package org.firstinspires.ftc.teamcode;

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

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

import com.qualcomm.robotcore.hardware.DcMotor;

import com.qualcomm.robotcore.hardware.DcMotorSimple;

import com.qualcomm.robotcore.hardware.Servo;

import org.firstinspires.ftc.robotcore.external.JavaUtil;

@TeleOp(name = "LWTV TeleOp version 2.5.4")

public class LWTV\_TeleopNOV30 extends LinearOpMode {

//servos

private Servo ClawL;

private Servo ClawR;

private Servo ARMR;

private Servo ARML;

//dc arm motor with gears :3

private DcMotor ARMMAIN;

//drive wheels

private DcMotor FR;

private DcMotor BR;

private DcMotor FL;

private DcMotor BL;

@Override

public void runOpMode() {

double SpeedMulti;

int OpenPos;

double ClosePos=.4;

int TargetRotation=0;

int StartPos=150;

int drivingPos=260

int acquiringPixelMainArm=270;

double gameplayPosServ=0;

double AcquiringPixelArmup=0;

double AcquiringPixelArmDrive=0;

double AcquiringPixelArmDown=0;

double y;

double x;

double rx;

double denominator;

//servos

ClawL = hardwareMap.get(Servo.class, "ClawL");

ClawR = hardwareMap.get(Servo.class, "ClawR");

ARMR = hardwareMap.get(Servo.class, "ARMR");

ARML = hardwareMap.get(Servo.class, "ARML");

//dc motor for arm

ARMMAIN = hardwareMap.get(DcMotor.class, "ARMMAIN");

//dc motors for driving

FR = hardwareMap.get(DcMotor.class, "FR");

BR = hardwareMap.get(DcMotor.class, "BR");

FL = hardwareMap.get(DcMotor.class, "FL");

BL = hardwareMap.get(DcMotor.class, "BL");

waitForStart();

if (opModeIsActive()) {

//setup variables related to scoring

OpenPos = 0;

ClosePos = 0.4;

ARMMAIN.setPower(1);

SpeedMulti = 1;

ClawR.setDirection(Servo.Direction.REVERSE);

ARMR.setDirection(Servo.Direction.REVERSE);

ARMMAIN.setMode(DcMotor.RunMode.STOP\_AND\_RESET\_ENCODER);

ARMMAIN.setMode(DcMotor.RunMode.RUN\_TO\_POSITION);

ARMMAIN.setTargetPosition(0);

//ARMMAIN.setDirection(DcMotor.Direction.REVERSE); disabled to simplify

ARMR.setPosition(OpenPos);

ARML.setPosition(OpenPos);

ClawL.setPosition(ClosePos);

ClawR.setPosition(ClosePos);

//setup for driving motors

telemetry.addData("current rotation", gamepad1.right\_stick\_x);

FR.setDirection(DcMotorSimple.Direction.REVERSE);

BR.setDirection(DcMotorSimple.Direction.REVERSE);

FL.setDirection(DcMotorSimple.Direction.FORWARD);

BL.setDirection(DcMotorSimple.Direction.REVERSE);

//move arm motors to position from rest pos

ARMMAIN.setTargetPosition(StartPos);

ARML.setPosition(gameplayPosServ);

ARMR.setPosition(gameplayPosServ);

while (opModeIsActive()) {

ARMMAIN.setTargetPosition(TargetRotation);

// Remember, this is reversed!

y = SpeedMulti \* -gamepad1.left\_stick\_y;

x = SpeedMulti \* -1 \* gamepad1.left\_stick\_x;

// Counteract imperfect strafing

rx = gamepad1.right\_stick\_x \* 0.7 \* SpeedMulti;

// Denominator is the largest motor power

// (absolute value) or 1.

// This ensures all the powers maintain

// the same ratio, but only when at least one is

// out of the range [-1, 1].

denominator = JavaUtil.maxOfList(JavaUtil.createListWith(JavaUtil.sumOfList(JavaUtil.createListWith(Math.abs(y), Math.abs(x), Math.abs(rx))), 1));

// Make sure your ID's match your configuration

FL.setPower(((y - x) + rx) / denominator);

BL.setPower((y + x + rx) / denominator);

FR.setPower(((y - x) - rx) / denominator);

BR.setPower(((y + x) - rx) / denominator);

telemetry.update();

// Arm and claw code

telemetry.addData("Arm rotation", ARMMAIN.getTargetPosition());

telemetry.addData("Arm position", TargetRotation);

telemetry.addData("LeftClawRot", ClawL.getPosition());

telemetry.addData("RightClawRot", ClawR.getPosition());

telemetry.update();

if (gamepad1.dpad\_up) {

TargetRotation=100;

ARML.setPosition=AcquiringPixelArmup;

ARMR.setPosition=AcquiringPixelArmup;

}

if(gamepad1.dpad\_right){

TargetRotation=drivingPos;

ARML.setPosition=AcquiringPixelArmDrive;

ARMR.setPosition=AcquiringPixelArmDrive;

}

if (gamepad1.dpad\_down) {

TargetRotation=acquiringPixelMainArm;

ARML.setPosition=AcquiringPixelArmDown;

ARMR.setPosition=AcquiringPixelArmDown;

}

if(gamepad1.a) {

ClawL.setPosition(OpenPos);

ClawR.setPosition(OpenPos);

}

if(gamepad1.b) {

ClawL.setPosition(ClosePos);

ClawR.setPosition(ClosePos);

}

}

}

}

}