package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;

import com.qualcomm.robotcore.eventloop.opmode.TeleOp;

@TeleOp(name="RobotTest 2.0", group="Test")

// @Disabled

public class RobotTest extends LinearOpMode {

PsaBot2020 robot = new PsaBot2020(); // Use a Pushbot's hardware

void tankMode() {

double fl = 0.0;

double fr = 0.0;

double bl = 0.0;

double br = 0.0;

if (gamepad1.left\_stick\_y < -0.2 || 0.2 < gamepad1.left\_stick\_y) {

fl = -gamepad1.left\_stick\_y;

bl = -gamepad1.left\_stick\_y;

} else if (gamepad1.left\_stick\_x < -0.2 || 0.2 < gamepad1.left\_stick\_x) {

fl = gamepad1.left\_stick\_x;

bl = -gamepad1.left\_stick\_x;

}

if (gamepad1.right\_stick\_y < -0.2 || 0.2 < gamepad1.right\_stick\_y) {

fr = -gamepad1.right\_stick\_y;

br = -gamepad1.right\_stick\_y;

} else if (gamepad1.right\_stick\_x < -0.2 || 0.2 < gamepad1.right\_stick\_x) {

fr = gamepad1.right\_stick\_x;

br = -gamepad1.right\_stick\_x;

}

robot.setDrivePower(fl, bl, fr, br);

}

// left joystick is for straight movement

// right joystick is for left and right turn

void omniMode() {

double fb\_mult = 0.8;

double st\_mult = 0.8;

double lr\_mult = 0.8;

double fl = 0.0;

double fr = 0.0;

double bl = 0.0;

double br = 0.0;

// Left Y = Fwd / Back

fl = -gamepad1.left\_stick\_y \* fb\_mult;

bl = -gamepad1.left\_stick\_y \* fb\_mult;

fr = -gamepad1.left\_stick\_y \* fb\_mult;

br = -gamepad1.left\_stick\_y \* fb\_mult;

// Left X = Strafe L/R

fl += gamepad1.left\_stick\_x \* st\_mult;

br += gamepad1.left\_stick\_x \* st\_mult;

fr += -gamepad1.left\_stick\_x \* st\_mult;

bl += -gamepad1.left\_stick\_x \* st\_mult;

// left and right turn

fl += gamepad1.right\_stick\_x \* lr\_mult;

bl += gamepad1.right\_stick\_x \* lr\_mult;

fr += -gamepad1.right\_stick\_x \* lr\_mult;

br += -gamepad1.right\_stick\_x \* lr\_mult;

double mx = Math.max(Math.max(Math.abs(fl), Math.abs(fr)), Math.max(Math.abs(bl), Math.abs(br)));

if (mx < 0.2) {

fl = fr = bl = br = 0.0;

} else if (mx > 1.0) {

fl \*= 1.0 / mx;

fr \*= 1.0 / mx;

bl \*= 1.0 / mx;

br \*= 1.0 / mx;

}

robot.setDrivePower(fl, bl, fr, br);

}

@Override

public void runOpMode() {

/\*

\* Initialize the drive system variables.

\* The init() method of the hardware class does all the work here

\*/

boolean tapeRetracting = false;

robot.init(hardwareMap, telemetry);

telemetry.addData("Status", "Initialized");

//telemetry.addData("Vertical", robot.armHorizontal);

telemetry.addData("Status", "Waiting for Start");

telemetry.addData("Version", "1.6");

telemetry.update();

/\*StateMachine armVerticalHalf = new StateMachineBuilder().numberOfStates(2).initialState(0).buildStateMachine();

armVerticalHalf.stateChange(gamepad1.dpad\_up);

if (armVerticalHalf.getState() == 1) {

robot.armVertical.setPower(1.0);

}\*/

if(gamepad1.y){

robot.setArm1(1.0);

}else if(gamepad1.x){

robot.setArm1((-1.0));

} else if (gamepad1.b) {

robot.setArm1((0.0));

}

if(gamepad1.y) {

robot.setArm2(1.0);

;

}else if(gamepad1.x){

robot.setArm2(-1.0);

}else if (gamepad1.b) {

robot.setArm2((0.0));

}

waitForStart();

while (opModeIsActive()) {

if (false) {

tankMode();

} else if (true) {

omniMode();

}

}

}

}

/\*\*

\* This if statement allows us to use the grabbers on the back of the robot.

\*/

/\*\*

\*This boolean allows us to control the horizontal arm on the robot with movements of the right stick of controller 2.

\* It also stops the arm when the joystick is not being moved.

\*/

/\*\*

\* This boolean allows us to control the vertical arm on the robot with movements of the left stick of controller 2.

\* It also stops the arm when the joystick is not being moved.

\*/

/\*\*

\* This if statement states that controls the tape measure with the left and right triggers on the controller.

\*/