

Axel LAYEB

Virgile CHABRAN

# Rapport Robot LEGO

Sed et lacus quis enim mattis nonummy

conception du logiciel interne d'un robot qui se déplace et qui tirer  
il faut donc développer le programme qui gère les entrées utilisateur, prépare la boisson,  
et nettoie la machine.

Le robot peut effectuer différentes actions

Se déplacer dans l'espace de manière omnidirectionnel

Tirer des projectiles.

Orienter la caméra (hauteur)

Recharger ?

Se mettre en mode automatique.

Notifier le contrôleur ( état des munitions, detection de cible)

A partir des instructions

Orientation du telephone

Ordre d'avancer/reculer

Ordre de tirer

Ordre d'automatisme

Les composants du programme sont donc :

séquence d'allumage / extinction

Attente de direction.

Déplacement

interface utilisateur

le modèle de données pour ce type de projet est évident; chaque acti

Structure:

```
{  
    float MoteurGauche;  
    float MoteurDroite;  
    float MoteurHauteur;  
    bool tirer;  
    bool auto;  
}
```

Le robot nécessite de communiquer avec un autre appareil (telephone), une solution idéale serait donc de flasher la brique de commande LEGO en Java et d'y brancher un telephone android pour communiquer via les serveur en réseau.

Construire un robot fonctionnel:

- établir une mécanique de déplacement/rotation.
- établir un system de visée/camera.
- établir un system de tir/rechargement.
- mise en place des capteurs.

Pseudocode:

```
struct newinstruction
{
    float MoteurGauche;
    float MoteurDroite;
    float MoteurHauteur;
    bool tirer;
    bool auto;
}

function autoturret();

function tirer();

function movmentRL(float gauche, float droite)
{
    ActionMotorRight = droite;
    ActionMotorLeft = gauche;
}

function movmentH(float haut)
{
    ActionMotorHight= haut;
}

auto = off;
if(1){
    if(auto = on)
    {
        struct new_instruction = readinstructions();
        auto = new_instruction.auto;
        autoturret();
    }
    if(auto = off)
    {
        struct new_instruction = readinstructions();
        auto = new_instruction.auto;
        movmentRL(new_instruction.MoteurGauche,
new_instruction.MoteurDroite);
        movmentH(new_instruction.MoteurHauteur);
        tir(new_instruction.tirer);
    }
}
```

la fonction d'affichage de message peut donc être considérée comme un ingrédient qui ne communique pas avec le microcontrôleur.

deux fonctionnalités supplémentaires utiles :

Traqué les mouvement/objet avec le capteur infra-rouge

Déplacements sur terrain obstrué

l'ihm est très simple, boutons-jauge pour avancer/reculer, un bouton on/off du mode automatique et un bouton pour faire feu.