/\*

\* Drivetrain:

\* 8 victors, 4 wheels

\* 4 CIM Motors, 4 Mini-CIMs

\* Two Joysticks, One Controller

\* Variable speed when joystick triggers are clicked

\*

\* Structure:

\* Fourbar Piston (Multi-Positional Solenoid)

\* Two smaller pistons that actuate

\*/

#include <WPILib.h>

class RobotDemo : public IterativeRobot

{

public:

Victor \*leftRear, \*leftRearCenter, \*leftFrontCenter, \*leftFront, \*rightRear, \*rightRearCenter, \*rightFrontCenter, \*rightFront;

Joystick \*leftJoy, \*rightJoy, \*controller;

Compressor \*compressor;

Relay \*bertha, \*smallRight, \*smallLeft;

Relay \*ledRing;

DigitalInput \*limitSwitch;

RobotDemo()

{

leftRear = new Victor (1);

leftRearCenter = new Victor (2); //left back wheel

leftFrontCenter = new Victor (3);

leftFront = new Victor (4); //left front wheel

rightRear = new Victor (5);

rightRearCenter = new Victor (6); //right rear wheel

rightFrontCenter = new Victor (7);

rightFront = new Victor (8); //right front wheel

leftJoy = new Joystick(1);

rightJoy = new Joystick(2);

controller = new Joystick(3);

compressor = new Compressor(1, 1);

limitSwitch = new DigitalInput(1);

}

void RobotDemo::RobotInit()

{

compressor->Start();

}

void RobotDemo::AutonomousInit()

{

}

void RobotDemo::TeleopInit()

{

}

void RobotDemo::TeleopPeriodic()

{

ArcadeShift(rightJoy->GetRawAxis(1), -rightJoy->GetRawAxis(2), rightJoy->GetRawAxis(3));

//TankShift(leftJoy->GetRawAxis(1), leftJoy->GetRawAxis(2), rightJoy->GetRawAxis(1), rightJoy->GetRawAxis(2));

if (controller->GetRawAxis(2) > 0.25)

{

bertha->Set(Relay::kReverse);

}

else if (controller->GetRawAxis(2) < -0.25)

{

bertha->Set(Relay::kForward);

}

else

{

bertha->Set(Relay::kOff);

}

if (controller->GetRawButton(1))

{

smallLeft->Set(Relay::kForward);

smallRight->Set(Relay::kForward);

}

else

{

smallLeft->Set(Relay::kReverse);

smallRight->Set(Relay::kReverse);

}

if (limitSwitch)

{

ledRing->Set(Relay::kOn);

}

else

{

ledRing->Set(Relay::kOff);

}

}

void ArcadeShift(double x, double y, double z)

{

double a = y, b = x, c = z;

double lf, lr, rf, rr; //lf = left front lr = left rear rf = right front rr = right rear

lf = a + b + c;

lr = a - b + c;

rf = a - b - c;

rr = a + b - c;

double max = 0;

if (max < abs(lf))

{

max = abs(lf);

}

if (max < abs(lr))

{

max = abs(lr);

}

if (max < abs(rf))

{

max = abs(rf);

}

if (max < abs(rr))

{

max = abs(rr);

}

if (max > 1)

{

lf = lf/max;

lr = lr/max;

rf = rf/max;

rr = rr/max;

}

leftFront->Set(lf);

leftFrontCenter->Set(lf);

leftRear->Set(lr);

leftRearCenter->Set(lr);

rightFront->Set(rf);

rightFrontCenter->Set(rf);

rightRear->Set(rr);

rightRearCenter->Set(rr);

}

void TankShift(double ly, double lx, double ry, double rx)

{

double lf, lr, rf, rr;

lf = ly + lx;

lr = ly - lx;

rf = ry - rx;

rr = ry + rx;

double max = 0;

if (max < abs(lf))

{

max = abs(lf);

}

if (max < abs(lr))

{

max = abs(lr);

}

if (max < abs(rf))

{

max = abs(rf);

}

if (max < abs(rr))

{

max = abs(rr);

}

if (max > 1)

{

lf = lf/max;

lr = lr/max;

rf = rf/max;

rr = rr/max;

}

leftFront->Set(lf);

leftFrontCenter->Set(lf);

leftRear->Set(lr);

leftRearCenter->Set(lr);

rightFront->Set(rf);

rightFrontCenter->Set(rf);

rightRear->Set(rr);

rightRearCenter->Set(rr);

}

};

START\_ROBOT\_CLASS(RobotDemo);