

Universidad EAFIT
Departamento de informática y sistemas
Ingeniería del Conocimiento

Trabajo Final
Pelea de Sumos

Desarrollado por:
Rubén Darío Espinosa Roldán
Juan Sebastian Velez Posada

Medellín - Colombia
Viernes, 8 de Junio del 2012

Indice

Indice
Objetivos
Específicos

Ingeniería del Conocimiento

Introducción

Actividades del proyecto

Diseño del robot

Elaboración del robot

Desarrollo del código

Puesta en marcha

Etapas de diseño (imagenes)

V1

V2

V2.1

V3

V3

Codigo Fuente

Clase Principal

Clase Mover

Conclusiones

Objetivos

Aplicar los conocimientos adquiridos en la asignatura Ingeniería del Conocimiento, en la cual se trataron temas acerca de construir maquinas inteligentes, para esto es necesario entender los conceptos de inteligencia artificial, agentes inteligentes, agentes ideales, etc. Esto con el fin de que estos agentes nos ayuden a solucionar problemas específicos en un ambiente dado.

Específicos

- Construir un robot con el fin de simular una pelea de sumos.
- Utilizar los LEGO MINDSTORMS para construir los robots.
- Desarrollar un programa en JAVA, el cual debe correr en el robot, el código debe cumplir con el objetivo de sacar los oponentes.
- Planear y monitorear el plan de trabajo del equipo.

Introducción

En el siguiente escrito explicaremos el proceso realizado para el trabajo final en la materia ingeniería del conocimiento, en específico hablaremos del tema inteligencia artificial el cual tomamos como base de la realización del trabajo o práctica correspondiente a la competencia de lego mindstorm, para el cual construimos un robot armado con piezas de lego y programado con el firmware lejos y en el IDE netbeans con el plugin de Lejos NXT.

¿Qué es LEGO MINDSTORMS? Con LEGO MINDSTORMS usted puede construir y programar robots que hacen lo que quieras! Con el contenido del kit obtiene todo lo necesario para construir y programar su propio robot inteligente de LEGO, y que lleve a cabo un montón de diferentes operaciones. El robot puede estar equipado, por ejemplo, con sensores que controlan los motores y reaccionan a la luz, sonido, tacto, etc¹

¹ "LEGO.com MINDSTORMS : What is in the box?." 2009. 5 Jun. 2012 <<http://mindstorms.lego.com/en-gb/history/default.aspx>>

Actividades del proyecto

Diseño del robot

Encargados: Juan Sebastian Velez Posada, Ruben Dario Espinosa Roldan

Descripción: Búsqueda en Internet de posibles diseño para el robot, buscar en la documentación de legos y ver video en youtube de pelea de sumos con legos con el fin de encontrar diseños que se puedan adaptar a las reglas de la competencia de nosotros.

Tiempo Estimado: 2 horas

Tiempo Real: 1 hora 40 minutos

Elaboración del robot

Encargados: Juan Sebastian Velez Posada, Ruben Dario Espinosa Roldan

Descripción: Utilización de las piezas del lego para armar el robot, es necesario tener claro el diseño que se va a utilizar, tener en cuenta las reglas del juego para no tener que hacer cambios de último momento.

Tiempo Estimado: 3 horas

Tiempo Real: 10 horas

Desarrollo del código

Encargados: Juan Sebastian Velez Posada, Ruben Dario Espinosa Roldan

Descripción: Desarrollo del código en JAVA, utilizando las librerías de LejosNXT, las cuales permiten la programación de los robots con este lenguaje.

Tiempo Estimado: 3 horas

Tiempo Real: 6 horas

Puesta en marcha

Encargados: Juan Sebastian Velez Posada

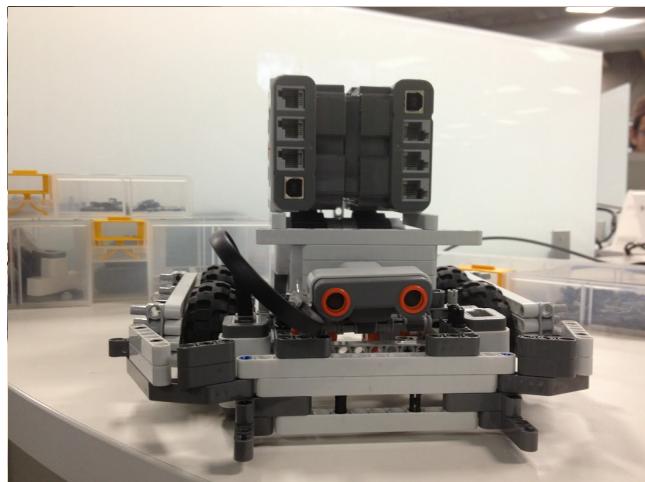
Descripción: Encargado de poner el robot en el doyho y encender el robot, en caso de algún problema en la “pelea” es el encargado de recoger el robot y ponerlo nuevamente en funcionamiento

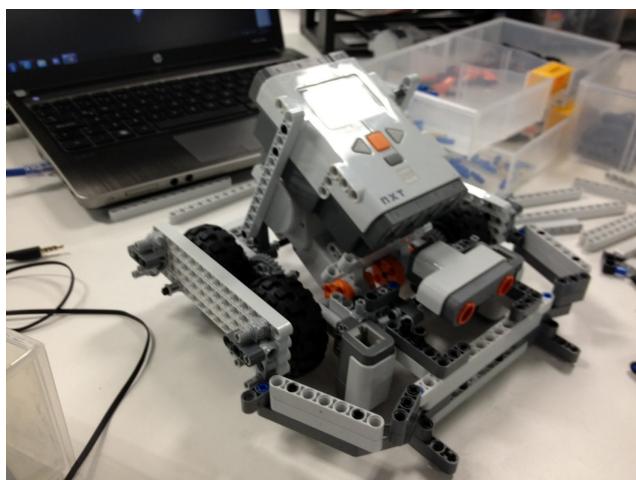
Tiempo Estimado: El tiempo que dure la competencia

Etapas de diseño (imagenes)

V1

V2

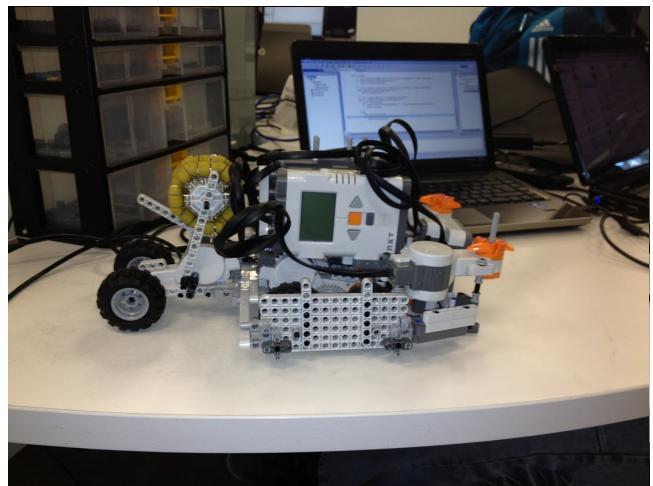




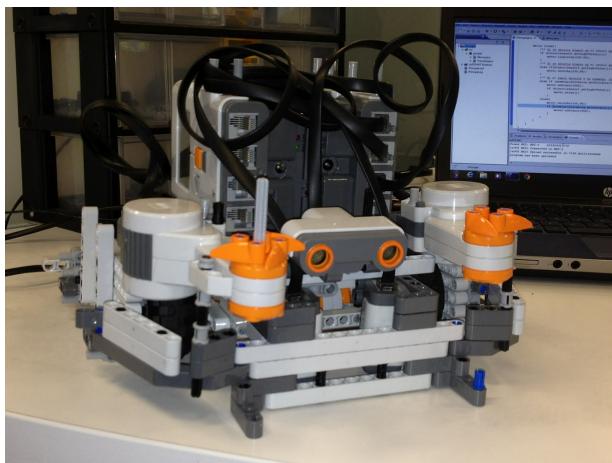
V2.1



V.3



V3



Código Fuente

Clase Principal

```
package principal;

import lejos.nxt.*;

public class Principal {

    /**
     * Retorna verdadero si el color es blanco
     * @param numero, el valor leido por el sensor de color
     * @return
     */
    static boolean blanco(int numero){
        boolean resultado;
        if (numero > 43){
            resultado = true;
        }else{
            resultado = false;
        }
        return resultado;
    }
    static boolean enemigo(int numero){
        boolean resultado;
        if (numero < 30){
            resultado = true;
        }else {
            resultado = false;
        }
        return resultado;
    }
    static void iniciar(){
    }
    /**
     * Inicio de la aplicación 'El Santo'
     * @param args
     */
    public static void main(String[] args){
        Mover mover = new Mover();
        mover.derecha(500, 600);
        /** Sensor de distancia */
        UltrasonicSensor distancia = new UltrasonicSensor(SensorPort.S3);
        /** Sensor de luz de la izquierda */
        LightSensor sensor1 = new LightSensor(SensorPort.S1);
        /** Sensor de luz de la derecha */
        LightSensor sensor2 = new LightSensor(SensorPort.S2);

        while (true){
            /** Si se detecta blanco en el sensor de la izquierda, el santo gira */
            if (blanco(sensor1.getLightValue())){
                mover.derecha(250,300);
            }
            /** Si se detecta blanco en el sensor de la derecha, el santo gira */
            else if(blanco(sensor2.getLightValue())){
                mover.izquierda(250,300);
            }
            /** Si el santo detecta a su enemigo, lo ataca */
            else if (enemigo(distancia.getDistance())){
                mover.adelante(1000);
                /*
                if (!enemigo(distancia.getDistance())){
                    mover.derecha(1000);
                }
                else if (enemigo(distancia.getDistance())){
                    mover.izquierda(1000);
                }
            }
        }
    }
}
```

```
        mover.atras();
        Delay.msDelay(1000);
    }
    if (blanco(sensor1.getLightValue()) || blanco(sensor2.getLightValue())){
        mover.atras();
    }
    */
}
/** Si el santo no encuentra ningun enemigo, lo busca */
else{
    mover.derecha(250,90);
    //mover.izquierda(250, 90);
    if (enemigo(distancia.getDistance())){
        mover.adelante(1000);
    }
}
}
```

Clase Mover

```
package principal;

import lejos.nxt.*;

public class Mover {
    /** Motor atras */
    NXTRegulatedMotor MotorA = Motor.A;
    /** Motor izquierda */
    NXTRegulatedMotor MotorB = Motor.B;
    /** Motor derecha */
    NXTRegulatedMotor MotorC = Motor.C;

    /**
     * Clase que contiene todos los metodos para mover al santo
     */
    public Mover(){
    }

    /**
     * Mover el santo hacia adelante
     * @param speed, Velocidad con la que se va a mover el santo
     */
    void adelante(int speed){
        MotorA.setSpeed(speed);
        MotorB.setSpeed(speed);
        MotorC.setSpeed(speed);

        MotorA.backward();
        MotorB.forward();
        MotorC.forward();
    }

    /*
     * Mover el santo hacia atras
     */
    void atras(){
        MotorA.stop();
        MotorB.setSpeed(700);
        MotorC.setSpeed(700);

        MotorB.backward();
        MotorC.backward();
    }

    /**
     * Mover el santo hacia la derecha
     */
    void derecha(int speed, int grados){
        MotorA.stop();
        MotorB.setSpeed(speed);
        MotorC.setSpeed(speed);

        MotorC.rotate(grados,true);
        MotorB.rotate(-grados);
    }

    /**
     * Mover el santo hacia la izquierda
     */
    void izquierda(int speed, int grados){
        MotorA.stop();
        MotorB.setSpeed(speed);
        MotorC.setSpeed(speed);
    }
}
```

```
        MotorB.rotate(grados,true);
        MotorC.rotate(-grados);
    }
```

Conclusiones

- Para la construcción de un robot lego mindstorm se debe de consultar en diferentes fuentes, ya que cualquier persona creería que es un juego de niños, pero la realidad es que se trata de un proceso complejo y que tiene una curva de aprendizaje algo compleja para poder construir un buen robot.
- La librería de Lejos en algunos métodos no esta muy bien documentada, por lo que se hace difícil entender cual es su funcionamiento.
- Trabajar con piezas limitadas y con poco tiempo nos llevo a forzar la imaginación para dar con un robot compacto, fuerte y bien programado con el fin de ganar la competencia de sumo.