```
#pragma config(Sensor, in1,    expander,       sensorAnalog)
#pragma config(Sensor, in2,    gyro,           sensorGyro)
#pragma config(Sensor, dgtl1,  armQuad,        sensorQuadEncoder)
#pragma config(Sensor, dgtl3,  gateSwitch,     sensorDigitalIn)
#pragma config(Sensor, dgtl4,  armSwitch,      sensorDigitalIn)
#pragma config(Sensor, dgtl5,  gateQuad,       sensorQuadEncoder)
#pragma config(Motor,  port2,           yOne,          tmotorVex393HighSpeed_MC29, openLoop, reversed)
#pragma config(Motor,  port3,           yTwo,          tmotorVex393HighSpeed_MC29, openLoop)
#pragma config(Motor,  port4,           yThree,        tmotorVex393HighSpeed_MC29, openLoop, reversed)
#pragma config(Motor,  port5,           gate,          tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port6,           frontRight,    tmotorVex393HighSpeed_MC29, openLoop, reversed)
#pragma config(Motor,  port7,           backRight,     tmotorVex393HighSpeed_MC29, openLoop, reversed)
#pragma config(Motor,  port8,           backLeft,      tmotorVex393HighSpeed_MC29, openLoop)
#pragma config(Motor,  port9,           frontLeft,     tmotorVex393HighSpeed_MC29, openLoop)
#pragma config(Motor,  port10,          intake,        tmotorVex393HighSpeed_HBridge, openLoop)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

#pragma platform(VEX)

#pragma competitionControl(Competition)
#pragma autonomousDuration(15)
#pragma userControlDuration(105)

#include "Vex_Competition_Includes.c"
#include "catapult.c"

task autonomous(){
	//Set gate and arm statuses to initial values
	setGate(GATE_OPEN);
	setArm(ARM_LOAD);

	//Zero Sensors
	resetSensors();

	//Turn the intake on
	motor[intake] = -127;

	//Start the gate and arm monitoring processes
	startTask(armPosition);
	startTask(gatePosition);

	//fire four balls
	ballCount = 0;
	while(ballCount < 4){
```

```
                setArm(ARM_FIRE);
                setDistance(TILE);
                wait1Msec(250);
        }
}

task usercontrol(){
        //Set gate and arm statuses to initial values
        setGate(GATE_OPEN);
        setArm(ARM_FIRE);

        //Fire for one cycle no matter what the armSwitch is giving
        override = true;

        //Start the gate and arm monitoring processes
        startTask(armPosition);
        startTask(gatePosition);

        //Turn intake on
        motor[intake] = -127;

        while(true){
                //Firing Controls
                if(vexRT[Btn7U] == 1){
                        setDistance(CORNER);
                        setArm(ARM_FIRE);
                        }
                        else{
                        if(vexRT[Btn7L] == 1){
                                setDistance(TILE);
                                setArm(ARM_FIRE);
                                }
                                else{
                                if(vexRT[Btn7D] == 1){
                                        setDistance(MID);
                                        setArm(ARM_FIRE);
                                        }
                                        else{
                                        if(vexRT[Btn7R] == 1){
                                                setDistance(SHORT);
                                                setArm(ARM_FIRE);
                                        }
                                }
                        }
```

```
                }

                //Manual Gate Open
                if(vexRT[Btn8R] == true){
                        setGate(GATE_OPEN);
                }

                //Drive Base controls
                if(vexRT[Btn6D] == true){
                        rotate(false);
                }
                else{
                        if(vexRT[Btn5D] == true){
                                rotate(true);
                        }
                        else{
                                if(vexRT[Btn6U] == true){
                                        mechanumDrive(false);
                                }
                                else{
                                        if(vexRT[Btn5U] == true){
                                                mechanumDrive(true);
                                        }
                                        else{
                                                tankDrive();
                                        }
                                }
                        }
                }
        }
}
```