

ticTacToeDisplay.c

```

* ticTacToeDisplay.c

#include "ticTacToeDisplay.h"
#include "supportFiles/utils.h" //needed to use the delays function
#include "../Lab2_SwitchesAndButtons/buttons.h" //needed to use buttons 0 and 1
#include "../Lab2_SwitchesAndButtons/switches.h" //needed to use switch 0
#include "stdio.h"

#define LEFT_VERT_LINE DISPLAY_WIDTH/3 //x coordinate for left vertical line
#define RIGHT_VERT_LINE 2*DISPLAY_WIDTH/3 //x coordinate for right vertical line
#define UPPER_HORIZ_LINE DISPLAY_HEIGHT/3 //y coordinate for upper horizontal line
#define LOWER_HORIZ_LINE 2*DISPLAY_HEIGHT/3 //y coordinate for lower horizontal line

#define X1_LEFT_COL DISPLAY_WIDTH/12 //x for left side of X's in left column
#define X2_LEFT_COL LEFT_VERT_LINE-DISPLAY_WIDTH/12 //x for right side of X's in left column
#define X1_MID_COL LEFT_VERT_LINE+DISPLAY_WIDTH/12 //x for left side of X's in middle column
#define X2_MID_COL RIGHT_VERT_LINE-DISPLAY_WIDTH/12 //x for right side of X's in middle column
#define X1_RIGHT_COL RIGHT_VERT_LINE+DISPLAY_WIDTH/12 //x for left side of X's in right column
#define X2_RIGHT_COL DISPLAY_WIDTH-DISPLAY_WIDTH/12 //x for right side of X's in right column

#define Y1_TOP_ROW DISPLAY_HEIGHT/15 //y for top of X's in top row
#define Y2_TOP_ROW UPPER_HORIZ_LINE-DISPLAY_HEIGHT/15 //y for bottom of X's in top row
#define Y1_MID_ROW UPPER_HORIZ_LINE+DISPLAY_HEIGHT/15 //y for top of X's in middle row
#define Y2_MID_ROW LOWER_HORIZ_LINE-DISPLAY_HEIGHT/15 //y for bottom of X's in middle row
#define Y1_BOT_ROW LOWER_HORIZ_LINE+DISPLAY_HEIGHT/15 //y for top of X's in bottom row
#define Y2_BOT_ROW DISPLAY_HEIGHT-DISPLAY_HEIGHT/15 //y for bottom of X's in bottom row

#define X_CIR_LEFT_COL DISPLAY_WIDTH/6 //x for midpoint of O's in left column
#define X_CIR_MID_COL DISPLAY_WIDTH/2 //x for midpoint of O's in middle column
#define X_CIR_RIGHT_COL 5*DISPLAY_WIDTH/6 //x for midpoint of O's in right column
#define Y_CIR_TOP_ROW DISPLAY_HEIGHT/6 //y for midpoint of O's in top row
#define Y_CIR_MID_ROW DISPLAY_HEIGHT/2 //y for midpoint of O's in middle row
#define Y_CIR_BOT_ROW 5*DISPLAY_HEIGHT/6 //y for midpoint of O's in bottom row
#define CIR_RADIUS DISPLAY_HEIGHT/10 //radius of all the O's

#define Y_0 0 //y coordinate value of zero
#define X_0 0 //x coordinate value of zero

#define RUNNING 1 //used to keep while loop going
#define TOUCH_DELAY 50 //50 ms, necessary delay after registering a touch

#define COLUMN_0 0 //left column on board
#define COLUMN_1 1 //middle column on board
#define COLUMN_2 2 //right column on board

#define ROW_0 0 //top row
#define ROW_1 1 //middle row
#define ROW_2 2 //bottom row

#define TEST_OVER "Yo yo, you done \nfinished dis test\n\r" //message to print when BTN1 is pressed
#define TEXT_SIZE 3 //text size of final message

```

ticTacToeDisplay.c

```
// Inits the tic-tac-toe display, draws the lines that form the board.
void ticTacToeDisplay_init(){
    display_init(); // Must init all of the software and underlying hardware for LCD.
    display_fillScreen(DISPLAY_BLACK); // Blank the screen.
    ticTacToeDisplay_drawBoardLines(); //we have to draw the board when starting the game
}

// Draws an X at the specified row and column.
// erase == true means to erase the X by redrawing it as background. erase == false, draw
the X as foreground.
void ticTacToeDisplay_drawX(uint8_t row, uint8_t column, bool erase){
    uint16_t x1, x2, y1, y2; //X's need 4 coordinate values
    if(column == COLUMN_0){ //if touch was in left column, we know what the x-coordinates
need to be
        x1 = X1_LEFT_COL; //these 2 constants are described above
        x2 = X2_LEFT_COL;
    }
    else if(column == COLUMN_1){ //if touch was in middle column, set the corresponding x
coordinates
        x1 = X1_MID_COL; //these 2 constants are described above
        x2 = X2_MID_COL;
    }
    else{ //if it gets here, we know the touch was in the right column
        x1 = X1_RIGHT_COL; //these 2 constants are described above
        x2 = X2_RIGHT_COL;
    }

    if(row == ROW_0){ //check if touch was in top row
        y1 = Y1_TOP_ROW; //these 2 constants are described above
        y2 = Y2_TOP_ROW;
    }
    else if(row == ROW_1){ //check if touch was in middle row
        y1 = Y1_MID_ROW; //these 2 constants are described above
        y2 = Y2_MID_ROW;
    }
    else{ //if it gets here, touch was in bottom row
        y1 = Y1_BOT_ROW; //these 2 constants are described above
        y2 = Y2_BOT_ROW;
    }

    //draw the 2 lines needed to form an x
    if(!erase){
        display_drawLine(x1, y1, x2, y2, DISPLAY_YELLOW);
        display_drawLine(x1, y2, x2, y1, DISPLAY_YELLOW);
    }
    else{
        display_drawLine(x1, y1, x2, y2, DISPLAY_BLACK);
        display_drawLine(x1, y2, x2, y1, DISPLAY_BLACK);
    }
}

// Draws an O at the specified row and column.
// erase == true means to erase the X by redrawing it as background. erase == false, draw
the X as foreground.
void ticTacToeDisplay_drawO(uint8_t row, uint8_t column, bool erase){
    uint16_t x,y; //to draw a circle, we just need x-y coordinate pair for the center of
```

ticTacToeDisplay.c

the circle and the radius

```
    if(column == COLUMN_0) x = X_CIR_LEFT_COL; //if touch was in left column, set it to
constant described above
    else if(column == COLUMN_1) x = X_CIR_MID_COL; //if touch was in middle column, set
it to constant described above
    else x = X_CIR_RIGHT_COL; //if touch was in right column, set it to constant
described above

    if(row == ROW_0) y = Y_CIR_TOP_ROW; //top row touch, set it to constant described
above
    else if(row == ROW_1) y = Y_CIR_MID_ROW; //middle row touch, set it to constant
described above
    else y = Y_CIR_BOT_ROW; //bottom row touch, set it to constant described above

    if(!erase) display_drawCircle(x, y, CIR_RADIUS, DISPLAY_YELLOW); //draw the O. Color
yellow
    else display_drawCircle(x, y, CIR_RADIUS, DISPLAY_BLACK);
}

// After a touch has been detected and after the proper delay, this sets the row and
column arguments
// according to where the user touched the board.
void ticTacToeDisplay_touchScreenComputeBoardRowColumn(uint8_t* row, uint8_t* column){
    int16_t x, y; //x-y coordinates of touch
    uint8_t z;    //pressure of touch
    display_getTouchedPoint(&x, &y, &z); //used to get the x,y coordinates for the touch

    //sets row to the horizontal third of the board the touch was in
    if(y <= UPPER_HORIZ_LINE) //check if touch was in top row
        *row = ROW_0;
    else if(y >= LOWER_HORIZ_LINE) //check if touch was in bottom row
        *row = ROW_2;
    else //check if touch was in middle row
        *row = ROW_1;

    //sets column to vertical third of the board the touch was in
    if(x <= LEFT_VERT_LINE) //check if touch was in left column
        *column = COLUMN_0;
    else if(x >= RIGHT_VERT_LINE) //check if touch was in right column
        *column = COLUMN_2;
    else //check if touch was in middle column
        *column = COLUMN_1;
}

// Runs a test of the display. Does the following.
// Draws the board. Each time you touch one of the screen areas, the screen will paint
// an X or an O, depending on whether switch 0 (SW0) is slid up (O) or down (X).
// When BTN0 is pushed, the screen is cleared. The test terminates when BTN1 is pushed.
void ticTacToeDisplay_runTest(){
    switches_init(); //must initialize switches
    buttons_init();  //must initialize buttons
    ticTacToeDisplay_init(); //call to initialize game board
    while(RUNNING){
        uint32_t buttons = buttons_read(); //used to get which of the 4 buttons are being
touched

        if((buttons & BUTTONS_BTN0_MASK) == BUTTONS_BTN0_MASK) { //we want to reset the
```

ticTacToeDisplay.c

```
game if BTN0 was pressed
    display_fillScreen(DISPLAY_BLACK); // Blank the screen.
    ticTacToeDisplay_drawBoardLines(); //redraw the game board
}
else if((buttons & BUTTONS_BTN1_MASK) == BUTTONS_BTN1_MASK) break; //if BTN1 is
pressed we reset
    if(display_isTouched()){ //check for touch

        uint8_t row, column;
        utils_msDelay(TOUCH_DELAY); //needed delay after touch is detected
        ticTacToeDisplay_touchScreenComputeBoardRowColumn(&row, &column); //call to
figure out row and column to put O/X in based on touch coordinates
        if(switches_read() & SWITCHES_SW0_MASK)ticTacToeDisplay_drawO(row, column,
false); //if SW0 is in the up position, we draw an O
        else ticTacToeDisplay_drawX(row, column, false); //if the switch is down,
draw an X
    }
}
display_fillScreen(DISPLAY_BLACK); // Blank the screen.
display_setTextSize(TEXT_SIZE); //set text size for final message
display_setTextColor(DISPLAY_GREEN); //final message will be green
display_println(TEST_OVER); //print final message
}

// This will draw the four board lines.
void ticTacToeDisplay_drawBoardLines() {
    display_drawLine(LEFT_VERT_LINE, Y_0, LEFT_VERT_LINE, DISPLAY_HEIGHT,
DISPLAY_YELLOW); //draw left vertical line
    display_drawLine(RIGHT_VERT_LINE, Y_0, RIGHT_VERT_LINE, DISPLAY_HEIGHT,
DISPLAY_YELLOW); //draw right vertical line
    display_drawLine(X_0, UPPER_HORIZ_LINE, DISPLAY_WIDTH, UPPER_HORIZ_LINE,
DISPLAY_YELLOW); //draw upper horizontal line
    display_drawLine(X_0, LOWER_HORIZ_LINE, DISPLAY_WIDTH, LOWER_HORIZ_LINE,
DISPLAY_YELLOW); //draw lower horizontal line
}
```