```
/*----------------------------------------------------------------------------*/
/*                                                                            */
/*    Module:       main.cpp                                                  */
/*    Author:       VEX                                                       */
/*    Created:      Thu Sep 26 2019                                           */
/*    Description:  Competition Template                                      */
/*                                                                            */
/*----------------------------------------------------------------------------*/

// ---- START VEXCODE CONFIGURED DEVICES ----
// Robot Configuration:
// [Name]               [Type]        [Port(s)]
// Intake               motor         1
// Controller1          controller
// BumperA              bumper        A
// ---- END VEXCODE CONFIGURED DEVICES ----

#include "vex.h"

using namespace vex;

// A global instance of competition
competition Competition;




int AutonSelected = 0;
int AutonMin = 0;
int AutonMax = 4;



// define your global instances of motors and other devices here



/*----------------------------------------------------------------------------*/
/*                           Pre-Autonomous Functions                         */
/*                                                                            */
/*  You may want to perform some actions before the competition starts.       */
/*  Do them in the following function.  You must return from this function    */
/*  or the autonomous and usercontrol tasks will not be started.  This        */
/*  function is only called once after the V5 has been powered on and         */
/*  not every time that the robot is disabled.                                */
```

```
/*---------------------------------------------------------------------------*/

void pre_auton(void) {
 // Initializing Robot Configuration. DO NOT REMOVE!

 //Draws Field Setup
Brain.Screen.setPenColor(white);
Brain.Screen.drawRectangle(5,5,475,235);

Brain.Screen.setFillColor(blue);
Brain.Screen.setPenColor(black);
Brain.Screen.drawRectangle(15,15, 220, 60);
Brain.Screen.printAt(110,50, "BLUE 1");

Brain.Screen.drawRectangle(260, 15, 210, 60);
Brain.Screen.printAt(360, 50, "BLUE 2");



Brain.Screen.setPenColor(black);
Brain.Screen.setFillColor(red);
Brain.Screen.drawRectangle(15, 170, 220, 60);
Brain.Screen.printAt(110, 210, "RED 2");

Brain.Screen.drawRectangle(260, 170, 210, 60);
Brain.Screen.printAt(360, 210, "RED 1");


Brain.Screen.setFillColor(black);
Brain.Screen.setPenColor(blue);
Brain.Screen.drawRectangle(15, 100, 180, 60);
Brain.Screen.printAt(25, 130, "BLUE GOAL");


Brain.Screen.setPenColor(red);
Brain.Screen.drawRectangle(280, 100, 180, 60);
Brain.Screen.printAt(360, 135, "RED GOAL");



}



 // All activities that occur before the competition starts
 // Example: clearing encoders, setting servo positions, .}
```

```
/*----------------------------------------------------------------------------*/
/*                                                                            */
/*                              Autonomous Task                               */
/*                                                                            */
/*  This task is used to control your robot during the autonomous phase of   */
/*  a VEX Competition.                                                        */
/*                                                                            */
/*  You must modify the code to add your own robot specific commands here.   */
/*----------------------------------------------------------------------------*/


//Program Autonomous Here

void screenPressed (void) {

 int x=Brain.Screen.xPosition();
 int y=Brain.Screen.yPosition();

   Drivetrain.setDriveVelocity(50, pct);
   HangMech.setVelocity(70, pct);
   Intake.setVelocity(85, pct);
   LeftCata.setVelocity(80, pct);
   RightCata.setVelocity(80, pct);


//BLUE 1 PROGRAM
if(x>=15 && x<=205 && y>=15 && y<= 45)
{
 Brain.Screen.clearScreen();
 Brain.Screen.printAt(200, 120, "BLUE 1 Running!");
 //CODE FOR BLUE 1 AUTONOMOUS HERE

 //Reverses To Flip Out Intake
  Drivetrain.driveFor(forward, 80 , mm);

 //Drives Forward
 Drivetrain.driveFor(reverse, 800 ,mm);

 //Turns Left
 RightDriveSmart.spinFor(reverse, 110 ,degrees, false);
 LeftDriveSmart.spinFor(forward, 110,degrees);

 //Drives Forward
 Drivetrain.driveFor(reverse, 300, mm);

 //Drives Reverse
```

```
 Drivetrain.driveFor(forward, 300 ,mm);



}

//BLUE 2 PROGRAM
if(x>=260 && x<=470 && y>=15 && y<= 45)
{
 Brain.Screen.clearScreen();
 Brain.Screen.printAt(200, 120, "BLUE 2 Running!");
 //CODE FOR BLUE 2 AUTONOMOUS HERE

}


//RED 2 PROGRAM
if(x>=15 && x<=205 && y>=170 && y<= 230)
{
 Brain.Screen.clearScreen();
 Brain.Screen.printAt(190, 120, "RED 2 Running!");
 //CODE FOR RED 2 AUTONOMOUS HERE
}


//RED 1 PROGRAM
if(x>=260 && x<=470 && y>=170 && y<= 230)
{


 Brain.Screen.clearScreen();
 Brain.Screen.printAt(190, 120, "RED 1 Running!");
 //CODE FOR RED 1 AUTONOMOUS HERE


 //Reverses To Flip Out Intake
  Drivetrain.driveFor(forward, 80 , mm);

 //Drives Forward
 Drivetrain.driveFor(reverse, 800 ,mm);

 //Turns Left
 RightDriveSmart.spinFor(reverse, 110 ,degrees, false);
 LeftDriveSmart.spinFor(forward, 110,degrees);

 //Drives Forward
 Drivetrain.driveFor(reverse, 300, mm);
```

```
 //Drives Reverse
 Drivetrain.driveFor(forward, 300 ,mm);


 //Turns Left
 RightDriveSmart.spinFor(reverse, 110 ,degrees, false);
 LeftDriveSmart.spinFor(forward, 110,degrees);

 //Drives Forward
 Drivetrain.driveFor(reverse, 800 ,mm);









}

}
/*---------------------------------------------------------------------------*/
/*                                                                           */
/*                              User Control Task                            */
/*                                                                           */
/*  This task is used to control your robot during the user control phase of */
/*  a VEX Competition.                                                       */
/*                                                                           */
/*  You must modify the code to add your own robot specific commands here.   */
/*---------------------------------------------------------------------------*/

void usercontrol(void) {
 // User control code here, inside the loop
 while (1) {
   // This is the main execution loop for the user control program.
   // Each time through the loop your program should update motor + servo
   // values based on feedback from the joysticks.
   Drivetrain.setDriveVelocity(100, pct);
   HangMech.setVelocity(70, pct);
   Intake.setVelocity(85, pct);
   LeftCata.setVelocity(80, pct);
   RightCata.setVelocity(80, pct);
```

```
    //Drive Control


  RightDriveSmart.spin(vex::directionType::rev, Controller1.Axis2.position(),
vex::velocityUnits::pct);
  LeftDriveSmart.spin(vex::directionType::rev, Controller1.Axis3.position(),
vex::velocityUnits::pct);







    //Catapult Control

    if (Controller1.ButtonL1.pressing()){
  LeftCata.spin(forward);
 RightCata.spin(forward);
}
else if (Controller1.ButtonL2.pressing()){
  LeftCata.spin(reverse);
 RightCata.spin(reverse);
}
else if (LimitS.pressing()){
  LeftCata.stop(brakeType::hold);
 RightCata.stop(brakeType::hold);
}
else {
 LeftCata.spin(forward);
 RightCata.spin(forward);
}



    //Intake Control


    if(Controller1.ButtonR1.pressing()) {
      Intake.spin(forward);
    }
    else if(Controller1.ButtonR2.pressing())  {
      Intake.spin(reverse);
    }

    else {
```

```
      Intake.stop(vex::brakeType::coast);
    }


//Hang Mech Control

 if(Controller1.ButtonX.pressing()) {
   HangMech.spin(forward);
 }

 else if(Controller1.ButtonY.pressing()) {
   HangMech.spin(reverse);
 }

 else {
   HangMech.stop(vex::brakeType::hold);
 }









    // ..........................................................................

    wait(20, msec); // Sleep the task for a short amount of time to
                    // prevent wasted resources.
 }
}

//
// Main will set up the competition functions and callbacks.
//
int main() {
 // Set up callbacks for autonomous and driver control periods.
 Competition.autonomous(screenPressed);
 Competition.drivercontrol(usercontrol);


Brain.Screen.pressed(screenPressed);


 // Run the pre-autonomous function.
 pre_auton();
```

```
 // Prevent main from exiting with an infinite loop.
 while (true) {
   wait(100, msec);
 }
}
```