**Rapport**

Intro :

Le projet qu’ils nous a «été demandé de faire, est composé de deux partie. Dans la première partie, il était question de créer une simulation informatique d’un robot capable de se mouvoir dans une espace. Après avoir mis en place la simulation il fallait donc donner un « comportement » au robot afin qu’il réalise des déplacements et tâches précise. La deuxième partie devait consister à essayer les comportements créés lors de la simulation mais sur un vrai robot. Malheureusement, due à l’épidémie du COVID19, la deuxième partie n’a pas pu être entamée. Dans ce rapport nous allons dans un premier temps expliquer les démarches que nous avons dû suivre afin de réaliser le projet et dans un deuxième temps nous expliquerons la composition du code final.

N’ayant aucune expérience dans la réalisation de projet informatique, nous avons dû apprendre les méthodes de travails à adopter ainsi que les outils nécessaires. Nous avons utilisé la méthode agile. Ainsi, nous devions, chaque semaine, nous mettre d’accord sur un objectif à atteindre pour la semaine d’après, par rapport au temps que l’on pouvait consacrer. Les premières semaines furent un peu compliquées car nous ne savions pas comment organiser le code et nous ne savions pas quelle bibliothèque graphique utiliser. Cela nous a pris un peu plus de temps que prévu mais nous avons réussi au bout de deux trois semaines à prendre un rythme de croisière afin de respecter les objectifs fixés. Grâce à l’utilisation d’une liste de tâche (Trello pour notre cas), nous pouvions tous participer et nous savions ce qu’il restait à faire à tous moment de la semaine. L’utilisation de GitHub nous a aussi été d’une grande aide, pour partager les codes des différents membres du groupe, malgré les quelques jours nécessaire pour maîtriser l’outil. Lors de la présentation du travail effectué aux professeurs, nous avons pu récolter des conseils de ces derniers afin d’améliorer et mieux organiser notre code. Globalement, notre code était toujours dans la bonne voie, il fallait qu’il soit plus modulable pour être parfait.

Nous allons maintenant expliquer le code final. Il est composé, d’un main, d’un fichier de constantes, de 5 classes, d’un module géométrie et de fichier pour les terrains.

Le module de géométrie est utilisé principalement pour gérer les coordonnées du robot dans le terrain sur lequel, il se déplace. Le robot est représenté par une classe. Il possède des cordonnées x et y, représenté par un point (qui fait partie du module expliqué précédemment). Le robot possède également des coordonnées de vitesse, représenté par un vecteur. Le robot est capable de changer sa vitesse selon le paramètre passé. Il est également capable de tourner d’un angle donné en paramètre. On peut également récupérer les données de position et de vitesse du robot. La deuxième classe est celle du contrôleur. Le contrôleur permet de donner des ordres au robot. C’est par cette classe qu’on donne des ordres de changement de vitesse ou de rotation au robot. Et grâce à un enchaînement de ses ordres, le contrôleur est capable de faire faire un carré au robot. La troisième classe est celle du capteur. Cette classe permet d’analyser l’environnement qui entoure le robot. Le capteur est capable de donner la distance du prochain obstacle sur la trajectoire du robot. La quatrième classe est l’arène. C’est l’environnement dans lequel se déplace le robot. Cette arène est enfaite constitué de 0 pour les espaces vide ou de 1 pour les obstacles. Pour créer l’arène, il faut donc lire un fichier créé auparavant (terrain1 et terrain2) et stocker dans un tableau les données. Au milieu du projet, nous avons décidé d’augmenter la précision de chaque case afin de mieux gérer le déplacement du robot. Ainsi, une case était divisée en 30 petites cases. Et enfin il y a la classe affichage qui permet d’afficher à l’écran le robot et son environnement. Nous avons utilisé la bibliothèque graphique Pygame. Le fichier constantes permet de stocker tous les paramètres de la simulation telle que l’image du robot ou encore la distance d’arrêt. Et enfin il y a le Main qui permet de faire tourner la simulation en utilisant tous les fichiers cités précédemment.