```
/*
 * MoleGame123.c
 *
 * Created: 2018-05-14 09:55:44
 * Author : ine15asc
 */

#include <avr/io.h>

#include <avr/io.h>


#define F_CPU         8000000UL

#define BTN_1                 0b01111101
#define BTN_2                 0b01111110
#define BTN_3                 0b01110111
#define BTN_4                 0b01101111
#define BTN_5                 0b01011111
#define BTN_6                 0b00111111

#define BT_1            0b11111101
#define BT_2            0b11111110
#define BT_3            0b11110111
#define BT_4            0b11101111
#define BT_5            0b11011111
#define BT_6            0b10111111

#define LIGHT_1     0b00000101
#define LIGHT_2     0b00000110
#define LIGHT_3     0b01000100
#define LIGHT_4     0b00001100
#define LIGHT_5     0b00010100
#define LIGHT_6     0b00100100

int highScoreNumber = 10000;
int lightOn = 0;
int lightTurnedOn = 1;
int rightButton = 0;
int score = 0;
int gameround = 0;
int buttonPressed = 0;
int currentTime = 0;
int pointCount = 0;
int endTime = 0;
int initiateGame = 1;
```

```
int playerOne = 0;
int playerTwo = 0;
int playerOnePoints = 0;
int playerTwoPoints = 0;
int gameMode = 0;
int a = 0;

volatile double count;
volatile double countClock;


//#include <asf.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <util/delay.h>


void setup_interrupt(void);
void setUpPorts(void);
void initTimers(void);


void randomLight(void);
void randomNumber(void);
void lightOff(void);
void allLightOn(void);


void display_setup(void);
void writeChar(char c);
void display_cmd(char c);
void writeString(char string[]);
void firstMessage();
void initiateGameMessage();
void clearDisplay();
void writePoints();
void twoPlayerRandomLight();
void twoPlayerDisplayScore();
void twoPlayerMode();
void twoPlayerTwoWritePoints();
void twoPlayerOneWritePoints();
void displayMenu();
void celebrateLight();
```

```
void highScore();
void checkButton();
void checkButtonTwo();

// **********************Port-setup*********************
void setUpPorts(){

	DDRB = 0b11111011;		// Sätter lamporna utåt
	lightOff();					//(pga vi kopplade tvärtom)
	DDRD = 0b10000000;			// Sätter knapparna inåt och interrupten inåt
(display, pinne 21 utåt)
	PORTB = 0xff;
	PORTD = 0b11111011;
	initTimers();			//ställer in timers
	setup_interrupt();		//ställer upp interrupt
	display_setup();

}

// **********************Display*********************
void display_setup(){
	DDRA = 0b11111111; // Sätter pinnarna för port A till ettor (ut)
	DDRC  |= _BV(PC0);
	DDRC  |= _BV(PD1); //sätter pinne 22-23 till ettor (D- är redan satt)
	_delay_ms(10);
	PORTC &= ~_BV(PC0);
	PORTC &= ~_BV(PC1); //RS 0 & RW 0

	// function set
	//PORTA= 0b00111111; // Sätter interface datalängd, antal linjer etc.
	display_cmd(0b00111111);
	_delay_ms(10);

	//Display on
	//PORTA = 0b00001111; // Inställningar för displayen
	display_cmd(0b00001111);
	_delay_ms(10);

	//Entry mode set
	//PORTA = 0b00000110;
	display_cmd(0b00000110);
	_delay_ms(10);
}

// **********************Timers*********************
void initTimers(){
```

```
	TCCR0 = TCCR0 | 0b00000011; // Prescaler 64 (Sida 85 i databladet.)
	TCCR2 = TCCR2 | 0b00000111; // Prescaler 1024. Sida 131 i databladet.
	TIMSK = TIMSK | 0b01000001; // Overflow interrupt aktivt. (Sida 83 i databladet.)
}

// **********************Interrupt*********************
void setup_interrupt(){
	GICR   |= (1<<INT0);
	MCUCR  |= (1<<ISC01) | (1<<ISC00);
	sei();
}


// **********************Lysdiod-kommandon*********************
void lightOff(){		//stänger av alla lampor
	PORTB |= 0b11111111;
	lightOn=0;

}

void allLightOn(){				//sätter på alla lampor
	PORTB &= ~(0b11111111);
}


void randomNumber(){
	lightTurnedOn= (int) count%6;
}

void randomLight(){
	randomNumber();
	switch(lightTurnedOn){
		case 0: PORTB &= ~LIGHT_1 ; lightOn=1; break;
		case 1: PORTB &= ~LIGHT_2 ; lightOn=2; break;
		case 2: PORTB &= ~LIGHT_3 ; lightOn=3; break;
		case 3: PORTB &= ~LIGHT_4 ; lightOn=4; break;
		case 4: PORTB &= ~LIGHT_5 ; lightOn=5; break;
		case 5: PORTB &= ~LIGHT_6 ; lightOn=6; break;
	}
}

void twoPlayerRandomNumber(){
	lightTurnedOn= (int) count%3;
}

void twoPlayerRandomLight(){
```

```
        twoPlayerRandomNumber();
        switch(lightTurnedOn){
                case 0: PORTB &= ~LIGHT_1 ; lightOn=1; PORTB &= ~LIGHT_4 ;   break;
                case 1: PORTB &= ~LIGHT_2 ; lightOn=2; PORTB &= ~LIGHT_5 ;  break;
                case 2: PORTB &= ~LIGHT_3 ; lightOn=3; PORTB &= ~LIGHT_6 ;  break;
        }
}

void celebrateLight(){
        for( a = 0; a < 5; a = a + 1 ){
                PORTB &= ~LIGHT_1;
                _delay_ms(50);
                PORTB &= ~LIGHT_2;
                _delay_ms(50);
                PORTB &= ~LIGHT_3;
                _delay_ms(50);
                PORTB &= ~LIGHT_4;
                _delay_ms(50);
                PORTB &= ~LIGHT_5;
                _delay_ms(50);
                PORTB &= ~LIGHT_6;
                _delay_ms(50);
                lightOff();
        }
}




// *******Knapparna och interrupten****************


ISR(INT0_vect){
        buttonPressed = 1;
        _delay_ms(20);

        if(initiateGame == 1){
                if(PIND == BTN_1 || PIND == BT_1){
                        gameMode = 1;
                }
                if(PIND == BTN_2 || PIND == BT_2){
                        gameMode = 2;
                }
                if(PIND == BTN_3 || PIND == BT_3){
                        gameMode =3;
                }
```

```
        } else{
                if(gameMode == 2){
                        checkButtonTwo();
                } else if(gameMode == 1) {
                        checkButton();
                } else{

                }
        }
}


void checkButton(){
        switch(lightOn){
                case 1 :
                if(PIND == BTN_1 || PIND == BT_1){
                        lightOn = 0;
                        lightOff();
                        endTime = countClock;
                        rightButton = 1;
                }
                break;
                case 2 :
                if(PIND == BTN_2 || PIND == BT_2){
                        lightOn = 0;
                        lightOff();
                        endTime = countClock;
                        rightButton = 1;
                }
                break;
                case 3 :
                if(PIND == BTN_3 || PIND == BT_3){
                        lightOn = 0;
                        lightOff();
                        endTime = countClock;
                        rightButton = 1;
                }
                break;
                case 4 :
                if(PIND == BTN_4 || PIND == BT_4){
                        lightOn = 0;
                        lightOff();
                        endTime = countClock;
                        rightButton = 1;
```

```
                }
                break;
                case 5 :
                if(PIND == BTN_5 || PIND == BT_5){
                        lightOn = 0;
                        lightOff();
                        endTime = countClock;
                        rightButton = 1;
                }
                break;
                case 6 :
                if(PIND == BTN_6 || PIND == BT_6){
                        lightOn = 0;
                        lightOff();
                        endTime = countClock;
                        rightButton = 1;
                }
                break;
        }
}

void checkButtonTwo(){
switch(lightOn){
        case 1 :
        if(PIND == BTN_1 || PIND == BT_1){
                lightOff();
                rightButton = 1;
                lightOn = 0;
                endTime = countClock;
                playerOne = 1;
                break;
        }
        if(PIND == BTN_4 || PIND == BT_4){
                lightOff();
                rightButton = 1;
                lightOn = 0;
                endTime = countClock;
                playerTwo = 1;
                break;
        }
        break;
        case 2 :
        if(PIND == BTN_2 || PIND == BT_2){
                lightOff();
                rightButton = 1;
                lightOn = 0;
```

```
                endTime = countClock;
                playerOne = 1;
                break;
        }
        if(PIND == BTN_5 || PIND == BT_5){
                lightOff();
                rightButton = 1;
                lightOn = 0;
                endTime = countClock;
                playerTwo = 1;
                break;
        }
        break;
        case 3 :
        if(PIND == BTN_3 || PIND == BT_3){
                lightOff();
                rightButton = 1;
                lightOn = 0;
                endTime = countClock;
                playerOne = 1;
                break;
        }
        if(PIND == BTN_6 || PIND == BT_6){
                lightOff();
                rightButton = 1;
                lightOn = 0;
                endTime = countClock;
                playerTwo = 1;
                break;
        }
        break;
}
}

void waitForButton(){
        buttonPressed = 0;
        while(buttonPressed == 0){
                writeString("");
        }
        _delay_ms(150);
        buttonPressed = 0;
}

// **********************Timers*********************

//Overflow Interrupt (för klockan)
```

```
ISR(TIMER0_OVF_vect){ // TIMER0_OVF_vect
	count++;
}

ISR(TIMER2_OVF_vect) { // TIMER0_OVF_vect
	countClock++;
}



// **********************Display-kommandon*********************
void display_cmd(char c){

	PORTD |= _BV(PD7); // RW, RS -> 0, E -> 1
	PORTC &= ~_BV(PC0); //RW
	PORTC &= ~_BV(PC1); //RS
	PORTA = c;   // skickar c
	PORTD &= ~_BV(PD7); // läser av c när E=0
	_delay_ms(10);
	PORTD |= _BV(PD7); // RW, RS -> 0, E -> 1
}

void writeChar(char c) {
	PORTA = c;
	PORTC |= _BV(PC1);
	PORTC &= ~_BV(PC0);
	PORTD |= _BV(PD7);
	PORTD &= ~_BV(PD7);  // sätter RS till 1 RW till 0
	_delay_us(50);
}

void writeString(char string[]){
	//display_cmd(0b00000000); // sätter cursorn i början när vi skriver
	int i = 0;
	while(string[i] != '\0'){ // "\0" betyder "null" i ASCII.
		writeChar(string[i]); //Skriver ut bokstaven.
		i++;
	}
}


void firstMessage(){
	writeString("Do you want to play ");
	writeString("Press any button    ");
	writeString("a game? ");
	_delay_ms(2);
```

```
}

void initiateGameMessage(){
                    display_cmd(0b00000001);
                    writeString("Get Ready");
                    _delay_ms(1000);
                    display_cmd(0b00000001); // clearar
                    writeString("3");
                    _delay_ms(1000);
                    display_cmd(0b00000001);
                    writeString("2");
                    _delay_ms(1000);
                    display_cmd(0b00000001);
                    writeString("1");
                    _delay_ms(1000);
                    display_cmd(0b00000001);
                    writeString("GO");
                    _delay_ms(500);
                    display_cmd(0b00000001);
}

void writePoints(){
          char str[5];
 sprintf(str, "%d", pointCount); //Int till char
 writeString(str);
}

void clearDisplay(){
          display_cmd(0b00000001);
}

void displayScore(){
          writeString("Score: ");
          writePoints();
          writeString("          For Menu");
          writeString("            Press Any Button");
          _delay_ms(1000);
          initiateGame = 1;
          pointCount = 0;
          gameround = 0;
}

void twoPlayerOneWritePoints(){
          char stri[5];
          sprintf(stri, "%d", playerOnePoints);
          writeString(stri);
```

```
}

void twoPlayerTwoWritePoints(){
	char strin[5];
	sprintf(strin, "%d", playerTwoPoints);
	writeString(strin);
}

void twoPlayerDisplayScore(){
	writeString("Player1 Score: ");
	twoPlayerOneWritePoints();
	writeString("    Press any button");

	writeString("    Player2 Score: ");
	twoPlayerTwoWritePoints();
	_delay_ms(1000);
	buttonPressed = 0;
	gameround = 0;
	playerOnePoints = 0;
	playerTwoPoints = 0;
}

void highScore(){
	if (highScoreNumber == 10000){
		clearDisplay();
		writeString("No one has played   ");
		writeString("button for Menu     ");
		writeString("yet. Press Any");
		_delay_ms(1000);
		_delay_ms(1000);
	}
	else {
		clearDisplay();
		writeString("HighScore is: ");
	char string[5];
	sprintf(string, "%d", highScoreNumber);
	writeString(string);
	writeString("For Menu            ");
	writeString("Press Any Button");
	_delay_ms(1000);
	}
}

void displayMenu(){
	clearDisplay();
	writeString("BTN1 = OnePlayerMode");
```

```
	writeString("BTN3 = HighScore    ");
	writeString("BTN2 = TwoPlayerMode");
	_delay_ms(20);
}



// **********************Game********************
void runGameRound(){
	while(gameround < 10) {
		randomLight();
		currentTime = countClock;
		rightButton = 0;
		while(rightButton == 0){
			writeString("");
		}
		rightButton = 0;
		pointCount = pointCount + (endTime - currentTime)/0.030637254902;  //Dela
med 3.97 för att få i millisekunder.
		gameround++;
		buttonPressed = 0;
		_delay_ms(10);
	}
	if(pointCount < highScoreNumber){
		highScoreNumber = pointCount;
		writeString("                ");
		writeString("                ");
		writeString("    NEW HIGHSCORE!");
		celebrateLight();
		clearDisplay();
	}
	gameMode = 0;
	gameround=0;
}


void twoPlayerMode(){
	while(gameround < 10){
		twoPlayerRandomLight();
		while(rightButton == 0){
			writeString("");
		}
		rightButton=0;
		if(playerOne == 1){
			playerOnePoints++;
		}
```

```
                if(playerTwo == 1){
                        playerTwoPoints++;
                }
                playerOne = 0;
                playerTwo = 0;
                rightButton = 0;
                gameround++;
                buttonPressed = 0;
                _delay_ms(100);
        }
        gameMode = 0;
        gameround = 0;

}


// **********************Main********************
int main (void) {
        setUpPorts();
        clearDisplay();

        firstMessage();
        waitForButton();
        displayMenu();
        gameMode = 0;
        while(gameMode == 0){
                waitForButton();
        }


        while(1){
                initiateGame = 0;
                if(gameMode == 1){
                        count = 0;
                        countClock = 0;
                        initiateGameMessage();
                        runGameRound();
                        displayScore();
                }
                if(gameMode == 2){
                        count = 0;
                        countClock = 0;
                        initiateGameMessage();
                        twoPlayerMode();
                        twoPlayerDisplayScore();
                 }
```

```
                if(gameMode==3){
                        highScore();
                }
                 waitForButton();
                 clearDisplay();
                 displayMenu();
                 gameMode=0;
                 initiateGame=1;
                 while(gameMode == 0){
                        waitForButton();
                 }
                 clearDisplay();


        }
}
```