombreux facteurs externes, tels que la vitesse du robot, l'inclinaison du support ou encore le niveau de charge des batteries, ce qui augmente la variabilité des résultats.

Enfin, la conception et l'impression d'une pièce en 3D destinée à supporter le feutre ont constitué une difficulté supplémentaire, ce domaine étant nouveau pour nous. La précision des mesures s'est révélée particulièrement critique, ce qui nous a conduits à effectuer des ajustements et rectifications manuels afin d'obtenir un support fonctionnel.

## Résultats

Nous avons réussi à produire les quatre formes distinctes, toutes suffisamment reconnaissables malgré la marge d'erreur, que nous avons corrigée autant que possible. Les photos ci-dessous montrent les résultats obtenus.



Les vidéos du robot traçant les formes sont disponibles sur le [GitHub](#) du projet. Les photos des deux formes dites « mystères » quant à elles sont disponibles en annexe.

Les formes développées dans ce projet ont également été conçues de manière à pouvoir être combinées entre elles. Cette approche permet de générer un plus grand nombre de variantes de dessins à partir d'un ensemble limité de formes de base, augmentant ainsi la diversité et la richesse visuelle du projet. Les combinaisons de formes sont également disponibles en annexe.

## Pistes d'amélioration

Dans le cadre d'évolutions possibles de notre projet, plusieurs fonctionnalités pourraient être ajoutées afin d'enrichir l'expérience ludique :

- Un arrêt d'urgence du robot lors du tracé, déclenché par l'activation d'un bumper.
- L'affichage d'une barre de progression du dessin sur l'écran OLED.
- La diffusion d'une musique lors du tracé de chaque forme, via le haut-parleur.
- Le remplacement des valeurs temporelles arbitraires utilisées pour les rotations et les déplacements par des mesures issues de capteurs infrarouges, afin de garantir un déplacement rectiligne et des mesures précises.
- La conception d'une pièce 3D avec un tube incliné pour accueillir le feutre, réduisant ainsi la marge d'erreur lors du tracé.
- Le développement de formes plus complexes pour la difficulté et la richesse du jeu Pictionary.

Enfin, le Pictionary Bot ouvre également des perspectives d'évolution intéressantes. En particulier, il pourrait être utilisé comme un outil pédagogique permettant l'apprentissage de : formes simples pour les plus jeunes, en associant le jeu et l'interaction avec un robot.

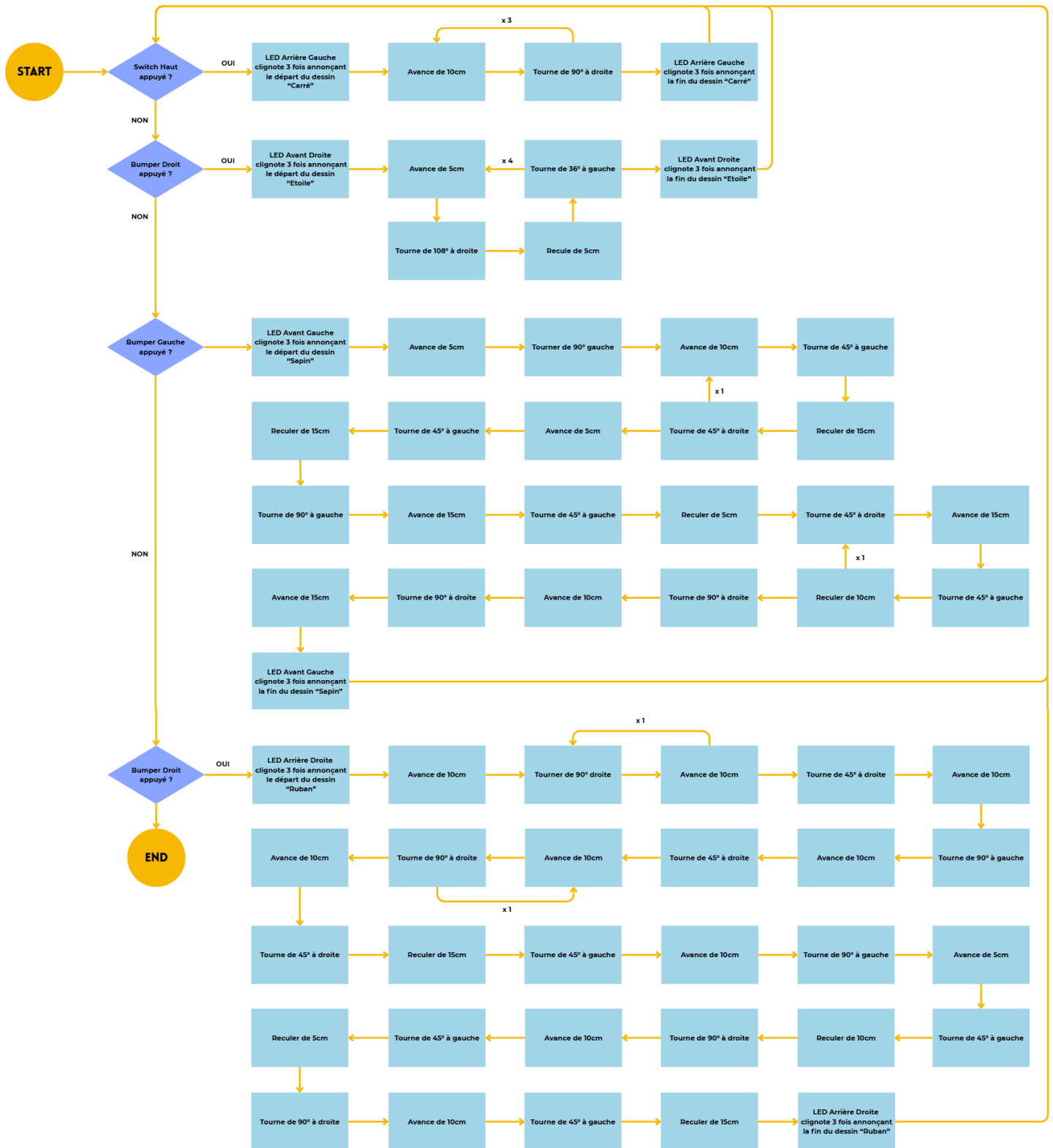
## Conclusion

Nous sommes fières du produit final et avons beaucoup apprécié la réalisation de ce projet. Il nous a permis de mettre en pratique et de consolider nos compétences en langage assembleur dans un contexte concret. Au-delà de la programmation, nous avons été confrontés à des problématiques d'ingénierie variées, telles que la conception mécanique du support du feutre et la précision des réelles.

Ce projet nous a offert l'opportunité de suivre l'intégralité d'un cycle de développement, depuis l'idée initiale et la conception, jusqu'à l'implémentation. Il nous a également permis de développer notre capacité à résoudre des problèmes pratiques et à ajuster des solutions en fonction des contraintes réelles. Enfin, l'aspect ludique et interactif du Pictionary Bot a rendu l'expérience particulièrement motivante et enrichissante, en nous confrontant à un projet à la fois technique et créatif.

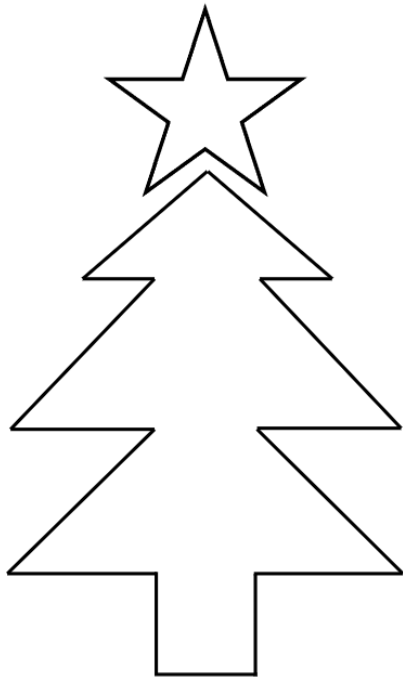


## Organigramme d'exécution complet

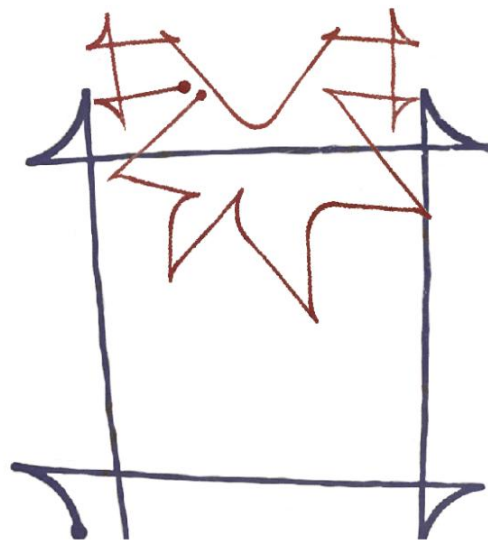
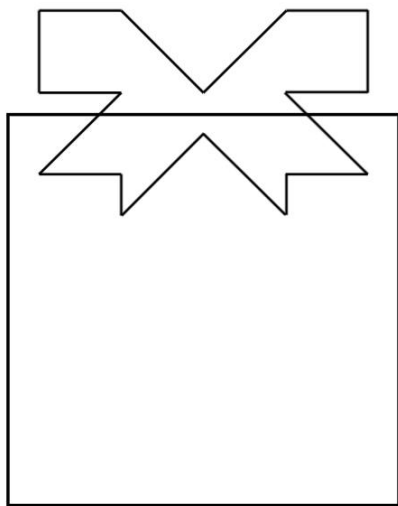


## Combinaison des formes géométriques

### SAPIN DE NOEL

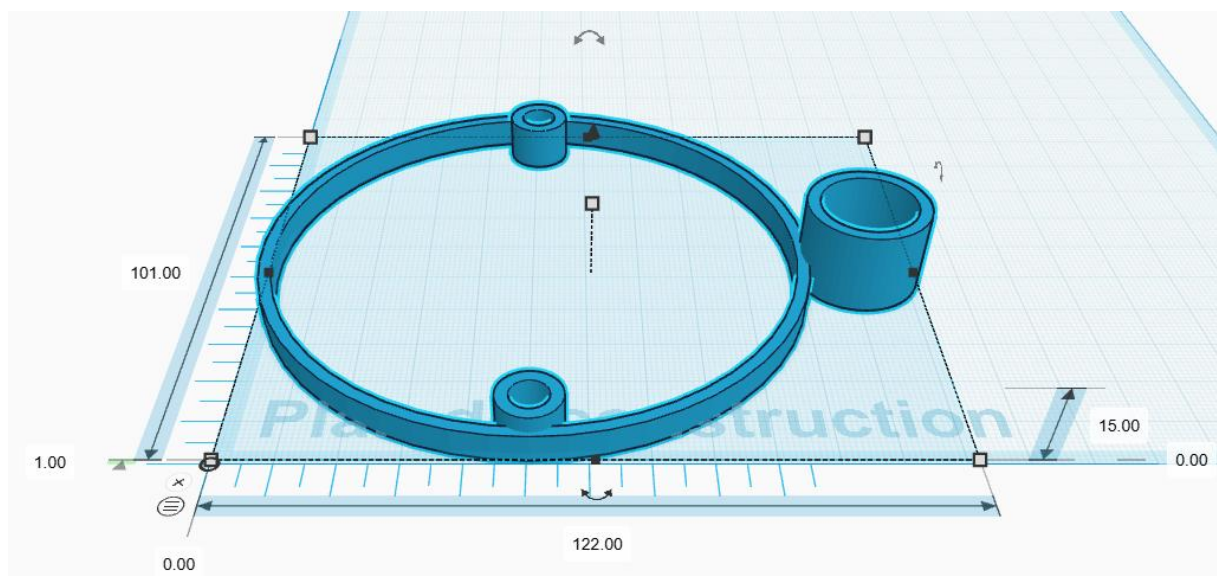
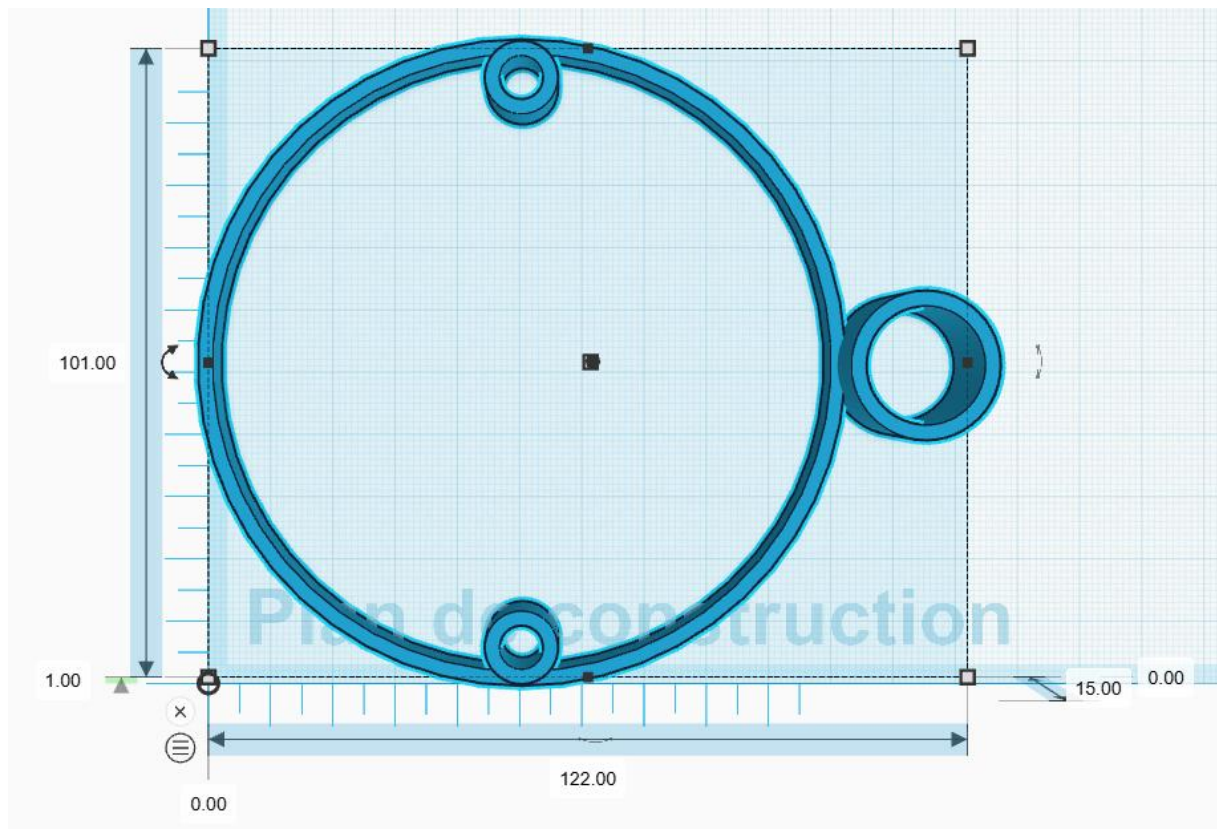


### CADEAU





## Modèle 3D du support du feutre



## Programme complet

L'intégralité du programme est disponible sur [GitHub](#), ce qui permet de faciliter la consultation du code et d'assurer un accès clair à l'ensemble des fichiers du projet.