Copyright (c) FIRST and other WPILib contributors.

// Open Source Software; you can modify and/or share it under the terms of

// the WPILib BSD license file in the root directory of this project.

package frc.robot;

import edu.wpi.first.wpilibj.DigitalInput;

import edu.wpi.first.wpilibj.GenericHID;

import edu.wpi.first.wpilibj.Joystick;

import edu.wpi.first.wpilibj.TimedRobot;

import edu.wpi.first.wpilibj.drive.DifferentialDrive;

import edu.wpi.first.wpilibj.motorcontrol.MotorController;

import edu.wpi.first.wpilibj.motorcontrol.Talon;

import edu.wpi.first.wpilibj2.command.button.JoystickButton;

/\*\*

 \* This is a demo program showing the use of the DifferentialDrive class,

 \* specifically it contains

 \* the code necessary to operate a robot with tank drive.

 \*/

public class Robot extends TimedRobot {

  private DifferentialDrive m\_myRobot;

  private Joystick m\_leftStick;

  private Joystick m\_rightStick;

  private final MotorController m\_leftMotor = new Talon(0);

  private final MotorController m\_rightMotor = new Talon(1);

  private final MotorController m\_Lift = new Talon(2);

  private final MotorController m\_Ext = new Talon(3);

  private final MotorController m\_claw = new Talon(8);

  DigitalInput toplimitSwitch = new DigitalInput(0);

  DigitalInput bottomlimitSwitch = new DigitalInput(1);

  DigitalInput toplimitSwitch1 = new DigitalInput(2);

  DigitalInput bottomlimitSwitch1 = new DigitalInput(3);

  DigitalInput toplimitSwitch11 = new DigitalInput(4);

  DigitalInput bottomlimitSwitch11 = new DigitalInput(5);

  double throttleA = (m\_leftStick.getRawAxis(2) + 1) / 2;

  double Lspeed = 1.0; // set lift wheel speed to max

  double Espeed = 1.0; // set extender speed to max

  long autoTimer;

  double deadzone = 0.15;

  public class OI {

    // Create the joystick and the 8 buttons on it

    // Create Joystick and Buttons

    Joystick   m\_LeftStick = new Joystick(0);

    JoystickButton Button1 = new JoystickButton(m\_leftStick, 0),

                   Button2 = new JoystickButton(m\_leftStick, 2),

                   Button3 = new JoystickButton(m\_leftStick, 3),

                   Button4 = new JoystickButton(m\_leftStick, 4),

                   Button5 = new JoystickButton(m\_leftStick, 5),

                   Button6 = new JoystickButton(m\_leftStick, 6),

                   Button7 = new JoystickButton(m\_leftStick, 7);

    Joystick   m\_rightStick = new Joystick(1);

    JoystickButton Button11 = new JoystickButton(m\_rightStick, 0),

                   Button22 = new JoystickButton(m\_rightStick, 2),

                   Button33 = new JoystickButton(m\_rightStick, 3),

                   Button44 = new JoystickButton(m\_rightStick, 4),

                   Button55 = new JoystickButton(m\_rightStick, 5),

                   Button66 = new JoystickButton(m\_rightStick, 6),

                   Button77 = new JoystickButton(m\_rightStick, 7);

  }

  @Override

  public void robotInit() {

    // We need to invert one side of the drivetrain so that positive voltages

    // result in both sides moving forward. Depending on how your robot's

    // gearbox is constructed, you might have to invert the left side instead.

    m\_rightMotor.setInverted(true);

    m\_leftStick = new Joystick(0);

    m\_rightStick = new Joystick(1);

    m\_myRobot = new DifferentialDrive(m\_leftMotor, m\_rightMotor);

  }

  public void autonomousInit() {

    autoTimer = System.currentTimeMillis();

  }

  public void autonomousPeriodic() {

    if (System.currentTimeMillis() < (autoTimer + 5000)) {

      m\_leftMotor.set(-0.55);

      m\_rightMotor.set(0.5);

    } else {

      m\_leftMotor.set(0.0);

      m\_rightMotor.set(0.0);

    }

  }

  @Override

  public void teleopPeriodic() {

      double lSpeed = m\_leftStick.getRawAxis(1);

      double rSpeed = -m\_rightStick.getRawAxis(1);

 if (lSpeed < deadzone && lSpeed > -deadzone)

        lSpeed = 0.0;

      if (rSpeed < deadzone && rSpeed > -deadzone)

        rSpeed = 0.0;

        m\_leftMotor .set(lSpeed \* throttleA);   // speed of robot is proportional to throttleA

        m\_rightMotor .set(rSpeed \* throttleA);

    if (Lspeed \* throttleA != 0 || Espeed \* throttleA != 0) {

      System.out.println("Lift Speed:");

      System.out.println(Lspeed \* throttleA);

      System.out.println("Ext Speed:");

      System.out.println(Espeed \* throttleA);

      PutString("DB/String 0", "This is a string");

      PutString("DB/String 1", "This is another string");

    }

   // m\_myRobot.tankDrive(-m\_leftStick.getY(), -m\_rightStick.getY());

    m\_myRobot.tankDrive(m\_leftStick.getY(), m\_rightStick.getY());

    Ext();

    Lift();

    claw();

  }

  private void PutString(String string, String string2) {

  }

  public void Lift() {

    if (m\_leftStick.getRawButton(1) == true && m\_leftStick.getRawButton(2) == !true && toplimitSwitch.get() == true) {

      m\_Lift.set(1.0);

    } else {

      if (m\_leftStick.getRawButton(2) == true && m\_leftStick.getRawButton(1) == !true && bottomlimitSwitch.get() == true) {

        m\_Lift.set(-1.0);

      } else {

        m\_Lift.set(0.0);

      }

    }

  }

  public void Ext() {

    if ((m\_rightStick.getRawButton(1) == true) && m\_rightStick.getRawButton(2) == !true

        && toplimitSwitch1.get() == true) {

      m\_Ext.set(1.0);

    } else {

      if ((m\_rightStick.getRawButton(2) == true) && m\_rightStick.getRawButton(1) == !true

          && bottomlimitSwitch1.get() == true) {

        m\_Ext.set(-1.0);

      } else {

        m\_Ext.set(0.0);

      }

    }

  }

  public void claw() {

    if ((m\_rightStick.getRawButton(3) == true) && m\_rightStick.getRawButton(4) == !true

        && toplimitSwitch11.get() == true) {

      m\_claw.set(1.0);

    } else {

      if ((m\_rightStick.getRawButton(4) == true) && m\_rightStick.getRawButton(3) == !true

          && bottomlimitSwitch11.get() == true) {

        m\_claw.set(-1.0);

      } else {

        m\_claw.set(0.0);

      }

    }

  }

}