

```

package org.firstinspires.ftc.teamcode;

import com.qualcomm.robotcore.eventloop.opmode.Autonomous;
import com.qualcomm.robotcore.eventloop.opmode.Disabled;
import org.firstinspires.ftc.robotcore.external.ClassFactory;
import
org.firstinspires.ftc.robotcore.external.navigation.VuMarkInstanceId;
import
org.firstinspires.ftc.robotcore.external.navigation.RelicRecoveryVuMar
k;
import
org.firstinspires.ftc.robotcore.external.navigation.VuforiaTrackableDe
faultListener;
import org.firstinspires.ftc.robotcore.external.matrices.VectorF;
import
org.firstinspires.ftc.robotcore.external.navigation.Orientation;
import
org.firstinspires.ftc.robotcore.external.navigation.AxesReference;
import org.firstinspires.ftc.robotcore.external.navigation.AxesOrder;
import org.firstinspires.ftc.robotcore.external.navigation.AngleUnit;
import
org.firstinspires.ftc.robotcore.external.navigation.VuforiaTrackables;
import
org.firstinspires.ftc.robotcore.external.navigation.VuforiaTrackable;
import org.firstinspires.ftc.robotcore.external.matrices.OpenGLMatrix;
import
org.firstinspires.ftc.robotcore.external.navigation.VuforiaLocalizer;
import com.qualcomm.robotcore.eventloop.opmode.OpMode;
import com.qualcomm.robotcore.hardware.ColorSensor;
import com.qualcomm.robotcore.hardware.DistanceSensor;
import com.qualcomm.robotcore.hardware.CRServo;
import com.qualcomm.robotcore.hardware.Servo;
import com.qualcomm.robotcore.hardware.DcMotor;
import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
import android.app.Activity;
import android.graphics.Color;
import android.view.View;

@Autonomous (name = "Red Back", group = "Red")
//@Disabled
public class RedBack extends LinearOpMode
{

    DcMotor leftWheel = null;

    DcMotor rightWheel = null;

    DcMotor pulley = null;

    Servo colorArm;

    CRServo grippers;

    ColorSensor sensorColor;

    DistanceSensor sensorDistance;

```

```

ControlClass controlClass;

boolean isVuforiaScanned = false;

public static final String TAG = "Vuforia VuMark Sample";

OpenGLMatrix lastLocation = null;

VuforiaLocalizer vuforia;

@Override
public void runOpMode () throws InterruptedException
{
    leftWheel = hardwareMap.get(DcMotor.class, "leftWheel");

//leftWheel.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);

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

//rightWheel.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);

    pulley = hardwareMap.get(DcMotor.class, "pulley");
    pulley.setZeroPowerBehavior(DcMotor.ZeroPowerBehavior.BRAKE);

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

    grippers = hardwareMap.get(CRServo.class, "grippers");

    sensorDistance = hardwareMap.get(DistanceSensor.class,
"colorSensor");

    sensorColor = hardwareMap.get(ColorSensor.class,
"colorSensor");

    controlClass = new ControlClass (leftWheel, rightWheel,
pulley, colorArm, grippers, false, false);

    controlClass.stopDriving ();

    controlClass.stopGrippers ();

    controlClass.stopPulley ();

    controlClass.liftArm ();

    boolean isVuforiaScaned = false;

    float hsvValues[] = {0F, 0F, 0F};

    final float values[] = hsvValues;

    final double SCALE_FACTOR = 255;

```

```

        int relativeLayoutId =
hardwareMap.appContext.getResources().getIdentifier("RelativeLayout",
"id", hardwareMap.appContext.getPackageName());
        final View relativeLayout = ((Activity)
hardwareMap.appContext).findViewById(relativeLayoutId);

        int cameraMonitorViewId =
hardwareMap.appContext.getResources().getIdentifier("cameraMonitorView
Id", "id", hardwareMap.appContext.getPackageName());
        VuforiaLocalizer.Parameters parameters = new
VuforiaLocalizer.Parameters(cameraMonitorViewId);

        parameters.vuforiaLicenseKey =
"AeTp02/////AAAAGRb39XgIz0/9sVz+xN6dPHpdS3/6EixBCKYbNr9Hw/vPNumcWEdd9
x4Hbk5rSeIGWS5+Of5euIDm3rKmE7GhbnVC4inOw+R5+sFSI63Qd/JVk+mg5bgXatQF2n7
3NPzKLlqMfQ6JPAEiWYzOrz5C0Sn3Cv9y3hejMCbf4eg9BmvHJogCq78iEjgShpjdWP+8g
Z5IsiXPTOLsE9pX3Zz5FV8HMDSFKpF/q0SS5IA0WyInTteYTu2Dx9glOrlIkTjrnzUJtPg
YqJ6+HJbXcnOxXyKnkT4zmxY/R/RcWF9cAx0gmSC9YmJGNxIc3rth9xv1X/UjpN7kFMbgH
aGAL/QmpOyk8cffciuibfiViagBsucS";

        parameters.cameraDirection =
VuforiaLocalizer.CameraDirection.BACK;
        this.vuforia =
ClassFactory.createVuforiaLocalizer(parameters);

        VuforiaTrackables relicTrackables =
this.vuforia.loadTrackablesFromAsset("RelicVuMark");
        VuforiaTrackable relicTemplate = relicTrackables.get(0);
        relicTemplate.setName("relicVuMarkTemplate");

        waitForStart ();

        relicTrackables.activate();

        controlClass.closeGrippers (1);

        controlClass.movePulleyUp (1);

        controlClass.dropArm ();

        while (opModeIsActive())
        {

            Color.RGBToHSV((int) (sensorColor.red() * SCALE_FACTOR),
                (int) (sensorColor.green() * SCALE_FACTOR),
                (int) (sensorColor.blue() * SCALE_FACTOR),
                hsvValues);

            RelicRecoveryVuMark vuMark =
RelicRecoveryVuMark.from(relicTemplate);
            if (vuMark != RelicRecoveryVuMark.UNKNOWN) {

                telemetry.addData("VuMark", "%s visible", vuMark);
                telemetry.update();

                OpenGLMatrix pose =

```

```

((VuforiaTrackableDefaultListener)relicTemplate.getListener()).getPose
();

        if (pose != null) {
            VectorF trans = pose.getTranslation();
            Orientation rot = Orientation.getOrientation(pose,
AxesReference.EXTRINSIC, AxesOrder.XYZ, AngleUnit.DEGREES);

            double tX = trans.get(0);
            double tY = trans.get(1);
            double tZ = trans.get(2);

            double rX = rot.firstAngle;
            double rY = rot.secondAngle;
            double rZ = rot.thirdAngle;
        }
    }
else {
    telemetry.addData("VuMark", "not visible");
    telemetry.update();
}

if (isVuforiaScanned == false)
{
    Thread.sleep(2000);
    if (vuMark == RelicRecoveryVuMark.LEFT)
    {

        telemetry.addData("VuMark", "Left");
        telemetry.update();

        int red = sensorColor.red();

        int blue = sensorColor.blue();

        if (blue>red)
        {

            controlClass.driveBackward (1500);

            /*controlClass.liftArm ();

            controlClass.turnRight (1000);

            controlClass.driveBackward (1000);

            controlClass.turnRight (300);

            controlClass.driveForward (750);

            controlClass.openGrippers (1);

            controlClass.movePulleyUp (1);

            controlClass.driveBackward (500);

```

```

        controlClass.turnLeft (175);
        controlClass.driveForward (1000*/
        isVuforiaScanned = true;
    } else
    {
        controlClass.driveForward (1000);
        isVuforiaScanned = true;
    }
} else if (vuMark == RelicRecoveryVuMark.RIGHT)
{
    telemetry.addData("VuMark", "Right");
    telemetry.update();
    int red = sensorColor.red();
    int blue = sensorColor.blue();
    if (blue>red)
    {
        controlClass.driveBackward (1500);
        controlClass.liftArm ();
        controlClass.turnRight (1000);
        controlClass.driveBackward (1000);
        controlClass.turnRight (350);
        controlClass.driveForward (750);
        controlClass.openGrippers (1);
        controlClass.movePulleyUp (1);
        controlClass.driveBackward (500);
        controlClass.turnLeft (175);
        controlClass.driveForward (1000);
        isVuforiaScanned = true;
    } else
    {
        controlClass.driveForward (200);

```

```

        controlClass.liftArm ();
        controlClass.driveBackward (1700);
        controlClass.turnRight (1000);
        controlClass.driveBackward (1000);
        controlClass.turnRight (350);
        controlClass.driveForward (750);
        controlClass.openGrippers (1);
        controlClass.movePulleyUp (1);
        controlClass.driveBackward (500);
        controlClass.turnLeft (175);
        controlClass.driveForward (1000);
        isVuforiaScanned = true;
    }

    } else if (vuMark == RelicRecoveryVuMark.CENTER ||
vuMark == RelicRecoveryVuMark.UNKNOWN)
    {

        telemetry.addData("VuMark", "Center/Not Visible");
        telemetry.update();

        int red = sensorColor.red ();

        int blue = sensorColor.blue ();

        if (blue>red)
        {

            controlClass.driveBackward (1500);

            /*controlClass.liftArm ();

            controlClass.turnRight (1000);

            controlClass.driveBackward (1000);

            controlClass.turnRight (300);

            controlClass.driveForward (750);

            controlClass.openGrippers (1);

```

```
        controlClass.movePulleyUp (1);
        controlClass.driveBackward (500);
        controlClass.turnLeft (175);
        controlClass.driveForward (1000);*/
        isVuuforiaScanned = true;
    } else
    {
        controlClass.driveForward (1000);
        isVuuforiaScanned = true;
    }
}
}
}
telemetry.update();
}
}
}
```