```c
/*
 * interface.c
 * Taylor Cowley and Andrew Okazaki
 */

#include <stdio.h>
#include <stdint.h>
#include "platform.h"
#include "xparameters.h"
#include "xaxivdma.h"
#include "xio.h"
#include "time.h"
#include "unistd.h"
#define TANK_HEIGHT 8
#define GREEN 0x0000FF00
#define GAME_X 320          // How wide our game screen is
#define LINE_Y 225          // Where the line at the bottom goes

#define EXTRA_TANK_0 250    // X coordinate of extra tanks
#define EXTRA_TANK_1 270    // X coordinate of extra tanks
#define EXTRA_TANK_2 290    // X coordinate of extra tanks
#define EXTRA_TANK_Y_OFFSET 5   // How far down the extra tanks are

// Packs each horizontal line of the figures into a single 32 bit word.
#define packword15(b14,b13,b12,b11,b10,b9,b8,b7,b6,b5,b4,b3,b2,b1,b0)  \
((b14 << 14) | (b13 << 13) | (b12 << 12) | (b11 << 11) | (b10 << 10) | \
 (b9  << 9 ) | (b8  << 8 ) | (b7  << 7 ) | (b6  << 6 ) | (b5  << 5 ) | \
 (b4  << 4 ) | (b3  << 3 ) | (b2  << 2 ) | (b1  << 1 ) | (b0  << 0 ) )

 // This seems like a *very bad* way to store the tank data, but this is what
 // we are doing for the moment.
static const int tank_15x8[TANK_HEIGHT] =
{
packword15(0,0,0,0,0,0,0,1,0,0,0,0,0,0,0),
packword15(0,0,0,0,0,0,1,1,1,0,0,0,0,0,0),
packword15(0,0,0,0,0,0,1,1,1,0,0,0,0,0,0),
packword15(0,1,1,1,1,1,1,1,1,1,1,1,1,1,0),
packword15(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),
packword15(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),
packword15(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),
packword15(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)
};

#define WORD_WIDTH 15

// --------------------------------------------
// Our declaration of functions to be used
void interface_draw_pixel(uint32_t * framePointer,uint32_t row,uint32_t col,uint32_t color);
// Ending declaration of internal functions
// --------------------------------------------

// This is 100% copied from aliens.c. Eventually it needs to move to its own global file
void interface_draw_pixel(uint32_t * framePointer,uint32_t row,uint32_t col,uint32_t color){
    #define DRAW_PIXEL_ROW_MULTIPLIER 1280  // 640 * 2 for screen doubling
    #define DRAW_PIXEL_ROW 640              // one row offset
    #define DRAW_PIXEL_DOUBLE 2             // for doubling

    // We draw 4 pixels for every 1 small-screen pixel
    framePointer[row*DRAW_PIXEL_ROW_MULTIPLIER + col*DRAW_PIXEL_DOUBLE] = color;
    framePointer[row*DRAW_PIXEL_ROW_MULTIPLIER + col*DRAW_PIXEL_DOUBLE+1] = color;
    framePointer[row*DRAW_PIXEL_ROW_MULTIPLIER+DRAW_PIXEL_ROW+ col*DRAW_PIXEL_DOUBLE] = color;
    framePointer[row*DRAW_PIXEL_ROW_MULTIPLIER+DRAW_PIXEL_ROW+ col*DRAW_PIXEL_DOUBLE + 1] = color;

}

// This draws the green line at the bottom of the screen
void interface_draw_line(uint32_t * framePointer){
    int row, col;                           // Initialize
    row = LINE_Y;                           // variables
    for(col=0;col<GAME_X;col++){            // Go along the screen and draw
        interface_draw_pixel(framePointer, row, col, GREEN);
    }
}

// This draws the extra tanks to the screen
void interface_draw_tanks(uint32_t * framePointer){
     int row, col;                          // Init loop vars
     for(row=0;row<TANK_HEIGHT;row++){      // Go through width
         for(col=0;col<WORD_WIDTH;col++){   // and height
             if((tank_15x8[row] & (1<<(WORD_WIDTH-col-1)))) {   // and draw 3 tanks
                 interface_draw_pixel(framePointer, row+EXTRA_TANK_Y_OFFSET, col+EXTRA_TANK_0, GREEN);
                 interface_draw_pixel(framePointer, row+EXTRA_TANK_Y_OFFSET, col+EXTRA_TANK_1, GREEN);
                 interface_draw_pixel(framePointer, row+EXTRA_TANK_Y_OFFSET, col+EXTRA_TANK_2, GREEN);
             }
         }
     }
}

```