

mother_ship.c

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

#include "mother_ship.h"
#include "sound/sound.h"
#include "interface.h" // enables update score
#include "util.h"

// Hard-coded definition for what the mother ship looks like
#define packword16(b15,b14,b13,b12,b11,b10,b9,b8,b7,b6,b5,b4,b3,b2,b1,b0) \
    ((b15<<15)|(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 ) )
static const uint32_t MOTHER_SHIP_16x7[] ={
    packword16(0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0),
    packword16(0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0),
    packword16(0,0,1,1,1,1,1,1,1,1,1,1,1,1,0,0),
    packword16(0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0),
    packword16(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1),
    packword16(0,0,1,1,1,0,0,1,1,0,0,1,1,1,0,0),
    packword16(0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0)};

#define MOTHER_SHIP_ROW 22 // Where the mother ship spawns at
#define MOTHER_SHIP_HEIGHT 7 // Mother ship is this tall
#define MOTHER_SHIP_WIDTH 16 // Mother ship is this wide
#define MOTHER_SHIP_MOVE_SPEED 2 // Mother ship moves this many pixels each
#define SCREEN_WIDTH 320 // Screen is 320 pixels wide
#define SHOCKING_PINK 0xFF6FFF // We want a cool color mother ship
#define MOTHER_SHIP_COLOR SHOCKING_PINK
#define BLACK 0x0 // Black color for erasing
#define BLINKING_TIMES 10 // How long we want the score to stay
#define MOTHER_SHIP_START_POINTS 123
#define MOTHER_SHIP_RAND_POINTS_MOD 500

struct{ // Defines our mother ship
    uint32_t row; // Lives at a certain row
    uint32_t col; // Lives at a certain column
    bool alive; // Is she alive?
}mother_ship;

uint32_t mother_ship_points=MOTHER_SHIP_START_POINTS;// Initial value of points for
killing her
uint32_t * frame; // The variable to write pixels to the screen
bool blinking; // Whether the score of dead mother ship is.

// Initializes the mother ship
void mother_ship_init(uint32_t * framePointer){
    blinking = false; // Death score isn't there
    frame = framePointer; // Store the screen frame
    mother_ship.row = MOTHER_SHIP_ROW; // She lives at this row
    mother_ship.alive = false; // She is not yet alive
    mother_ship.col = 0; // She spawns at left of screen
}

// Shows the points after a successful mother ship kill
```

mother_ship.c

```

void mother_ship_points_blink(){
    if(!blinking)
        return;                                // If not blinking, don't go

    static uint32_t times_blink = 0;            // We blink for a time
    times_blink++;                               // Which counts up

    if(times_blink > BLINKING_TIMES){           // If we have displayed enough
        times_blink = 0;                        // Reset timer and erase it.
        interface_alien_ship_points(mother_ship_points, mother_ship.col, true);
        blinking = false;                       // And we aren't running no more
    }
}

// Spawns a mother ship
void mother_ship_spawn(){
    if(mother_ship.alive)                       // Can't spawn when alive!
        return;
    // Erases any previously-drawn points
    interface_alien_ship_points(mother_ship_points, mother_ship.col, true);
    mother_ship.col = 0;                        // Spawns at left
    mother_ship.alive = true;                   // She is now alive
    mother_ship_draw(MOTHER_SHIP_COLOR);       // Draw her.
    blinking = false;                           // No score blinking anymore
}

// Moves the mother ship right
void mother_ship_move(){
    if(mother_ship.alive){                     // Can't move when dead!
        mother_ship_draw(BLACK);                // Erase old version
        mother_ship.col += MOTHER_SHIP_MOVE_SPEED; // Move her
        if(mother_ship.col > SCREEN_WIDTH-MOTHER_SHIP_WIDTH){ // She left.
            mother_ship.alive = false;           // So is now dead
            mother_ship.col = SCREEN_WIDTH;       // And off the screen
            sound_motherShipStop();
            return;                               // Exit
        }
        mother_ship_draw(MOTHER_SHIP_COLOR);    // Draw her!
        sound_init_motherShipLow();              // play high pitch sound
    }
}

// Detects a bullet collision on the mother ship
bool mother_ship_detect_collision(uint32_t row, uint32_t col){
    // If it is at the right row and in-between her columns
    if(row == mother_ship.row+MOTHER_SHIP_HEIGHT
        && col>mother_ship.col && col < mother_ship.col+MOTHER_SHIP_WIDTH){
        sound_motherShipStop();
        sound_init_motherShipHigh();
        mother_ship_points = rand()%MOTHER_SHIP_RAND_POINTS_MOD +
MOTHER_SHIP_START_POINTS;                      // Make random point
        interface_increment_score(mother_ship_points); // Player gets points
        mother_ship.alive = false;                // She dies
    }
}

```

mother_ship.c

```
    mother_ship_draw(BLACK);                                // and gets erased
    // Her points get drawn
    interface_alien_ship_points(mother_ship_points, mother_ship.col, false);
    blinking = true;                                        // drawing her points
    return true;                                           // We hit something!
}
return false;                                           // nope, not hit.
}

// Draws the mother ship
void mother_ship_draw(uint32_t color){
    int r, c;
    for(r=0;r<MOTHER_SHIP_HEIGHT;r++){                    // Go through width
        for(c=0;c<MOTHER_SHIP_WIDTH;c++){                // and height
            if((MOTHER_SHIP_16x7[r] & (1<<(MOTHER_SHIP_WIDTH-c-1)))){ //draw ship
                util_draw_pixel(frame,r+mother_ship.row,c+mother_ship.col,color);
            }
        }
    }
}
```