# Etude de faisabilité IHM

Pour le besoin du projet nous avons besoin d’un périphérique permettant de contrôler à distance le déplacement du Robot en mode manuel.

Au vu du cahier des charges, ce contrôle doit pouvoir agir précisément avec un temps de réaction court (bonne réactivité). Le déplacement demandé sera sur une surface plane, soit un déplacement en 2D. Le périphérique aura donc besoin de gérer la marche avant/arrière et la rotation du robot pour permettre une complète mobilité.

Connaissant les besoins du système de pilotage, il faut maintenant tester différentes solutions envisagées afin d’avoir l’interface le plus adapté. Nous avons bien entendu pensé au clavier de l’ordinateur mais aussi à l’utilisation d’une manette. L’étude de faisabilité sera faite sur la manette. Elle sera faite en deux parties :

* Vérifier si l’utilisation d’une manette est belle est bien avantageux par rapport au clavier.
* Vérifier si une communication entre l’ordinateur et la manette peut être possible.

1. **Comparaison des attribues entre les deux solutions envisagées.**

Une étude sur internet nous a permis de dresser un tableau comparatif entre les deux interfaces (Tableau 1) :

|  |  |  |
| --- | --- | --- |
|  | Manette | Clavier |
| Ergonomie | Haute | Faible |
| Coût | 30€ | Inclus dans le pc |
| Précision | Meilleur précision dans le choix de l’angle | Angle limité par la combinaison des 4 flèches. |
| Réactivité | Instantanée | Instantanée |
| Contrôles | 2 Joysticks + 2 gâchettes + 1 pad directionnel + 11 boutons  (En fonction du type de manette) | 120 touches |

Tableau 1 Etude comparatives entre le clavier et la manette pour une interface IHM

Au vue de la précision et de l’Ergonomie, La manette semble meilleure. En effet elle permettra un meilleur contrôle du robot grâce à ses joysticks. Le seul problème est que la manette comporte moins de variable qu’un clavier. Remis dans son contexte, cela n’est pas gênant. Le nombre de variable gérée par une manette est suffisant pour une mobilité en 2D. Ainsi, le choix d’une manette comme interface est une bonne alternative au clavier.

1. **Vérifier si une communication entre l’ordinateur et la manette est possible.**

Afin d’éviter un souci de batterie et de connectivité, nous avons choisis de prendre une manette avec une connexion filaire. La marque a été faite sur une manette Xbox360 qui a l’avantage d’avoir des libraires disponibles gratuitement pour le développement.

Une fois la manette choisi, il fallait prendre des décisions sur son emploi.

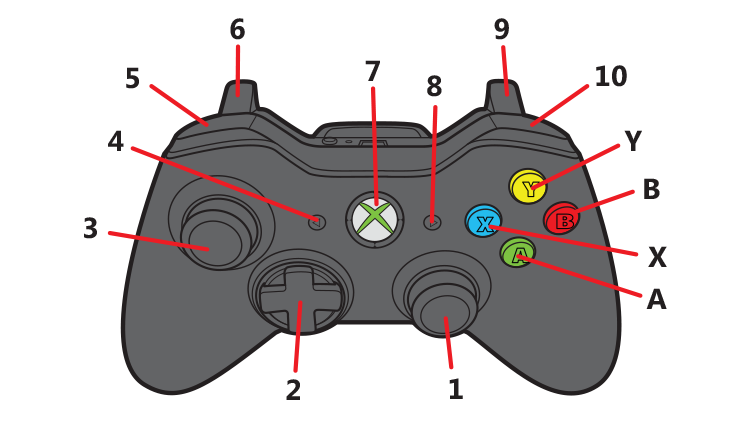


Figure 1 Les différents contrôles exploitables sur une manette Xbox360

Les décisions sont relatives et peuvent être discuté. Nous avons choisis d’utiliser le Joystick gauche (Figure 1 [3]) pour commander l’orientation du déplacement. La gâchette gauche (Figure 1 [6]) pour la marche arrière, la gâchette droite (Figure 1 [9]) pour la marche avant. Le bouton A (Figure 1 [A]) pour effectuer l’action de déposer un objet. Les gâchettes retournant une valeur compris entre [0 et 1] permettent de gérant la vitesse allant de 0 à 100% de la vitesse du robot.

Pour effectuer la communication entre la manette et l’ordinateur, il suffit simplement de brancher la manette sur l’ordinateur en USB, ensuite le système d’exploitation Windows détecte automatiquement la manette. Ensuite pour récupérer les variables dans notre application, nous disposons d’une classe Listener XboxCtrlListener qui récupère tous les évènements.

Il n’y a donc aucun problème à récupérer chaque évènements de la manette, et pas de limitation dans notre cas.

# Différents test effectués

1. Récupération d’un squelette d’application fournis par les M2-Informatique, première exécution non fonctionnelle car il manque la librairie XboxController
2. Ajout de la connexion à la manette et affiche d’un message d’erreur si aucune manette n’est détecté.
3. Rajout des évènements, test pour récupérer les valeurs des boutons, l’application ne se lance pas car des fichiers .dll sont manquants, il a fallu donc rajouter les DLL xboxcontroller.dll et xboxcontroller64.dll dans le dossier Manette/lib du projet.
4. Récupération de la valeur du Joystick [3] sous la forme de cordonnées polaires.

Sens horaire de 0 à 360°

1. Lors d’un test sur la maquette, le contrôle d’un moteur était erroné, le câble étant défectueux le moteur fonctionnait à 100% (problème pas test)