```
/*
* PongHockey.c
*
* Created: 2018-04-24 11:15:42
* Author : Axel Berglund och Carl-Johan Arvidsson
*/

#include <avr/io.h>
#define F_CPU 1000000UL
#include <util/delay.h>
#include <time.h>
#include <stdlib.h>


#define RW_H        PORTD |= (1<<PD3); //R/W 1

#define RW_L        PORTD &= ~(1<<PD3); //R/W 0


#define RS_H        PORTD |= (1<<PD2); //D/I 1

#define RS_L        PORTD &= ~(1<<PD2); //D/I 0


#define E_H                 PORTD |= (1<<PD4); //E 1

#define E_L                 PORTD &= ~(1<<PD4); //E 0


#define RED_ON              PORTD |=(1<<PD5) //Red lamp on

#define RED_OFF             PORTD &= ~(1<<PD5) //Red lamp off

#define GREEN_ON            PORTD |=(1<<PD6) //Green lamp on

#define GREEN_OFF           PORTD &= ~(1<<PD6) //Green lamp off

#define RIGHT_JS_PRESSED            PINA & (1<<PA5) //Right joystick pressed
#define LEFT_JS_PRESSED             PINA & (1<<PA2) //Right joystick pressed
#define BUTTON_PRESSED              PIND & (1<<PD7) //Button pressed

#define UP 0 //Directions when moving paddle
#define DOWN 1
#define LEFT 2
#define RIGHT 3
#define NONE 4

#define DEFAULT_PADDLE_SIZE 6 //Size of paddle (max 8)
#define DEFAULT_BALL_SIZE 2 //Size of ball
#define DEFAULT_MAX_SCORE 3 //Max score
#define PADDLE_DELAY 50 //Time between paddle updates/joystick readings (ms)
#define BALL_DELAY 25 //Time between ball updates (ms)

#define START_GAME 0
#define SETTINGS 1
#define QUIT 2
#define MENU_Y 40
```

```
/*
*           Sets ball size
*/
int ball_size( int change){ static int size =
            DEFAULT_BALL_SIZE; size +=
            change; if(size > 3){
                        size = 3;
            } if(size < 1){ size
            = 1;
            }
            return size;
}

/*
*           Sets paddle size
*/
int paddle_size( int change){ static int size =
            DEFAULT_PADDLE_SIZE;
            size += change;
            if(size > 8){ size =
            8;
            } if(size < 5){ size
            = 5;
            }
            return size;
}

int max_score(int change){ static int score =
            DEFAULT_MAX_SCORE; score +=
            change; if(score > 5) { score = 5;
            }
            if(score < 2) { score
                        = 2;
            }
            return score;
}

/*
*           Sets x address of LCD
*/
void set_x_address(int x){
            RW_L;
            RS_L;
            PORTB = 0b10111000 + x;
            E_H;
            E_L;
}

/*
*           Sets y address of LCD
*/
void set_y_address(int y) {
            RW_L;
            RS_L;
            PORTB = 0b01000000 + y;
            E_H;
            E_L;
```

```
}

/*
*        Enables chip2 (left side)
*/
void chip_2(){
        PORTD &= ~(1<<PD0);
        PORTD |= (1<<PD1);
}

/*
*        Enables chip1 (left side)
*/
void chip_1(){
        PORTD &= ~(1<<PD1);
        PORTD |= (1<<PD0);
}

/*
*        Turns on display
*/
void display_on(){
        E_H;
        RW_L; //RW 0
        RS_L //RS 0
        PORTC |= (1<<PC0); //Reset 1 PORTB
        = 0b00111111; //Display on
        chip_1();
        E_H; E_L;
        chip_2();
        E_H;
        E_L;
}

/*
*        Turns off display
*/
void display_off(){
        E_H;
        RW_L; //RW 0
        RS_L //RS 0
        PORTC |= (1<<PC0); //Reset 1 PORTB
        = 0b00111110; //Display on
        chip_1();
        E_H; E_L;
        chip_2();
        E_H;
        E_L;
}

/*
*        Draws data at x and y
*/
void draw_data(int page, int y, uint8_t
        data){ if(y<64){ chip_1(); } else {
                chip_2();
                y -= 64;
                }
                set_x_address(page); set_y_address(y);
                RW_L;
                RS_H;
                PORTB = data;
                E_H;
                E_L;
}

/*
*               Clears display
*/
 void clear_display(){ for(int x=0;x<8;x++){ for(int
                     y=0;y<128;y++){ draw_data(x,y,0);
                                }
                }
}

/*
*               Draw arrow
*/
void draw_arrow(int page, int y){
                uint8_t data = 0b00011100;
                draw_data(page,y,data);
                y++;
                data = 0b00010100; for(int
                i=0;i<12;i++){ draw_data(page, y,
                data);
                                y++;
                }
                data = 0b00110110;
                draw_data(page, y, data);
                y++; data = 0b01000001;
                draw_data(page, y, data);
                y++; data = 0b01100011;
                draw_data(page, y, data);
                y++; data = 0b00110110;
                draw_data(page, y, data);
                y++; data = 0b00011100;
                draw_data(page, y, data);
                y++; data = 0b00001000;
                draw_data(page, y, data);
                y++;
}

/*
*               Draw "start"
*/
void draw_start(int y, int page){
                //S uint8_t data =
                0b00101110;
                draw_data(page,y,data);
                y++;
                for(int i=0;i<3;i++){ data =
                                0b00101010;
                                draw_data(page,y,data);
                                y++;
                }
                data = 0b00111010; draw_data(page,y,data);
```

```
        y+=2;
         //T
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                 y++;
        }
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                 y++;
        }
        y++;

        //A
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<3;i++){ data =
                0b00001010;
                draw_data(page,y,data);
                 y++;
        }
        data = 0b00111110; draw_data(page,y,data);
        y+=2;

        //R data = 0b00111110;
        draw_data(page,y,data);
        y++;
        for(int i=0;i<2;i++){ data =
                0b00001010;
                draw_data(page,y,data);
                 y++;
        }
        data = 0b00011010;
        draw_data(page,y,data);
        y++; data =
        0b00101100;
        draw_data(page,y,data);
        y+=2;
         //T
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                 y++;
        }
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                 y++;
        }
}

/*
*       Draw "settings"
*/
```

```
void draw_settings(int y, int page){
        //S uint8_t data =
        0b00101110;
        draw_data(page,y,data);
        y++;
        for(int i=0;i<3;i++){ data =
                0b00101010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111010; draw_data(page,y,data);
        y+=2;

        //E
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<2;i++){ data =
                0b00101010;
                draw_data(page,y,data);
                y++;
        }
        for(int i=0;i<2;i++){ data =
                0b00100010;
                draw_data(page,y,data);
                y++;
        }
        y++;
         //T
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                y++;
        }
        y++;
         //T
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                y++;
        }
        y++;
         //I
        for(int i=0;i<2;i++){ data =
                0b00100010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<2;i++){ data =
                0b00100010;
                draw_data(page,y,data);
                y++;
        }
        y++;

        //N data = 0b00111110;
        draw_data(page,y,data);
        y++; data =
        0b00000100;
        draw_data(page,y,data);
        y++; data =
        0b00001000;
        draw_data(page,y,data);
        y++; data =
        0b00010000;
        draw_data(page,y,data);
        y++; data =
        0b00111110;
        draw_data(page,y,data);
        y+=2;

        //G data = 0b00111110;
        draw_data(page,y,data);
        y++; data =
        0b00100010;
        draw_data(page,y,data);
        y++;
        for(int i=0;i<2;i++){ data =
                0b00101010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111010; draw_data(page,y,data);
        y+=2;

        //S
        data = 0b00101110;
        draw_data(page,y,data); y++;
        for(int i=0;i<3;i++){ data =
                0b00101010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111010;
        draw_data(page,y,data);
    }


    /*
    *       Draw "quit"
    */
    void draw_quit(int y, int page){ //Q
        uint8_t data =
        0b00111110;
```

```
        draw_data(page,y,data);
        y++; data = 0b00100010;
        draw_data(page,y,data);
        y++; data = 0b01110010;
        draw_data(page,y,data);
        y++; data = 0b00100010;
        draw_data(page,y,data);
        y++; data = 0b00111110;
        draw_data(page,y,data);
        y+=2;

        //U data = 0b00111110;
        draw_data(page,y,data);
        y++;
        for(int i=0;i<3;i++){ data =
                0b00100000;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111110; draw_data(page,y,data);
        y+=2;
         //I
        for(int i=0;i<2;i++){ data =
                0b00100010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<2;i++){ data =
                0b00100010;
                draw_data(page,y,data);
                y++;
        }
        y++;
         //T
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<2;i++){ data =
                0b00000010;
                draw_data(page,y,data);
                y++;
        }
}

/*
*       Draws the main menu options
*/
void draw_menu(int y){
        draw_start(y, 2);
        draw_settings(y, 3);
        draw_quit(y, 4);
}
```

```
/*
*       Sets ports, turns on display and clears
*/
void init(){
        DDRA = 0x00; //Port A input
        DDRB = 0xFF; //Port B output
        DDRD = 0xFF; //Port D output
        DDRD &= ~(1<<PD7); //PD7 input DDRC
        |= (1<<PC0); //Reset (PC0) output
        display_on();
        clear_display();
}

/*
*       Initiate ADC
*/
void adc_init(){
        ADMUX |= (1 << REFS0);
        ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0); //division factor 128
}

/*
*       Convert analog to digital (joysticks)
*/
uint16_t adc_read(uint8_t channel){
        ADMUX &= 0b11111000; //Clear last 3 bits
        ADMUX |= channel; //Last 3 bits to correct channel ADCSRA |=
        (1<<ADSC); //Start conversion while(ADCSRA & (1<<ADSC));
        //Wait until conversion complete
        return (ADC); //Return result
}


/*
*       Fixes the border, if paddles/ball are close
*/
void fix_border(int page, uint8_t
        *data){ //Top / bottom edges
        if(page == 7){
                *data |= (1<<7) | (1<<6);
        }

        if(page == 0){
                *data |=(1<<0) | (1<<1);
        }
}

/*
*       Draws the ball, top left at (x,y)
*/
void draw_ball(int x, int y){ int page = x/8; int top =
        x%8; if(top > (ball_size(0)-1)){ //Pixels
        on same page uint8_t data = 0x00;
        fix_border(page, &data); for(int
        i=0;i<ball_size(0);i++){ data |=
        (1<<(top-i));
                }
        for(int j=0;j<ball_size(0);j++){
            draw_data(page,y+j,data);
        }
        } else {
        //Pixels on different pages

        //First page uint8_t data
        = 0x00;
        fix_border(page,&data);
        for(int a=0;a<=top;a++){
        data |= (1<<a);
        }
        for(int b=0;b<ball_size(0);b++){
            draw_data(page,y+b,data);
        }
        //Second page data = 0x00;
        fix_border(page-1,&data); for(int c = 0;
        c<(ball_size(0)-top-1);c++){ data |=
        (1<<(7-c));
        }
        for(int d=0;d<ball_size(0);d++){ draw_data(page-
            1,y+d,data);
        }
    }
}


/*
*   Clears the ball (call function before drawing ball to remove previous)
*/
void clear_ball(int x, int y) {
    int page = x/8;
    int top = x%8;
    if(top > (ball_size(0)-1)){ //Pixels on same
        page uint8_t data = 0x00;
        fix_border(page, &data); for(int
        j=0;j<ball_size(0);j++){
        draw_data(page,y+j,data);
        }
        } else {
        //Pixels on different pages

        //First page uint8_t data = 0x00;
        fix_border(page,&data); for(int
        b=0;b<ball_size(0);b++){
        draw_data(page,y+b,data);
        }
        //Second page data = 0x00;
        fix_border(page-1,&data); for(int
        d=0;d<ball_size(0);d++){
        draw_data(page-1,y+d,data);
        }
    }
}


/*
*   Fixes so the paddles does not disappear when below the ball
*/
```

```
void fix_ball(int paddle_page, int paddle_y, int ball_x, int ball_y, uint8_t
        *data){ int y = 0; for(int i=0;i<ball_size(0);i++){ if(paddle_y ==
        ball_y+i){ y = paddle_y;
                }
        } if(y!=0){ int page = ball_x/8;
        int top = ball_x%8;
                if(top > (ball_size(0)-1)){ //Pixels on same page
                        uint8_t data = 0x00; if(page ==
                        paddle_page){ fix_border(page, &data);
                        for(int i=0;i<ball_size(0);i++){ data |=
                        (1<<(top-i));
                                }
                        }
                        } else {
                        //Pixels on different pages

                        //First page uint8_t data = 0x00;
                        if(paddle_page == page){
                        fix_border(page,&data); for(int
                        a=0;a<=top;a++){ data |= (1<<a);
                                }
                        }
                        //Second page data = 0x00; if(paddle_page ==
                        page-1){ fix_border(page-1,&data); for(int c = 0;
                        c<(ball_size(0)-top-1);c++){ data |= (1<<(7-c));
                                }
                        }
                }
        }
}

/*
*       Draws the paddle with top pixel at (x,y)
*/
void draw_paddle(int x, int y, int direction, int ball_x, int
        ball_y){ int page = x/8; //Calculates page int top =
        x%8; //Calculates lowest pixel of paddle if(direction
        == LEFT) {
                //Clears previous paddle and fixes border

                uint8_t data = 0x00; fix_border(page-
                1,&data); fix_ball(page-1, y+1, ball_x,
                ball_y, &data); draw_data(page-
                1,y+1,data);

                data = 0x00; fix_border(page,&data);
                fix_ball(page, y+1, ball_x, ball_y,
                &data); draw_data(page,y+1,data);

        }
        if(direction == RIGHT) {
                //Clears previous paddle and fixes border

                uint8_t data = 0x00; fix_border(page-
                1,&data); fix_ball(page-1, y-1, ball_x,
                ball_y, &data);
                draw_data(page-1,y-1,data);
                data = 0x00; fix_border(page,&data);
                fix_ball(page, y-1, ball_x, ball_y,
                &data); draw_data(page,y-1,data);
        }
        if(top ==paddle_size(0) && direction == UP) {
                //Clears the bottom pixel, left on a different page and fixes
                border uint8_t data = 0x00; fix_border(page-1,&data);
                fix_ball(page-1, y, ball_x, ball_y, &data); draw_data(page-
                1,y,data);
        }

        if(top == 7 && direction == DOWN){
                //Clears top pixel (different page), fixes
                border uint8_t data = 0x00;
                fix_border(page+1,&data); fix_ball(page+1, y,
                ball_x, ball_y, &data);
                draw_data(page+1,y,data);//Clear
        }

        //Calculates where the paddle should be and draws a paddle_size pixel high paddle (max 8)
        if(top>(paddle_size(0) -1)){ //All pixels on same page uint8_t data = 0x00; for(int
        i=0;i<paddle_size(0) ;i++){ data |= (1<<(top-i));
                }
                fix_border(page,&data); fix_ball(page,
                y, ball_x, ball_y, &data);
                draw_data(page, y, data);
                } else {
                //Some pixels on different
                page uint8_t data = 0x00;
                for(int i=0; i<=top;i++){
                        //Calculates pixels on first page
                        data |= (1<<i);
                }
                //Draws pixels on first page
                fix_border(page,&data); fix_ball(page,
                y, ball_x, ball_y, &data);
                draw_data(page, y, data); data =
                0x00;
                for(int j=0;j<(paddle_size(0)-top-1);j++){
                        //Calculates pixels on second page
                        data |= (1<<(7-j));
                }
                //Draws pixels on second page
                fix_border(page-1,&data); fix_ball(page-
                1, y, ball_x, ball_y, &data);
                draw_data(page-1,y,data);
        }
}


/*
*   Draws the board (surrounding edges) of the game
*/
void draw_board(){ //Sides for(int
        x=0;x<=2;x++){ draw_data(x,0,0xFF);
        draw_data((7-x),0,0xFF);
        draw_data(x,127,0xFF);
        draw_data((7-x),127,0xFF);
        draw_data(x,1,0xFF); draw_data((7-
        x),1,0xFF); draw_data(x,126,0xFF);
```

```
                draw_data((7-x),126,0xFF);
        }
        //Top/bottom for(int y=2;y<126;y++){
        draw_data(0,y,0b00000011);
        draw_data(7,y,0b11000000);
        }
}

/*
*       Initiates a new game
*/
void init_game(){
        clear_display();
        draw_board();
}

/*
*       Draws "0"
*/
void draw_0(int page, int y){ uint8_t
        data = 0b00111110;
        draw_data(page,y,data);
        y++;
        for(int i=0;i<3;i++){ data =
                0b00100010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00111110;
        draw_data(page,y,data);
}

/*
*       Draws "1"
*/
void draw_1(int page, int y){ y++;
        uint8_t data =
        0b00000010;
        draw_data(page,y,data);
        y++; data = 0b00111110;
        draw_data(page,y,data);
}

/*
*       Draws "2"
*/
void draw_2(int page, int y){ uint8_t
        data = 0b00111010;
        draw_data(page,y,data);
        y++;
        for(int i=0;i<3;i++){ data =
                0b00101010;
                draw_data(page,y,data);
                y++;
        }
        data = 0b00101110;
        draw_data(page,y,data);
}
```

```
/*
*           Draws "3"
*/
void draw_3(int page, int y){ uint8_t
            data = 0b00100010;
            draw_data(page,y,data);
            y++; data = 0b00100010;
            draw_data(page,y,data);
            y++;
            for(int i=0;i<2;i++){ data =
                        0b00101010;
                        draw_data(page,y,data);
                        y++;
            }
            data = 0b00111110;
            draw_data(page,y,data);
}

/*
*           Draws "4"
*/
void draw_4(int page, int y){ uint8_t
            data = 0b00001110;
            draw_data(page,y,data);
            y++;
            for(int i=0;i<3;i++){ data =
                        0b00001000;
                        draw_data(page,y,data);
                        y++;
            }
            data = 0b00111110;
            draw_data(page,y,data);
}

/*
*           Draws "5"
*/
void draw_5(int page, int y){ uint8_t
            data = 0b00101110;
            draw_data(page,y,data);
            y++;
            for(int i=0;i<3;i++){ data =
                        0b00101010;
                        draw_data(page,y,data);
                        y++;
            }
            data = 0b00111010;
            draw_data(page,y,data);
}

/*
*           Draws a line
*/
void draw_line(int page, int y){
            y+=2;
            uint8_t data = 0b00001000; for(int
            i=0;i<5;i++){
            draw_data(page,y,data);
                        y++;
```

```
        }
}

/*
*       Draws a number depending on score
*/
void draw_score(int score, int page, int
        y){ switch(score){ case 0:
        draw_0(page,y); break; case 1:
        draw_1(page,y); break; case 2:
        draw_2(page,y); break; case 3:
        draw_3(page,y); break; case 4:
        draw_4(page,y); break;
                case 5:
                draw_5(page,y);
                break;
        }
}

/*
*       Draws the score
*/
void scores(int green, int red){
        draw_score(green, 4, 40);
        draw_line(4,62);
        draw_score(red, 4, 80);
        _delay_ms(200);
        _delay_ms(200);
}

/*
*       Starts the game
*/
void game_start(){ init_game(); int x1 = 30; int
        y1 = 90; int x2 = 30; int y2 = 40; int
        ball_x = 34; int ball_y = 64; int
        score_red = 0; int score_green = 0;
        x1++; draw_paddle(x1,y1,UP,ball_x,
        ball_y); x2++; draw_paddle(x2,y2,UP,
        ball_x,ball_y);
        draw_ball(ball_x, ball_y);

        int delay = 0;

        int ball_dx = -1;
        int     ball_dy = -1;
        int r = rand() % 4; if(r ==
        0){ ball_dx = -1; ball_dy =
        -1; } else if(r == 1){
        ball_dx = -1; ball_dy = 1;
        } else if(r == 2){ ball_dx =
        1; ball_dy = 1; } else {
        ball_dx = 1;
                ball_dy = -1;
        }

        while (1) {

                //Check wall collision
                if((ball_dx > 0 && ball_x == 61) || (ball_dx < 0 && ball_x == (ball_size(0)+1))){
                    //Top/bottom collision
                    ball_dx *= -1;
                }

                if(!((ball_dy < 0 && ball_dx > 0 && ball_y <= 2 && ball_x >= 22+ball_size(0)&&ball_x<=38) || (ball_dx > 0 &&
ball_dy > 0 && ball_y >= 126 - ball_size(0) && ball_x >= 22+ball_size(0)&&ball_x<=38) ||
(ball_dx<0&&ball_dy<0&&ball_y<=2&&ball_x>=24+ball_size(0)&&ball_x<=40) || (ball_dx < 0 && ball_dy>0&&ball_y>=126-
ball_size(0)&&ball_x>=24+ball_size(0)&&ball_x<=40))){

                    if((ball_dy > 0 && ball_y == (126-ball_size(0))) || (ball_dy < 0 && ball_y == 2)){
                        //Sides collision
                        ball_dy *= -1;
                    }
                }

                //Check paddle collision
                if(ball_dy < 0) {
                    //Ball y decreasing
                    if(ball_y - y1 == 1 && ball_x - ball_size(0) + 1 <= x1 && ball_x >= x1 - paddle_size(0) + 1){
                        ball_dy *= -1;
                    }
                    if(ball_y - y2 == 1 && ball_x - ball_size(0) + 1 <= x2 && ball_x >= x2 - paddle_size(0) + 1){
                        ball_dy *= -1;
                    }

                    //"Diagonal" kollision
                    if(ball_dx > 0) { //X
                    increasing
                        if(ball_x == x1 - paddle_size(0) && ball_y == y1 + 1) {
                            ball_dx*=-1; ball_dy*=-
                            1;
                        }
                        if(ball_x == x2 - paddle_size(0) && ball_y == y2 + 1) {
                            ball_dx*=-1; ball_dy*=-
                            1;
                        }

                        if(ball_size(0) == 2 || 3){ if((ball_y == y1 && ball_x == x1 - paddle_size(0)) || (ball_y == y2
                            && ball_x == x2
- paddle_size(0))){
                        ball_dx *= -1;
                            }
                        }
                        if(ball_size(0) == 3){

                            if((ball_y == y1-1 && ball_x == x1 - paddle_size(0)) || (ball_y == y2-1 && ball_x
== x2 - paddle_size(0))){
                        ball_dx *= -1;
                            }
                        }

                        } else {
                        //X decreasing
                        if(ball_x == x1 + ball_size(0) && ball_y == y1 + 1){
                            ball_dx *= -1;
                            ball_dy *= -1;
                        }
                        if(ball_x == x2 + ball_size(0) && ball_y == y2 + 1){
                            ball_dx *= -1;
                            ball_dy *= -1;
                        }

                        if(ball_size(0) == 2 || 3){ if((ball_y == y1 && ball_x == x1+ball_size(0))||(ball_y ==
                            y2 && ball_x ==
x2+ball_size(0))){
                        ball_dx *= -1;
                            }
                        }
                        if(ball_size(0) == 3){ if((ball_y == y1-1 && ball_x == x1+ball_size(0))||(ball_y == y2-1
                            && ball_x ==
x2+ball_size(0))){
                        ball_dx *= -1;
                            }
                        }
                    }
                    } else {
                    //Ball y increasing
                    if(ball_y + ball_size(0) == y1 && ball_x - ball_size(0) + 1 <= x1 && ball_x >= x1 - paddle_size(0) + 1)
                        { ball_dy *= -1;
                    }
                    if(ball_y + ball_size(0) == y2 && ball_x - ball_size(0) + 1 <= x2 && ball_x >= x2 - paddle_size(0) + 1)
                        { ball_dy *= -1;
                    }

                    //Diagonal kollision
                    if(ball_dx > 0) { //X
                    increasing
                        if(ball_x == x1 - paddle_size(0) && ball_y == y1 - ball_size(0)){
                            ball_dx *= -1;
                            ball_dy *= -1;
                        }
                        if(ball_x == x2 - paddle_size(0) && ball_y == y2 - ball_size(0)){
                            ball_dx *= -1;
                            ball_dy *= -1;
                        }

                        if(ball_size(0) == 2||3){ if((ball_y == y1 -1 && ball_x == x1-paddle_size(0)) || (ball_y == y2-1
                            && ball_x ==
x2-paddle_size(0))){
                        ball_dx *= -1;
                            }
                        }
                        if(ball_size(0) == 3){ if((ball_y == y1-2 &&ball_x == x1-paddle_size(0))||(ball_y == y2-2
                            &&ball_x ==
x2-paddle_size(0))){
                        ball_dx *= -1;
                            }
                        }

                        } else {
                        //X decreasing
                        if(ball_x == x1 + ball_size(0) && ball_y == y1 - ball_size(0)){
                            ball_dx *= -1;
                            ball_dy *= -1;
                        }
                        if(ball_x == x2 + ball_size(0) && ball_y == y2 - ball_size(0)){
                            ball_dx *= -1;
                            ball_dy *= -1;
                        }

                        if(ball_size(0) == 2 || 3){ if((ball_y == y1-1 && ball_x == x1+ball_size(0)) ||(ball_y == y2-
                            1 && ball_x ==
x2+ball_size(0)) ){
                        ball_dx *= -1;
                            }
                        }
                        if(ball_size(0) == 3){ if((ball_y == y1-2 && ball_x == x1+ball_size(0)) ||(ball_y == y2-2
                            && ball_x ==
x2+ball_size(0)) ){
                        ball_dx *= -1;
                            }
                        }



                    }
                }

                //Move ball & draw if((delay %
                BALL_DELAY) == 0){
                clear_ball(ball_x, ball_y); if(ball_y !=
                126-ball_size(0)){
                        clear_ball(ball_x, ball_y+1);
                    }
                    if(ball_y != 2){
                        clear_ball(ball_x,ball_y-1);
                    }
                    ball_x += ball_dx; ball_y += ball_dy;
                    draw_ball(ball_x, ball_y);
                    draw_paddle(x1,y1,NONE,ball_x,ball_y)
                    ;
                    draw_paddle(x2,y2,NONE,ball_x,ball_y)
                    ;

                }

                //Check paddle collision
                if(ball_dy < 0) {
                    //Ball y decreasing
                    if(ball_y - y1 == 1 && ball_x - ball_size(0) + 1 <= x1 && ball_x >= x1 - paddle_size(0) + 1){
                        ball_dy *= -1;
                        ball_y++;
                    }
                    if(ball_y - y2 == 1 && ball_x - ball_size(0) + 1 <= x2 && ball_x >= x2 - paddle_size(0) + 1){
                        ball_dy *= -1;
                        ball_y++;
                    }

                    //"Diagonal" kollision
                    if(ball_dx > 0) { //X
                    increasing
                        if(ball_x == x1 - paddle_size(0) && ball_y == y1 + 1) {
```

```
            ball_dx*=-1; ball_dy*=-
            1;
        }
        if(ball_x == x2 - paddle_size(0) && ball_y == y2 + 1) {
            ball_dx*=-1; ball_dy*=-
            1;
        }
        } else {
        //X decreasing
        if(ball_x == x1 + ball_size(0) && ball_y == y1 + 1){
            ball_dx *= -1;
            ball_dy *= -1;
        }
        if(ball_x == x2 + ball_size(0) && ball_y == y2 + 1){
            ball_dx *= -1;
            ball_dy *= -1;
        }
    }
    } else {
    //Ball y increasing
    if(ball_y + ball_size(0) == y1 && ball_x - ball_size(0) + 1 <= x1 && ball_x >= x1 - paddle_size(0) + 1)
    {
        ball_dy *= -1;
        ball_y--;
    }
    if(ball_y + ball_size(0) == y2 && ball_x - ball_size(0) + 1 <= x2 && ball_x >= x2 - paddle_size(0) + 1)
    {
        ball_dy *= -1;
        ball_y--;
    }

    //Diagonal kollision
    if(ball_dx > 0) { //X
    increasing
        if(ball_x == x1 - paddle_size(0) && ball_y == y1 - ball_size(0)){
            ball_dx *= -1;
            ball_dy *= -1;
        }
        if(ball_x == x2 - paddle_size(0) && ball_y == y2 - ball_size(0)){
            ball_dx *= -1;
            ball_dy *= -1;
        }
        } else {
        //X decreasing
        if(ball_x == x1 + ball_size(0) && ball_y == y1 - ball_size(0)){
            ball_dx *= -1;
            ball_dy *= -1;
        }
        if(ball_x == x2 + ball_size(0) && ball_y == y2 - ball_size(0)){
            ball_dx *= -1;
            ball_dy *= -1;
        }
    }
}

//Move paddles
if((delay % PADDLE_DELAY) == 0) {
        //Read joysticks every PADDLE_DELAY
        ms uint16_t adc_x1 = adc_read(3); uint16_t
        adc_y1 = adc_read(4); uint16_t adc_x2 =
        adc_read(0); uint16_t adc_y2 =
        adc_read(1); if(adc_x1>800){ if(x1<56){
        x1++;
                        draw_paddle(x1,y1,UP,ball_x,ball_y); //Move paddle up
                }
                } else if(adc_x1<200) {
                if(x1>paddle_size(0)+6){ x1--
                ;
                        draw_paddle(x1,y1,DOWN,ball_x,ball_y); //Move paddle down
                }
        }
        if(adc_y1>800){
                if(y1>66){ y1--
                ;
                        draw_paddle(x1,y1,LEFT,ball_x,ball_y); //Move paddle left
                }
                } else if(adc_y1<200)
                { if(y1<120){ y1++;
                        draw_paddle(x1,y1,RIGHT,ball_x,ball_y); //Move paddle right
                }
        }
        if(adc_x2>800){
                if(x2>paddle_size(0)+6){ x2--
                ;
                        draw_paddle(x2,y2,DOWN,ball_x,ball_y); //Move paddle 2 down
                }
                } else if(adc_x2<200)
                { if(x2<56){ x2++;
                        draw_paddle(x2,y2,UP,ball_x,ball_y); //Move paddle 2 up
                }
        }
        if(adc_y2>800){
                if(y2<62){
                y2++;
                        draw_paddle(x2,y2,RIGHT,ball_x,ball_y); //Move paddle 2 right
                }
                } else if(adc_y2<200) {
                if(y2>7){ y2--;
                        draw_paddle(x2,y2,LEFT,ball_x,ball_y); //Move paddle 2 left
                }
        }
}

//Checks button press
if(BUTTON_PRESSED){
        clear_display();
        return;
}


if(ball_y <= 1){
        clear_display();
        score_green++;
        RED_ON;
        scores(score_red,score_green);
        RED_OFF;
        if(score_green == max_score(0)){
            //Green won
            int blink_delay = 0;
            while(!(RIGHT_JS_PRESSED)){
                if(blink_delay%200 == 0){
                GREEN_ON;
                }
                if(blink_delay%200 == 100){
                GREEN_OFF;
                }
                _delay_ms(1);
                blink_delay++;
            }
            GREEN_OFF;
            clear_display();
            return;
        }
        init_game(); //Restart game x1 = 30;
        y1 = 90; x2 = 30; y2 = 40; ball_x =
        34; ball_y = 64; x1++;
        draw_paddle(x1,y1,UP,ball_x,ball_y)
        ; x2++;
        draw_paddle(x2,y2,UP,ball_x,ball_y)
        ;
        draw_ball(ball_x, ball_y);

        delay = 0; int r = rand() %
        4; if(r == 0){ ball_dx = -1;
        ball_dy = -1; } else if(r ==
        1){ ball_dx = -1; ball_dy
        = 1; } else if(r == 2){
        ball_dx = 1; ball_dy = 1; }
        else { ball_dx = 1;
            ball_dy = -1;
        }
    }
    if(ball_y >= 126){
        clear_display();
        GREEN_ON;
        score_red++;
        scores(score_red,score_green);
        GREEN_OFF;
        if(score_red == max_score(0)){
            //Red won
            int blink_delay = 0;
            while(!(RIGHT_JS_PRESSED)){
                if(blink_delay%200 == 0){
                    RED_ON;
                }
                if(blink_delay%200 == 100){
                    RED_OFF;
                }
                _delay_ms(1);
                blink_delay++;
            }
            RED_OFF;
            clear_display();
            return;
        }
```

```
            init_game(); //Restart game x1 = 30;
            y1 = 90; x2 = 30; y2 = 40; ball_x =
            34; ball_y = 64; x1++;
            draw_paddle(x1,y1,UP,ball_x,ball_y)
            ; x2++;
            draw_paddle(x2,y2,UP,ball_x,ball_y)
            ;
            draw_ball(ball_x, ball_y);

            delay = 0; int r = rand() %
            4; if(r == 0){ ball_dx = -1;
            ball_dy = -1; } else if(r ==
            1){ ball_dx = -1; ball_dy
            = 1; } else if(r == 2){
            ball_dx = 1; ball_dy = 1; }
            else { ball_dx = 1;
                ball_dy = -1;
            }
        }


        delay++;
        _delay_ms(1);
    }
}

/*
*       Draws "ball"
*/
void draw_settings_ball(int page, int y){
    //B uint8_t data =
    0b00111110;
    draw_data(page,y,data);
    y++;
    for(int i=0;i<3;i++){ data =
        0b00101010;
        draw_data(page,y,data);
        y++;
    }
    data = 0b00011100; draw_data(page,y,data);
    y+=2;

    //A
    data = 0b00111110;
    draw_data(page,y,data); y++;
    for(int i=0;i<3;i++){ data =
        0b00001010;
        draw_data(page,y,data);
        y++;
    }
    data = 0b00111110; draw_data(page,y,data);
    y+=2;
     //L
    data = 0b00111110;
    draw_data(page,y,data); y++;
    for(int i=0;i<4;i++){ data =
        0b00100000;
        draw_data(page,y,data);
        y++;
        }
        y++;
         //L
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<4;i++){ data =
                0b00100000;
                draw_data(page,y,data);
                 y++;
        }
}

/*
*       Draws "paddle"
*/
void draw_settings_paddle(int page, int y){
        //P
        uint8_t data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<3;i++){ data =
                0b00001010;
                draw_data(page,y,data);
                 y++;
        }
        data = 0b00001100;
        draw_data(page,y,data); y++;

        //A
        data = 0b00111110;
        draw_data(page,y,data); y++;
        for(int i=0;i<3;i++){ data =
                0b00001010;
                draw_data(page,y,data);
                 y++;
        }
        data = 0b00111110;
        draw_data(page,y,data); y+=2;

        //D data = 0b00111110;
        draw_data(page,y,data);
        y++;
        for(int i=0;i<3;i++){ data =
                0b00100010;
                draw_data(page,y,data);
                 y++;
        }
        data = 0b00011100;
        draw_data(page,y,data); y+=2;

        //D data = 0b00111110;
        draw_data(page,y,data);
        y++;
        for(int i=0;i<3;i++){ data =
                0b00100010;
                draw_data(page,y,data);
                 y++;
        }
        data = 0b00011100;
        draw_data(page,y,data); y+=2;
```

```
     //L
    data = 0b00111110;
    draw_data(page,y,data); y++;
    for(int i=0;i<4;i++){ data =
        0b00100000;
        draw_data(page,y,d
        ata);
        y++;
    }
    y++;

    //E
    data = 0b00111110;
    draw_data(page,y,data); y++;
    for(int i=0;i<2;i++){ data =
        0b00101010;
        draw_data(page,y,data);
        y++;
    }
    for(int i=0;i<2;i++){ data =
        0b00100010;
        draw_data(page,y,data);
        y++;
    }
}

/*
*       Draw "score"
*/
void draw_settings_score(int page, int y){
    //S uint8_t data =
    0b00101110;
    draw_data(page,y,data);
    y++;
    for(int i=0;i<3;i++){ data =
        0b00101010;
        draw_data(page,y,data);
        y++;
    }
    data = 0b00111010; draw_data(page,y,data);
    y+=2;

    //C data = 0b00111110;
    draw_data(page,y,data);
    y++;
    for(int i=0;i<4;i++){ data =
        0b00100010;
        draw_data(page,y,data);
        y++;
    }
    y++;

    //O data = 0b00111110;
    draw_data(page,y,data);
    y++;
    for(int i=0;i<3;i++){ data =
        0b00100010;
        draw_data(page,y,data);
        y++;
    }
    data = 0b00111110; draw_data(page,y,data);
    y+=2;

    //R data = 0b00111110;
    draw_data(page,y,data);
    y++;
    for(int i=0;i<2;i++){ data =
        0b00001010;
        draw_data(page,y,data);
        y++;
    }
    data = 0b00011010;
    draw_data(page,y,data);
    y++; data =
    0b00101100;
    draw_data(page,y,data);
    y+=2;

    //E
    data = 0b00111110;
    draw_data(page,y,data); y++;
    for(int i=0;i<2;i++){ data =
        0b00101010;
        draw_data(page,y,data);
        y++;
    }
    for(int i=0;i<2;i++){ data =
        0b00100010;
        draw_data(page,y,data);
        y++;
    }
    y++;


}

/*
*       Draws the settings menu
*/
void redraw_settings(int selected){
    clear_display(); int size =
    ball_size(0); for(int
    i=0;i<size*2;i+=2){ draw_data(2,
    80+i, 0xFF);
    }
    size = paddle_size(0); for(int
    i=0;i<size*2;i+=2){ draw_data(4,
    80+i, 0xFF);
    }
    size = max_score(0); for(int
    i=0;i<size*2;i+=2){
    draw_data(6,80+i, 0xFF);
    }
    draw_settings_ball(2,30); draw_settings_paddle(4,30);
    draw_settings_score(6,30);
    draw_arrow(2+2*selected, 5);
}
```

```
/*
*        Initializes the settings and draws the menu
*/ int settings_start(){ int
selected = 0;
        redraw_settings(selected);

          while(1){ uint16_t adc_x1 = adc_read(3);
                  uint16_t adc_y1 = adc_read(4);
            if(adc_x1>800){ //Up if(selected == 2){
                                selected = 0;
                                } else {
                                selected++;
                        }
                        redraw_settings(selected);
                        } else if(adc_x1<200) {
                        //Down if(selected ==
                        0){ selected = 2; } else
                        {
                                selected--;
                        }
                        redraw_settings(selected);
                }

                if(adc_y1 > 800){ //Left
                        if(selected ==
                        0){ //Ball
                                ball_size(-1);
                        }
                        if(selected == 1){
                                //Paddle
                                paddle_size(-1);
                        }
                        if(selected == 2){
                                //Score
                                max_score(-1);
                        }
                        redraw_settings(selected);
                        _delay_ms(100);
                        } else if(adc_y1 <
                        200){ //Right
                        if(selected == 0){ //Ball
                                ball_size(1);
                        }
                        if(selected == 1){
                                //Paddle
                                paddle_size(1);
                        }
                        if(selected == 2){
                                //Score
                                max_score(1);
                        }
                        redraw_settings(selected);
                        _delay_ms(100);
                }


                //Select
        if(LEFT_JS_PRESSED){
            clear_display();
            return 1;
        }
        _delay_ms(100);
    }
}

/*
*   Inits menu and draws it
*/
void menu_start(){
    clear_display();
    draw_menu(MENU_Y);
    int display = 1;
    display_off(); int selected = START_GAME;
    while(1){ uint16_t adc_x1 = adc_read(3);
    if(display == 0){ if(adc_x1>800){
    clear_display();
                draw_menu(MENU_Y);
                if(selected == QUIT){ selected =
                    START_GAME;
                    } else {
                    selected++;
                }
                } else if(adc_x1<200) {
                clear_display();
                draw_menu(MENU_Y);
                if(selected == START_GAME){
                selected = QUIT;
                    } else {
                    selected--;
                }
            }
            draw_arrow(2+selected, 5);
        }


        //Select
        if(RIGHT_JS_PRESSED && display == 0){
            if(selected == START_GAME){
                game_start(); //Start game
                draw_menu(MENU_Y);
            }

            if(selected == SETTINGS){
                while(settings_start() != 1);
                draw_menu(MENU_Y);
            }

            if(selected == QUIT){
                //Turn off display
                display_off();
                display = 1;
            }

        }
        if(BUTTON_PRESSED && display == 1){
            display_on();
            display = 0;
        }
        _delay_ms(100);
    }
}

/*
*   Start of application, inits ports and adc and starts main menu
*/
int main(void)
{
    srand(time(NULL));
    init(); adc_init();
    menu_start();
}
```