

```

// Harris Christiansen
// Copyright Harris Christiansen 2013
// FRC Team 3245 - Waterford School
// Las Vegas Regionals 2012 - Rev 1
// Written March 4, 2012

// Updated for Gyro 11-17-12
// Ported to NetBeans frc13 1-8-13

// Package
package edu.wpi.first.wpilibj.templates;
// For 2012 Robot - Single Stick Control

// Imports
import edu.wpi.first.wpilibj.camera.*;
import edu.wpi.first.wpilibj.*;

public class Corvey extends IterativeRobot {

    // Motors
    private Jaguar leftMotor, rightMotor;
    private Victor tipperMotor, shootMotorOne, shootMotorTwo,
topLiftMotor, btmLiftMotor;

    // Current Motor Speeds
    private double leftSpeed, rightSpeed, tipSpeed, shootSpeed,
topLiftSpeed, btmLiftSpeed;
    private double setShootSpeed;
    private boolean driveDirection=true; // True=Normal, False=Reverse

    // Autonomous Period Tracker
    private int currentPeriod=0;
    private boolean autoFeed=false;

    // Controllers
    Joystick pilotStick, copilotStick;

    // Camera
    //AxisCamera camera;

    // Pilot Controls
    int leftStick=2, rightStick=6, btmFeedBtn=7, fastBtn=6, slowBtn=5;

    // CoPilot Controls
    int tipUpBtn=4, tipDnBtn=2, CObtmFeedBtn=7, topFeedBtn=8,
revFeedBtn=4, shootBtn=8, fastShootBtn=3, slowShootBtn=1,
fastShootIncBtn=10, slowShootIncBtn=9, setShootFullBtn=11,
reverseDriveBtn=13;

```

```

// Gyro
//Gyro mainGyro;

// Encoder
//Encoder leftEncoder;

// Auto State
int autoState=0;

public void robotInit() {
    // PWM Ports
    leftMotor = new Jaguar(2);
    rightMotor = new Jaguar(1);
    tipperMotor = new Victor(5);
    shootMotorOne = new Victor(6);
    shootMotorTwo = new Victor(7);
    topLiftMotor = new Victor(4);
    btmLiftMotor = new Victor(3);

    // Joysticks
    pilotStick = new Joystick(1);
    copilotStick = new Joystick(2);

    // Camera
    //camera = AxisCamera.getInstance();

    // Zero Everything

```

```

leftSpeed=0;rightSpeed=0;tipSpeed=0;shootSpeed=0;topLiftSpeed=0;btmLiftSpeed=0;

```

```

    //mainGyro=new Gyro(1,1);
    //mainGyro.setSensitivity(.007);

    //leftEncoder=new Encoder(1,2);
    //leftEncoder.start();
}
int autoShootDelay=1;
public void autonomousInit() {
    //autoShootDelay=40;
    setShootSpeed=0.35;
    currentPeriod=0;
    driveDirection=true;
}
public void autonomousPeriodic() {
    if(autoFeed) {
        autoReverse();
    } else {
        autoWaitShoot();
    }
}

```

```

        //autoTurnDrive();
    }
    updateMotors();
}
public void teleopInit() {
    setShootSpeed=0.95;
    // Zero Everything

leftSpeed=0;rightSpeed=0;tipSpeed=0;shootSpeed=0;topLiftSpeed=0;btmLiftSpeed=0;
    driveDirection=true;
}
public void teleopPeriodic() {
    //updateLifter();
    ///updateShooterSpeed();
    //updateShooter();
    //updateTipper();
    updateDrive();
    updateReverseDrive();
    updateShootLift();
    updateMotors();
}
public void disabledInit() {

}
public void disabledPeriodic() {
    if(DriverStation.getInstance().getDigitalIn(1))
{ autoShootDelay=40; }
    if(DriverStation.getInstance().getDigitalIn(2))
{ autoShootDelay=215; }
    if(DriverStation.getInstance().getDigitalIn(3))
{ autoShootDelay=435; }
    if(DriverStation.getInstance().getDigitalIn(4))
{ autoFeed=true; }
}
public void testInit() {

}
public void testPeriodic() {

}

//// Custom Functions
////////// Custom Functions //////////

public int currentPeriodShootLift=0;
public void updateShootLift() {
    if(pilotStick.getRawButton(shootBtn)) {
        if(currentPeriodShootLift<=120) {

```

```

        tipSpeed=0;
        shootSpeed=setShootSpeed;
        topLiftSpeed=0;
        btmLiftSpeed=0;
    }
    else {
        tipSpeed=0;
        shootSpeed=setShootSpeed;
        topLiftSpeed=0.65;
        btmLiftSpeed=0.42;
    }
    currentPeriodShootLift++;
} else {
    currentPeriodShootLift=0;
    tipSpeed=0;
    shootSpeed=0.0;
    topLiftSpeed=0;
    btmLiftSpeed=0;
}
// Reverse Feeder
if(pilotStick.getRawButton(revFeedBtn))
{btmLiftSpeed=-0.3;topLiftSpeed=-0.3;}
if(pilotStick.getRawButton(btmFeedBtn)) {btmLiftSpeed=0.5;}
}

// Lifter Functions
public void updateLifter() {
    // Bottom Feeder
    if(pilotStick.getRawButton(btmFeedBtn)) {btmLiftSpeed=0.5;}
    else if(copilotStick.getRawButton(C0btmFeedBtn))
{btmLiftSpeed=0.5;}
    else {btmLiftSpeed=0.0;}

    // Top Feeder
    if(copilotStick.getRawButton(topFeedBtn)) {topLiftSpeed=0.5;}
    else {topLiftSpeed=0.0;}

    // Reverse Feeder
    if(copilotStick.getRawButton(revFeedBtn))
{btmLiftSpeed=-0.3;topLiftSpeed=-0.3;}
}

// Shooter Functions
public void updateShooter() {
    if(copilotStick.getRawButton(shootBtn))
{shootSpeed=setShootSpeed;}
    else {shootSpeed=0.0;}
}

// Shooter Speed Changer

```

```

    private boolean shooterIncred=false;
    public void updateShooterSpeed() {
        if(copilotStick.getRawButton(fastShootBtn)&&!shooterIncred)
{setShootSpeed=setShootSpeed
+0.05;shooterIncred=true;System.out.println((int)
(setShootSpeed*100));}
        else if(copilotStick.getRawButton(slowShootBtn)&&!
shooterIncred)
{setShootSpeed=setShootSpeed-0.05;shooterIncred=true;System.out.printl
n((int)(setShootSpeed*100));}
        else if(pilotStick.getRawButton(fastShootIncBtn)&&!
shooterIncred) {setShootSpeed=setShootSpeed
+0.01;shooterIncred=true;System.out.println((int)
(setShootSpeed*100));}
        else if(pilotStick.getRawButton(slowShootIncBtn)&&!
shooterIncred)
{setShootSpeed=setShootSpeed-0.01;shooterIncred=true;System.out.printl
n((int)(setShootSpeed*100));}
        if(!copilotStick.getRawButton(fastShootBtn)&&!
copilotStick.getRawButton(slowShootBtn)&&!
pilotStick.getRawButton(fastShootIncBtn)&&!
pilotStick.getRawButton(slowShootIncBtn)) {shooterIncred=false;}
        if(copilotStick.getRawButton(setShootFullBtn))
{ setShootSpeed=0.65; }
    }
    private boolean driveChanged=false;
    public void updateReverseDrive() {
        if(copilotStick.getRawButton(reverseDriveBtn)&&!
driveChanged&&driveDirection) {driveChanged=true;
driveDirection=false;}
        else if(copilotStick.getRawButton(reverseDriveBtn)&&!
driveChanged&&!driveDirection) {driveChanged=true;
driveDirection=true;}
        if(!copilotStick.getRawButton(reverseDriveBtn))
{driveChanged=false;}
    }

    // Tipper Functions
    public void updateTipper() {
        if(copilotStick.getRawButton(tipUpBtn)) {tipSpeed=0.30;}
        else if(copilotStick.getRawButton(tipDnBtn)) {tipSpeed=-0.22;}
        else {tipSpeed=0.0;}
    }

    // Drive Functions
    public void updateDrive() {
        double drivePercent=0.55;
        if(pilotStick.getRawButton(fastBtn)) {drivePercent=1.00;}
        else if(pilotStick.getRawButton(slowBtn)) {drivePercent=0.3;}
        leftSpeed=pilotStick.getRawAxis(leftStick)*drivePercent;
    }

```

```

    rightSpeed=pilotStick.getRawAxis(rightStick)*drivePercent;
}

// Set Motor Speeds
public void updateMotors() {
    if(driveDirection) { // Normal - Drive Forward
        leftMotor.set(leftSpeed);
        rightMotor.set(-rightSpeed); // Motor Reversed
    } else if(!driveDirection) { // Reverse Drive
        leftMotor.set(-rightSpeed);
        rightMotor.set(leftSpeed); // Motor Reversed
    }
    tipperMotor.set(-tipSpeed); // Motor Reversed
    shootMotorOne.set(shootSpeed);
    shootMotorTwo.set(shootSpeed);
    topLiftMotor.set(-topLiftSpeed); // Motor Reversed
    btmLiftMotor.set(-btmLiftSpeed); // Motor Reversed

    // Output Gyro+Encoder Data
    //double currentAngle=mainGyro.getAngle();
    //int currentTicks=leftEncoder.get();
    //System.out.println("Angle: "+currentAngle+" Ticks:
"+currentTicks);
}

public void autoWaitShoot() {
    if(currentPeriod<=autoShootDelay) {
        leftSpeed=0;
        rightSpeed=0;
        tipSpeed=0;
        shootSpeed=0;
        topLiftSpeed=0;
        btmLiftSpeed=0;
    }
    else if(currentPeriod<=1600) {
        leftSpeed=0;
        rightSpeed=0;
        tipSpeed=0;
        shootSpeed=setShootSpeed;
        topLiftSpeed=0.4;
        btmLiftSpeed=0.35;
    }
    currentPeriod++;
}

public void autoReverse() {
    if(currentPeriod<=1600) {
        leftSpeed=0;
        rightSpeed=0;
        tipSpeed=0;
        shootSpeed=0.0;
    }
}

```

```
        topLiftSpeed=-0.4;
        btmLiftSpeed=-0.35;
    }
    currentPeriod++;
}
```

```
//// Autonomous
public void autoTurnDrive() {
    if(autoState==0) { turnToAngle(360); }
    //else if(autoState==1) { driveAtAngle(30.0,1200); }
    //else if(autoState==2) { turnToAngle(210); }
    //else if(autoState==3) { driveAtAngle(210.0,1200); }
    //else if(autoState==4) { turnToAngle(0); }
    else { zeroAll(); }
}
```

```
public void zeroAll() {
    leftSpeed=0;
    rightSpeed=0;
    tipSpeed=0;
    shootSpeed=0.0;
    topLiftSpeed=0.0;
    btmLiftSpeed=0.0;
}
```

```
public void nextState() {
    //leftEncoder.reset();
    autoState++;
}
```

```
// P Loop Turning
public void turnToAngle(double targetAngle) {
    //double currentAngle=mainGyro.getAngle();
    double currentAngle=0;
    double angleOff=targetAngle-currentAngle;
```

```

    double mult=0.03;
    double turnSpeed=angleOff*mult;
    if(turnSpeed>0) {
        leftSpeed=-turnSpeed;
        rightSpeed=0;
    } else {
        leftSpeed=0;
        rightSpeed=turnSpeed;
    }
    if(Math.abs(currentAngle-targetAngle)>=3) {
        leftSpeed=0;
        rightSpeed=0;
        nextState();
    }
}

// P Loop Driving
public void driveAtAngle(double targetAngle, int ticksForward) {
    //double currentAngle=mainGyro.getAngle();
    //int currentTicks=leftEncoder.get();
    double currentAngle=0;
    int currentTicks=0;
    double mult=0.04;
    double angleOff=targetAngle-currentAngle;
    double turn = mult*angleOff;
    if(ticksForward>0&&currentTicks<(ticksForward-100)){
        leftSpeed=-0.5-turn;
        rightSpeed=-0.5+turn;
    }
    else if(ticksForward<0&&currentTicks>(ticksForward+100)){
        leftSpeed=0.5-turn;
        rightSpeed=0.5+turn;
    }
    else {
        leftSpeed=0;
        rightSpeed=0;
        nextState();
    }
}
}
}

```