```c
/**
 * @file autoSkills.c
 * @brief Programming skills
 * Copyright (C) 2017 Ethan Wells
 *
 * This program is free software: you can redistribute it and/or modify it
 * under the terms of the GNU General Public License as published by the Free
 * Software Foundation, either version 3 of the License, or(at your option) any
 * later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
 * FOR A PARTICULAR PURPOSE. See the GNU General Public License for more
 * details.
 *
 * You should have received a copy of the GNU General Public License along
 * with this program. If not, see <https://www.gnu.org/licenses/>
 */

#include "../include/auto.h"

void autonLeft22();
void armPID(void *none);

void autonSkills() {
        // Get the mobile goal using the left red 22 point auton routine
        autonLeft22();

        // ////////////////////////////////////////////////////////////////// 22 POINTS

        resetDrive();

        driveToPosition(-200, -200, 750);
        turnTo(136, 2500);     // TURN AROUND,
        GO(mogoPT, MOGO_DOWN); // DROP A MOGO INTAKE;
        delay(350);

        // Reset sensors
        resetDrive();
        driveToPositionAngle(685, 675, 135, 1550);
        sensorRefresh(sonic);
        #ifdef DEBUG_MODE
                printf("\n\n%d\n\n", sonic->value);
        #endif
        turnTo(203, 2500); // TURN AROUND,
```

```
// Reset sensors
resetDrive();

driveToPositionAngle(1650, 1650, 206, 3200);

// Mogo intake up
mogoP(MOGO_UP + 100);
turnTo(380, 2900); // TUUUURN AROUUUND

// Reset sensors
resetDrive();

driveToPositionAngle(1300, 1400, 373, 1875);
// Bring mogo intake to middle
TaskHandle mogoHandle = GO(mogoPT, MOGO_MID);
driveToPositionAngle(2375, 2475, 371, 2750);

if (taskGetState(mogoHandle))
        taskDelete(mogoHandle);

mogoP(MOGO_DOWN);
driveSet(-64, -64);
delay(150);

GO(mogoPT, MOGO_MID - 100);
delay(150);

driveSet(70, 70);
delay(200);
driveSet(28, 28);
delay(300);
driveSet(10, 10);
delay(125);
sensorReset(&gyro);

static const float driveChange = .76;
driveSettings[0].kP *= driveChange;
driveSettings[1].kP *= driveChange;
driveToPosition(2325, 2325, 1800);
driveSettings[0].kP /= driveChange;
driveSettings[1].kP /= driveChange;


// ////////////////////////////////////////////////////////////// 32 POINTS

turnTo(-104, 3400);
```

```
sensorRefresh(sonic);
#ifdef DEBUG_MODE
        printf("\n\n%d\n\n", sonic->value);
#endif

driveSet(44, 44);
int sonicGoal = 62; // 62; // 43;
do {
        sensorRefresh(sonic);
        info();
        delay(20);
} while (sonic->value == 0 || sonic->value > sonicGoal);
driveSet(-35, -35);
delay(175);
driveSet(0, 0);

turnTo(-201, 2900); // TURN AROUND,

// Reset sensors
resetDrive();

mogoHandle = GO(mogoPT, MOGO_DOWN);
delay(300);
driveToPositionAngle(1430, 1430, -205, 2600);

// Mogo intake up
mogoP(MOGO_UP);
driveToPositionAngle(750, 750, -204, 1200);
turnTo(-362, 2000); // TUUUURN AROUUUND

// Reset sensors
resetDrive();

driveToPositionAngle(900, 800, -354, 1876);
// Bring mogo intake to middle
mogoHandle = GO(mogoPT, MOGO_MID);
driveToPositionAngle(1325, 1225, -343, 2200);
driveSet(17, 17);
delay(250);

if (taskGetState(mogoHandle))
        taskDelete(mogoHandle);
mogoP(MOGO_DOWN);
driveSet(10, 10);
delay(250);
```

```
// Reset drive and gyro sensors
resetDrive();
sensorReset(&gyro);

driveSet(-64, -64);
delay(250);
mogoHandle = GO(mogoPT, MOGO_MID);
// Back up a small amount
driveToPosition(-450, -450, 750);

// ////////////////////////////////////////////////////////////////// 42 POINTS

// Turn around to aim for the red mogo across the field
turnTo(175, 3000);
resetDrive();

// Drive across the field
driveToPositionAngle(600,  600,  176, 1200);
// Mogo intake down to pick it up
mogoHandle = GO(mogoPT, MOGO_DOWN);
driveToPositionAngle(3300, 3200, 174, 3900);
mogoHandle = GO(mogoPT, MOGO_MID - 200);
driveToPosition(3700, 3600, 1200);
turnTo(173, 600);

resetDrive();

driveToPosition(100, 500, 1000);
driveToPosition(830, 1180, 800);
driveToPosition(1380, 1280, 1000);
driveToPositionAngle(1600, 1500, 185, 1700);

mogoHandle = GO(mogoPT, MOGO_MID + 125);
driveSet(127, 127);
delay(450);

// Reset drive encoders & gyro
// resetDrive();

if (taskGetState(mogoHandle))
        taskDelete(mogoHandle);

mogo.power = 127;
motorUpdate(&mogo);
driveSet(64, 64);
```

```
delay(175);
sensorReset(&gyro);

mogo.power = 30;
motorUpdate(&mogo);
driveSet(-127, -127);

delay(125);

if (taskGetState(mogoHandle))
        taskDelete(mogoHandle);
mogoHandle = GO(mogoPT, MOGO_MID);

driveToPosition(1100, 1000, 2095);

driveSet(70, 70);
delay(250);
driveSet(28, 28);
delay(300);
driveSet(10, 10);
delay(150);
sensorReset(&gyro);
resetDrive();

// ////////////////////////////////////////////////////////////////// 62 POINTS

// driveSettings[0].kP *= driveChange;
// driveSettings[1].kP *= driveChange;
driveToPosition(-100, -100, 750);
// driveSettings[0].kP /= driveChange;
// driveSettings[1].kP /= driveChange;

turnTo(-102, 3400);

sensorRefresh(sonic);
#ifdef DEBUG_MODE
        printf("\n\n%d\n\n", sonic->value);
#endif

driveSet(44, 44);
sonicGoal = 84;
do {
        sensorRefresh(sonic);
        info();
        delay(20);
```

```
} while (sonic->value == 0 || sonic->value > sonicGoal);
driveSet(-35, -35);
delay(175);
driveSet(0, 0);

turnTo(-216, 2900); // TURN AROUND,

// Reset sensors
resetDrive();

mogoHandle = GO(mogoPT, MOGO_DOWN);
delay(300);
driveToPositionAngle(1430, 1430, -220, 2600);

// Mogo intake up
mogoP(MOGO_UP);
driveToPositionAngle(750, 750, -219, 1200);
turnTo(-355, 2000); // TUUUURN AROUUUND

// Reset sensors
resetDrive();

driveToPositionAngle(900, 800, -347, 1876);
// Bring mogo intake to middle
mogoHandle = GO(mogoPT, MOGO_MID);
driveToPositionAngle(1325, 1225, -336, 2200);

// Reset sensors
resetDrive();

if (taskGetState(mogoHandle))
        taskDelete(mogoHandle);
mogoP(MOGO_DOWN);
driveSet(17, 17);
delay(250);

resetDrive();
sensorReset(&gyro);

driveSet(-64, -64);
delay(250);
mogoHandle = GO(mogoPT, MOGO_MID);
// Back up a small amount
driveToPosition(-450, -450, 750);

// ////////////////////////////////////////////////////////////// 72 POINTS
```

```
} /* autonSkills */
```