Rapport

Projet Finite State Machine

Contrôleur de robot ménager

Nabil ABBAR

Yassine BOUKHIRI

Année universitaire :

2021-2022

Introduction

Dans le cadre de notre deuxième année du cycle d'ingénieur en informatique à polytech Nice sophia, à la fin du module de Finite State Machine, il nous a été confié un projet qui nous permet d'utiliser pleinement les connaissances apprises tout au long des séances de TD. Notre projet consiste à réaliser une statechart d'un contrôleur de robot ménager afin qu'il puisse nettoyer optimalement une pièce tout évitant les obstacles. Le projet a d'autres extensions comme le rangement des objets présents dans la scène et la gestion du crayon.

MVP :

- **Présentation de la fonctionnalité :**

Le robot se déplace dans l'environnement en évitant tous les obstacles et ne se bloque pas. Il évite de taper dans les objets. Ainsi qu'il ne franchit pas les murs virtuels qui délimitent des espaces où le robot ne doit pas aller. Pour activer cette fonctionnalité de base pour le robot, l'utilisateur choisit la deuxième option sur le menu principal.

- **Résultat obtenu :**

Comme résultat de cette fonctionnalité du robot, la terre devient plus propre comme le robot la parcourt en nettoyant chaque partie du sol qui passe par. Voici un exemple qui illustre ceci :

Sol avant nettoyage

Sol après nettoyage

- **Choix d'implémentation :**

La statechart se divise en deux majeures grandes parties : **WORKING** et **BLOCKED**. On trouve dans la première les états dans lesquels le robot fonctionne normalement. Cela regroupe la fonctionnalité de dessin et de nettoyage avec et sans rangement des objets.

On rentre à cet état initialement depuis le menu principal ou on permet à nos utilisateurs de choisir le comportement du robot.

Et puis on a l'état **BLOCKED.** Dans cet état là, le robot est devant un obstacle et doit le contourner. Dans le cadre de notre conception, on a considéré 3 types d'obstacles :

1. **Obstacles à gauche ou à droite :**

a. ***Obstacles à gauche :***

Ce type d'obstacles est détecté lorsque le capteur gauche se déclenche ou le pare-chocs gauche touche l'obstacle. Dans ce cas, le robot tourne à droite et continue le parcours.

b. ***Obstacles à droite :***

De même ces obstacles sont détectés à l'aide du capteur de face ou celui à droite ou bien le pare-chocs droit. Et pour esquiver ces obstacles, on tourne à gauche.

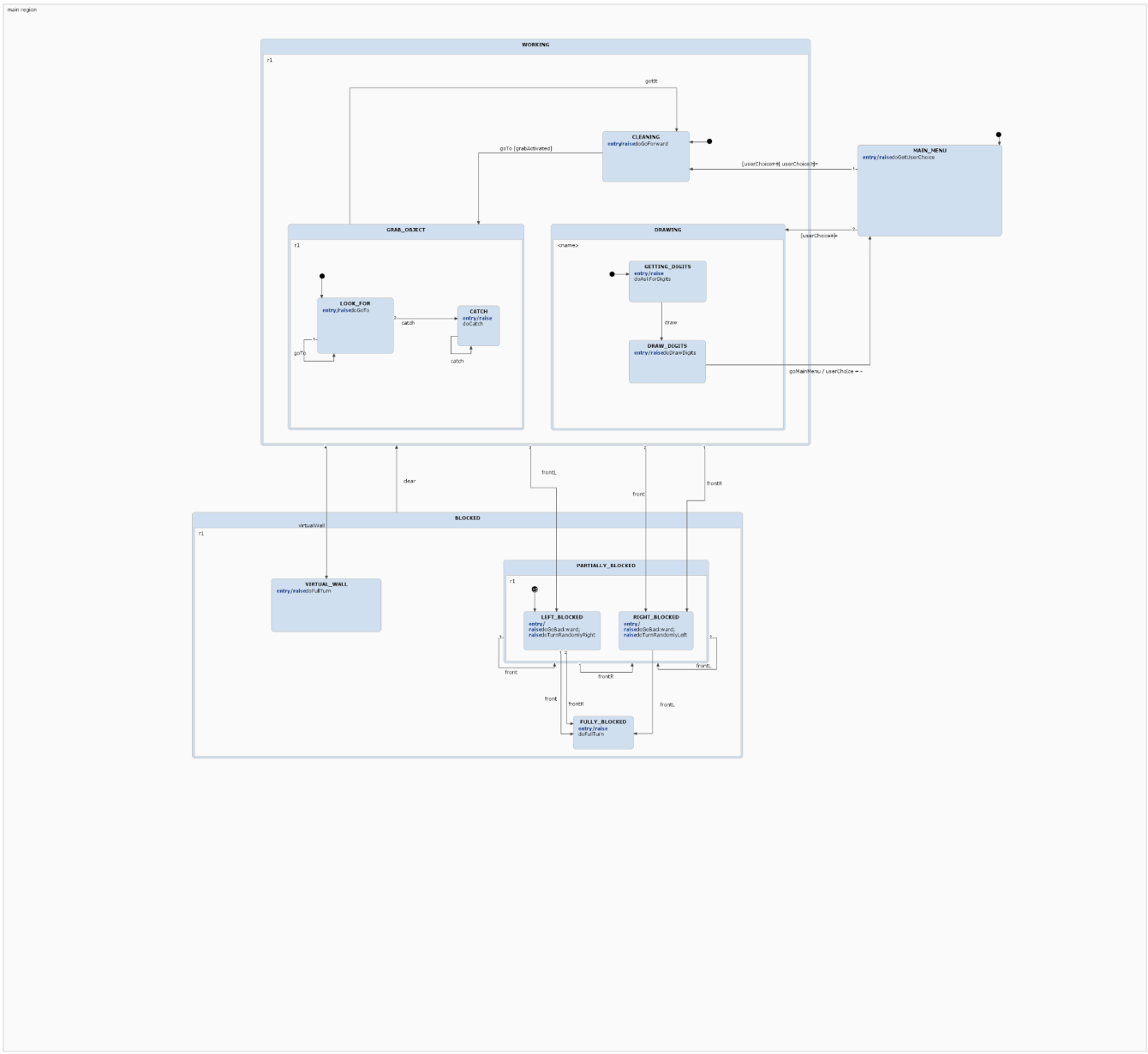
2. **Obstacles à gauche et à droite à la fois :**

Ces obstacles déclenchent tous les 3 capteurs à la fois. Ce sont généralement des obstacles très volumineux. Pour les contourner on effectue un demi-tour.

3. **Murs virtuels :**

Ces murs sont des obstacles invisibles qui sont définis par les utilisateurs à l'aide d'un appareil pour empêcher le robot de franchir quelques espaces. Face à ces derniers, notre robot effectue un demi-tour.

Finalement, lorsqu'on réussit à sortir de l'état **BLOCKED**, on revient au nettoyage normal de la scène. En passant vers la state **WORKING** une autre fois.



Statechart du projet tout entier

Rangement des objets présents dans la scène :

- **Présentation de l'extension :**

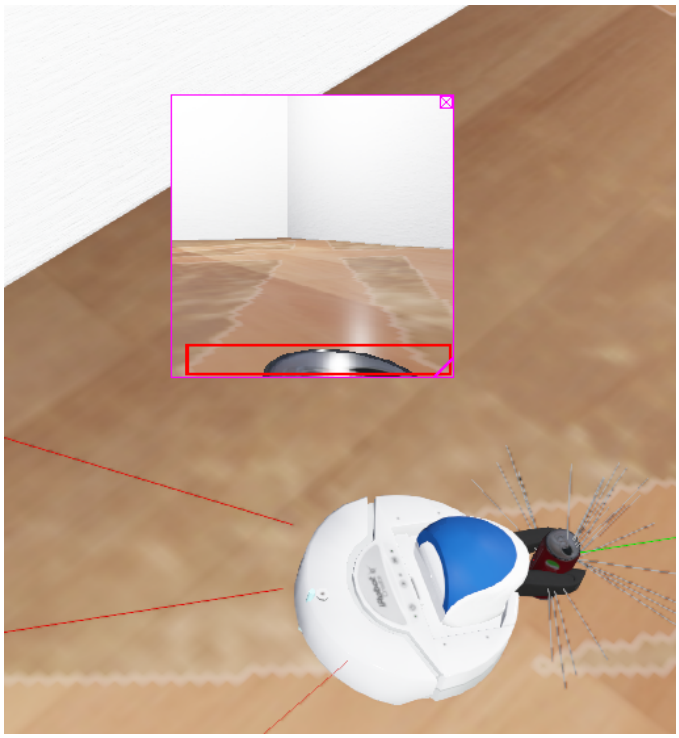
Cette extension permet au robot à l'aide de sa caméra frontale de détecter les objets présents dans la pièce, se diriger vers eux et les prendre grâce à la pince qui se trouve sur l'arrière.

On notera aussi que lors du nettoyage de la chambre, si le robot détecte des objets sur sa caméra il se dirigera vers eux en évitant les obstacles.

Cette fonctionnalité peut être activée par une entrée utilisateur et pour l'utiliser, l'utilisateur doit choisir "1" comme option sur le menu principal.

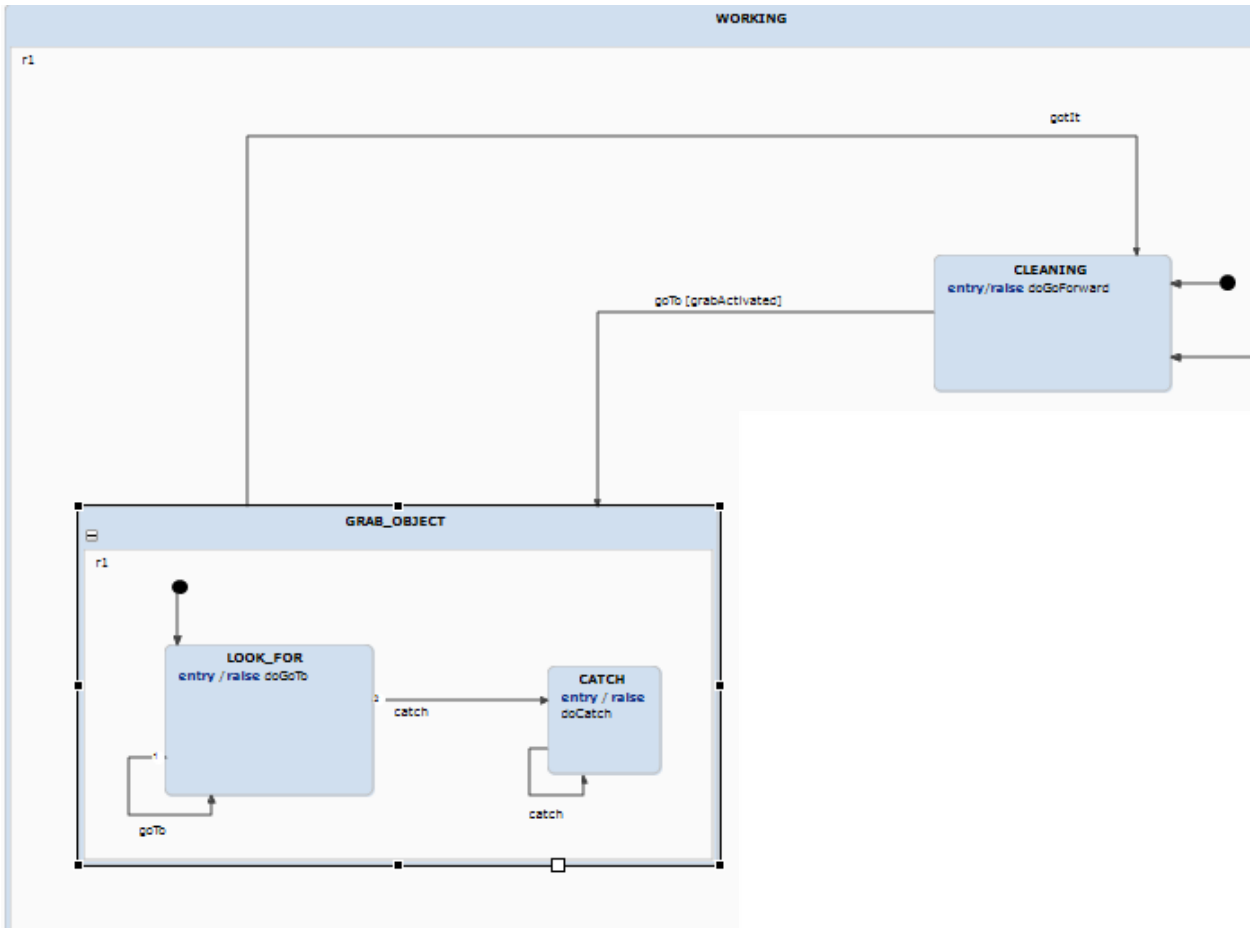
- **Résultat obtenu :**

Lorsque l'option de rangement des objets est activée, le robot pendant son parcours de la pièce prend les objets jetés par terre et continue son nettoyage. On trouve sur le schéma suivant un exemple d'objet ramassé par notre robot :



Le robot a rangé une canette

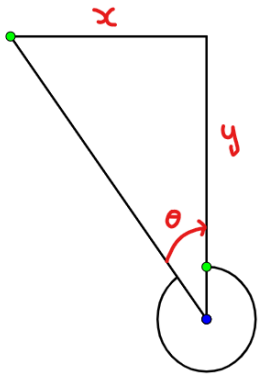
- **Choix d'implémentation :**



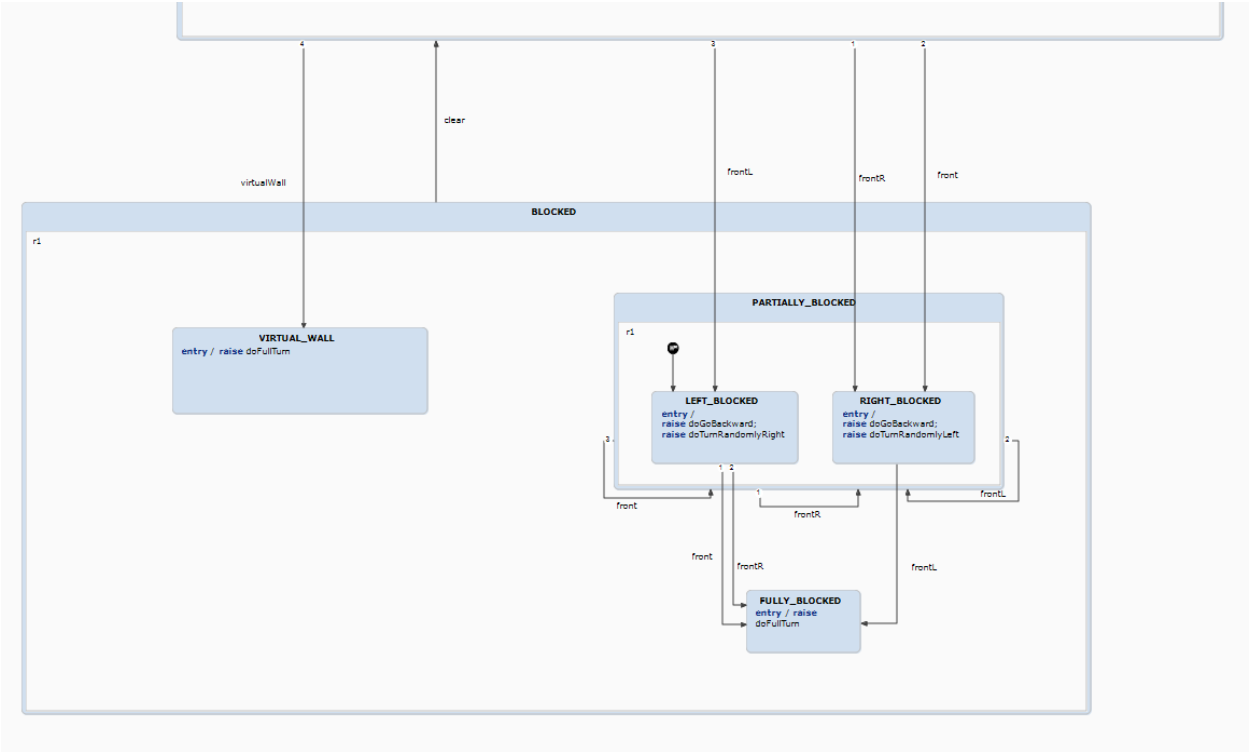
Dans cette partie du statechart (état composite **WORKING**) on a utilisé le composite state **GRAB_OBJECT** qui n'est accessible que si l'utilisateur active la fonctionnalité de rangement d'objet.Il a comme entrée l'état **LOOK_FOR** qui s'occupe de rechercher les objets et se diriger vers eux .

L'objet doit être au centre de la caméra frontale du robot, pour cela on utilise l'équation mathématique suivante : **angle = arctan(x/y)**

Et on remarque qu'on utilise une transition **goTo** qui boucle sur le même état pour pouvoir toujours recentrer l'objet sur la caméra même si l'objet rencontre un obstacle.
Sauf que cela n'est valable que si l'objet est toujours sur le champ de vision du robot.



Quand l'objet est en face du robot et a une petite distance, il fait un tour complet pour positionner la pince en face de l'objet et on passe ainsi à l'état **CATCH** dans lequel le robot attrape l'objet. Et par suite revient au nettoyage normal.



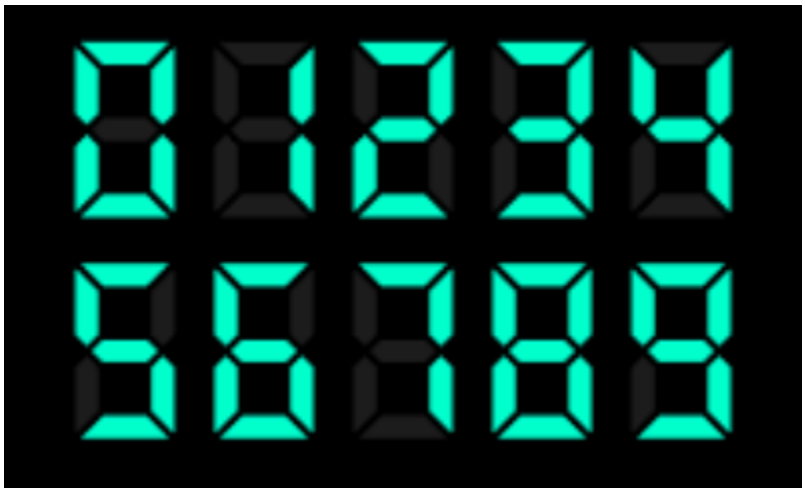
L'état composite **WORKING** est lié à cet état composite **BLOCKED** afin de permettre au robot d'éviter les obstacles et le mur virtuel en rangeant les objets.

Gestion du crayon :

- **Présentation de l’extension :**

Avec l’ajout de cette extension, notre robot peut dessiner des chiffres choisis par l’utilisateur sur la terre. Ce dernier ne peut entrer que 4 chiffres à dessiner pour le robot. Le dessin se fait en suivant une police numérique, présentée dans la figure suivante, car la nature du robot ne permet pas de dessiner des formes assez complexes.

Pour utiliser cette fonctionnalité, l'utilisateur doit choisir 3 comme option sur le menu principal, puis il rentre 1 à 4 chiffres au robot qui prendra la responsabilité de les dessiner sur terre.

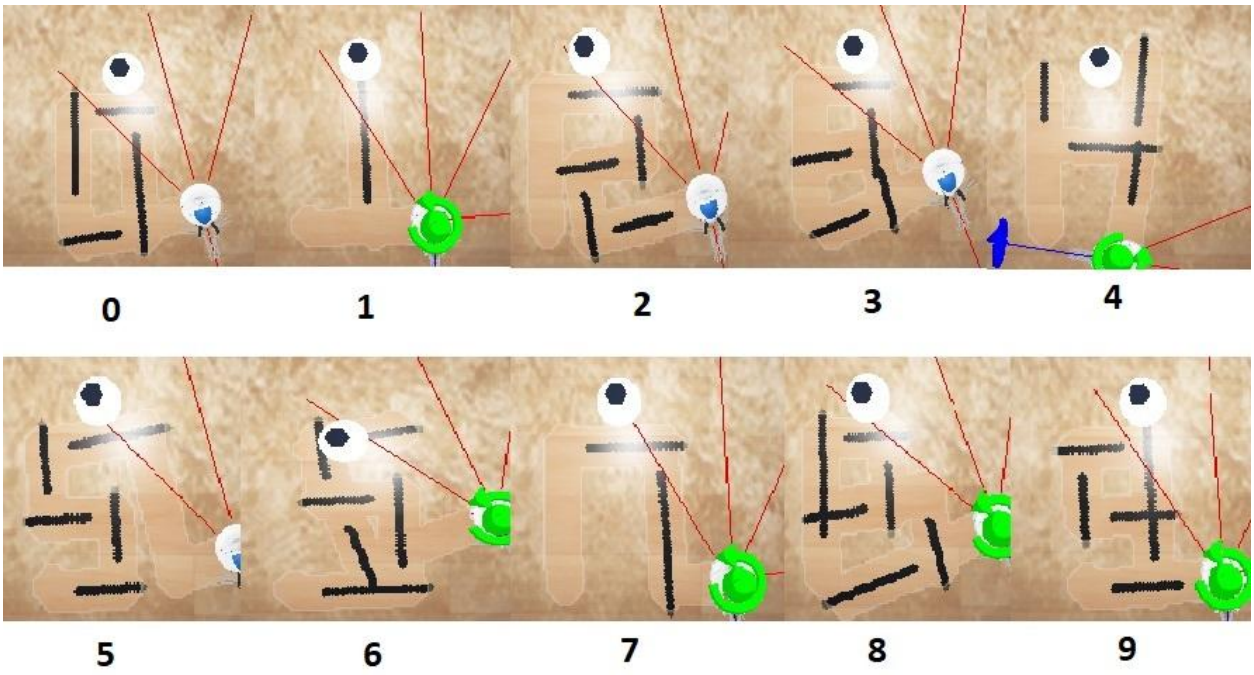


Police numérique

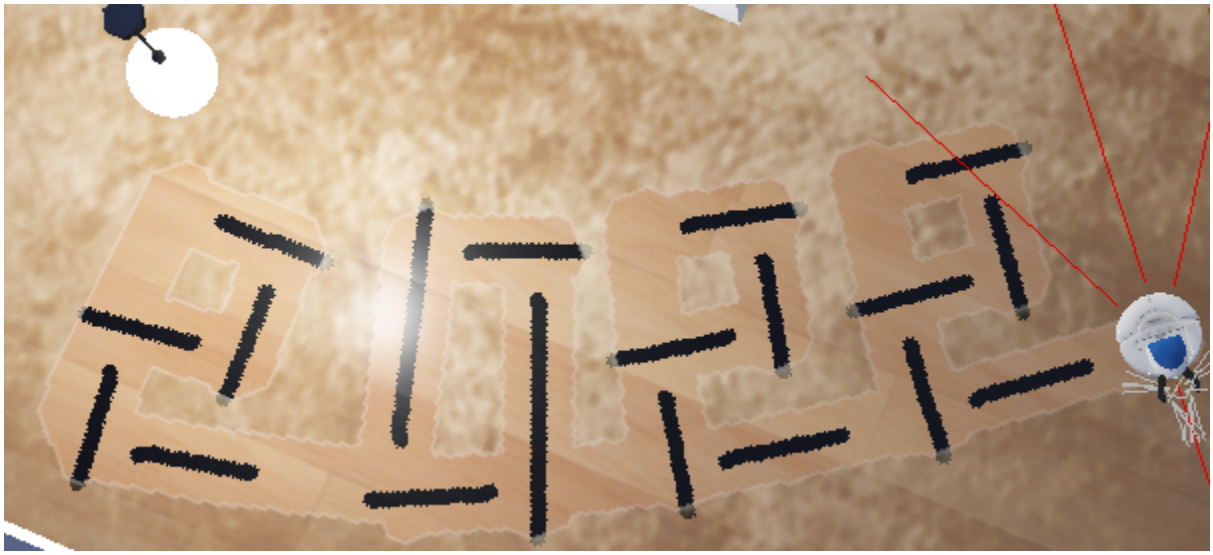
- **Résultat obtenu :**

Comme le résultat obtenu était satisfaisant pour la plupart des chiffres mais ceux qui ont une forme plus complexe et qui donc nécessite plus une procédure plus longue de dessin se tracent moins bien sur terre.

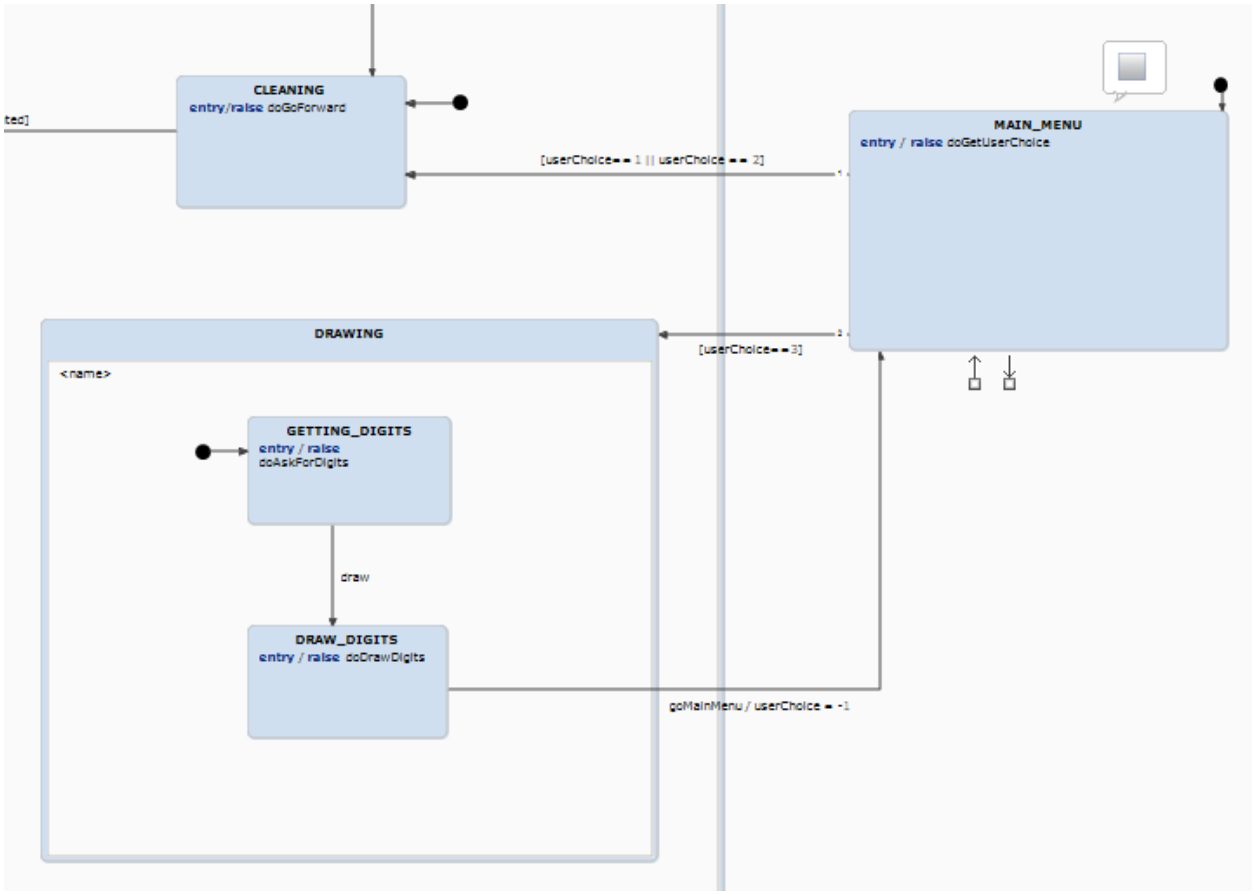
Voici donc le résultat par chiffre :



Et le résultat pour un ensemble de chiffre (2022 dans ce cas) :



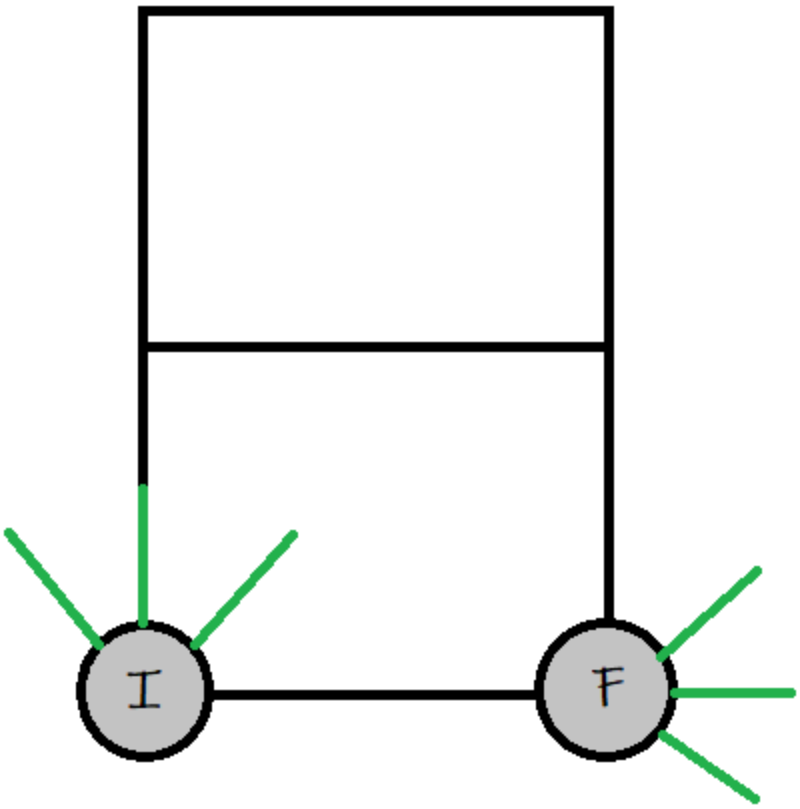
- **Choix d’implémentation :**



Pour commencer, le dessin fait aussi partie du composite state **WORKING** et par suite il est arrêté d’un la rencontre d’un obstacle.

Et afin d’implémenter cette fonctionnalité, on a décidé de la diviser sous forme de deux states. La première pour l'interaction avec l’utilisateur et donc lui demander les chiffres à tracer sur terre et la deuxième pour les dessiner. La logique du dessin se fait dans le code.

Tout d’abord, on dessine chaque chiffre en commençant toujours par le même point et on finit toujours sur le même point en dessinant pour chaque chiffre particulier ses arêtes. Voici un schéma qui explique graphiquement ce qu’on vient de mentionner :



Et bien sûr on met des espaces entre les chiffres s'il y'en a plusieurs.

Et lorsque le robot finit l'écriture d'un mot, on revient au menu principal et on demande à l'utilisateur une autre fois de choisir comment il veut utiliser son robot.