

<<IERG3810>>

<<Microcontroller and Embedded Systems Laboratory>>

Report on Project

<< Jumping Pig>>

Group: 19

Member: Chan Kai Yin (CUID: 1155124983)

Ip Tsun Yu (CUID: 1155144668)

Submission Date: 29-04-2022

Disclaimer

I declare that the assignment here submitted is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website <http://www.cuhk.edu.hk/policy/academichonesty/>

Jesse

Chan Kai Yin

29-04-2022

Signature

Name

Date

Derek

Ip Tsun Yu

29-04-2022

Signature

Name

Date

I. OBJECTIVES

- Game design on the STM32.
- Combines features learn from the labs

II. Description

This game is a single-player game. The player needs to control the main character Pig to dodge / jump over the obstacles to escape from the typhoon. The score depends on how far the Pig can go.

III. Game Control

We provide two difficulties for players to choose. Easy and hard modes refer to different speeds at which obstacles move. In Hard mode, obstacles move faster, making the game more challenging.

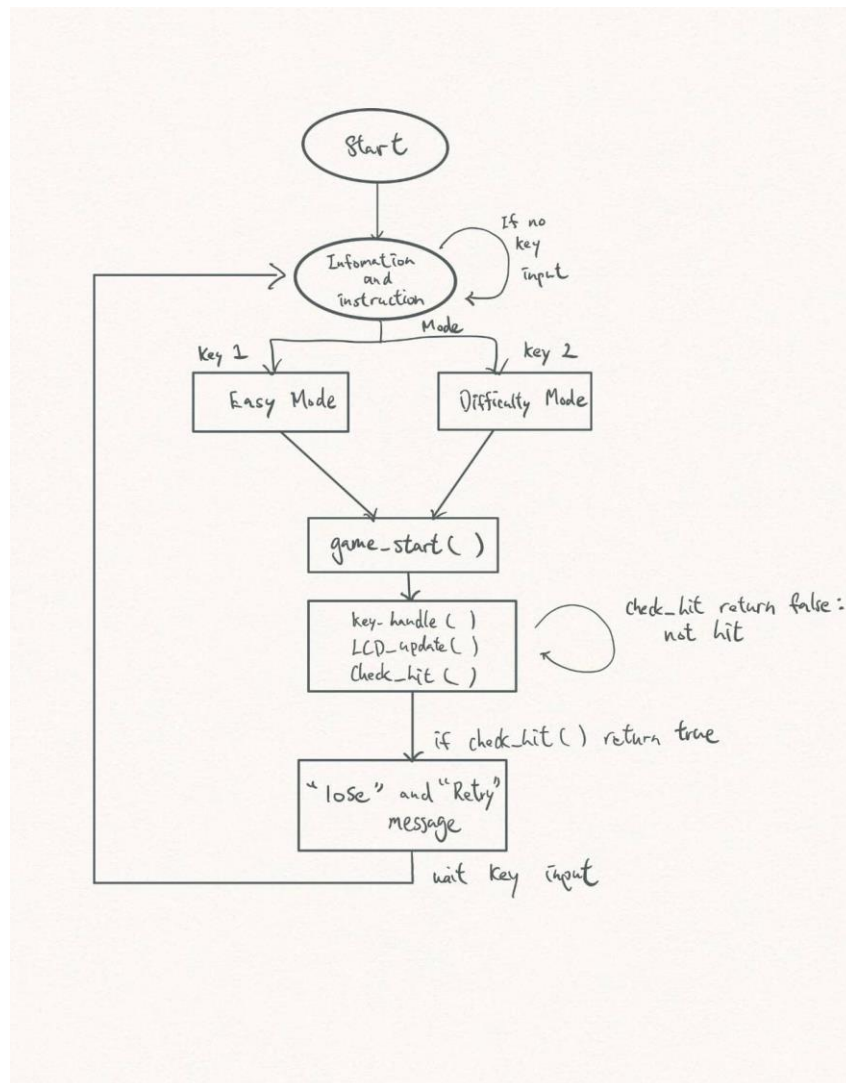
The score keep increase while playing and displayed on upper left area of the LCD screen. The game stops the score increment if the player is hit by the obstacles. After a hit occurs, the screen will display the score, the game lost message and wait for a restart.

IV. Equipment & Software

- Personal computer
- Project Board
- PS2 Keyboard
- KEIL MKD-ARM
- STM321F10x Standard Firmware Library

V. Program Flow Chart

When game start, a list of procedures will be executed as stated on the chart. First choose the difficulties for the game. Then while the game running, it updates the LCD for game animation and do the hit detection. The game ends with a page showing lose message, score and retry instructions.



VI. Program Component

1. Cover Page Display

It includes the name and SID of me and my groupmate Derek on the top area. Then, it shows the instruction of game and the key corresponding to different difficulties. It waits if no input control for the game.



2. Difficulties Choose

The program detects and catch what user input for the difficulties modes. Then it starts with game_start() function along with setting the game_speed variables which different between the easy and hard mode.

```
//easy mode, keyUp, exti0
} else if (game_mode == 1){
    game_speed = 1;
    USART_print(1, "Game Start\n");
    IERG3810_TFTLCD_FillRectangle(white, 0, 240, 0, 320);
    game_start();

    // hard mode:
} else if (game_mode == 2) {
    USART_print(1, "Game Start\n");
    game_speed = 2;
    //IERG3810_TFTLCD_FillRectangle(white, 0, 240, 0, 320);

    game_start();
}
```

3. The Count Down Before Game Start

The countdown is designed for giving time for player to get ready. It is using IERG3810_TFTLCD_SevenSegment() function to draw the timer. The CountFrom3() function gives about 3 second preparation time.

```
void CountFrom3(void){
    uint6 i= 3;
    IERG3810_TFTLCD_FillRectangle(white,80, 80, 80,160);
    while(1){
        IERG3810_TFTLCD_SevenSegment(red,80,80,i);
        while(1){if (task1HeartBeat % 100 == 0) break;}
        IERG3810_TFTLCD_FillRectangle(white,80, 80, 80,160);
        if (i == 0){ // 0
            break;
        }
        i--;
    }
}
```

4. LCD Screen Update

The game graphic works with moving pixel to make the animation. It first draws a colored character and later on erase it and update the colored character into new locations. An example of cloud animation, the IERG3810_TFTLCD_ShowColud makes the drawing. To make the animation smoother and more random, the updated location will use a random increment for each time update. Same idea also applies to the animation of the Pig. The timer for controlling the speed animation is using the SysTick Timer.

```
ran = rand() % (230+1 - 200) + 200;
IERG3810_TFTLCD_ShowColud(140-x_inc*i, y, 0,color,bgcolor); // Cloud
IERG3810_TFTLCD_ShowColud(140-x_inc_2*i, ran, 0,color,bgcolor); // cloud 2
IERG3810_TFTLCD_ShowColud(140-x_inc_3*i, ran+20, 0,color,bgcolor); // cloud 3
IERG3810_TFTLCD_FillRectangle(black, 0, 240, 290, 320);

//IERG3810_TFTLCD_ShowColud(pig_x_pos, 100, 2,white,white); // clear pig2
IERG3810_TFTLCD_ShowColud(pig_x_pos, 100, 1,color,bgcolor); // pig1

IERG3810_TFTLCD_ShowColud(bin_cont, 100, 3,color,bgcolor); //bin

while(1){
    if (task1HeartBeat % 30 == 0){
        IERG3810_TFTLCD_ShowColud(140-x_inc*i, y, 0,color,color); // Cloud
        IERG3810_TFTLCD_ShowColud(140-x_inc_2*i, ran, 0,color,color); // cloud 2
        IERG3810_TFTLCD_ShowColud(140-x_inc_3*i, ran+20, 0,color,color); // cloud 3
        IERG3810_TFTLCD_ShowColud(bin_cont, 100, 3,color,color); // Clear Bin
        bin_control = 0;
        break;
    }
}
```

5. Character Drawing

The character which includes the Pig, rubbish bin and clouds are draw with hand on the Dot Matrix and save it inside the "IMFONT.H" file.

```
#ifndef __IMFONT_H
#define __IMFONT_H
const unsigned char IM[4][32]={
{
0xfc, 0x1f, 0xf9, 0xc7, 0xf3, 0xf7, 0xf7, 0xf7,
0xe7, 0xf7, 0xef, 0xe7, 0xcf, 0xe7, 0xdf, 0xe7,
0xdf, 0xef, 0xdf, 0xcf, 0xdf, 0xef, 0xdf, 0xcf,
0xdf, 0xdf, 0xcf, 0xdf, 0xe0, 0x1f, 0xff, 0xff
},
{
0xff, 0x8f, 0xfe, 0x3e, 0xc0, 0x0c, 0xdf, 0xe1,
0xdf, 0xf7, 0xd0, 0x10, 0xd6, 0xfb, 0xd6, 0xfb,
0x96, 0xfb, 0xb9, 0xfa, 0x8f, 0xc0, 0xcf, 0xd7,
0xdf, 0x53, 0xeb, 0x18, 0xeb, 0x7f, 0xe0, 0x7f
},
{
0xff, 0x8f, 0xfe, 0x3f, 0xc0, 0x0f, 0xdf, 0xe7,
0xdf, 0xf0, 0xd0, 0x12, 0xd6, 0xf9, 0xd6, 0xf8,
0x96, 0xfb, 0xb9, 0xfb, 0x8f, 0xc0, 0xcf, 0xda,
0xdf, 0x59, 0xeb, 0x1c, 0xeb, 0x7f, 0xe0, 0x7f
},
},
// pig2

// rubbish bin:
{0xff, 0xff, 0xe0, 0x00, 0xcf, 0xff, 0x98, 0x3f,
0xbb, 0x8d, 0xb7, 0x4f, 0xb7, 0x9d, 0xb7, 0x1f,
0xb2, 0x9d, 0x98, 0x1f, 0xce, 0x7f, 0xe7, 0xff,
0xf0, 0x00, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff
}
};

#endif
```

6. Interrupt

The Jump is controlled by the key 1 on the keyboard. An interrupt occurs when detecting the input and update the location of Pig.

```
if(ps2count >= 11){
// Jump
if(ps2key == 0x69){
IERG3810_TFTLCD_ShowColud(pig_x_pos, 100, 1,color,color); // clear pig1
IERG3810_TFTLCD_ShowColud(pig_x_pos, 100, 2,color,color); // clear pig2
IERG3810_TFTLCD_ShowColud(pig_x_pos+30, 150, 2,color,bgcolor); // make pig2
while(1){if (task1HeartBeat % 20 == 0) break;}

IERG3810_TFTLCD_ShowColud(pig_x_pos+30, 150, 2,color,color); // clear pig2
IERG3810_TFTLCD_ShowColud(pig_x_pos+50, 100, 2,color,bgcolor); // make pig2
pig_x_pos = pig_x_pos + 50;

//while(1){if (task1HeartBeat % 60 == 0) break;}
}
ps2key = 0;
ps2count = 0;
EXTI->PR = 1 << 11;
bin_control = 0;
IERG3810_TFTLCD_ShowColud(bin_cont, 100, 3,color,color); // Clear Bin
// bin_cont = bin_cont - 30;
}
```

7. The Hit Detection

The game ends when the Pig hits the obstacles. The hit detection is based on the pixel distance of those characters. The game will lose if the distance of the Pig and the rubbish bin are less than 4.

```
if(abs(bin_cont - pig_x_pos) < 4){  
    flag = 0;  
}
```

8. Information Output

In case for debugging or making extension. The program can output the game state information to the PC through the USART. The function USART_print() sends the data to the PC.

```
//easy mode, keyUp, exti0  
} else if (game_mode == 1){  
    game_speed = 1;  
    USART_print(1, "Game Start\n");  
    IERG3810_TFTLCD_FillRectangle(white, 0, 240, 0, 320);  
    game_start();  
  
    // hard mode:  
} else if (game_mode == 2) {  
    USART_print(1, "Game Start\n");  
    game_speed = 2;  
    //IERG3810_TFTLCD_FillRectangle(white, 0, 240, 0, 320);  
  
    game_start();  
}
```

When the player loses the game, it will send the score and game lose information.

```
1 }
2 void lose(int score){
3     char sc_str[100];
4
5     sprintf(sc_str, "Lose, Score: %d \n", score-1);
6     USART_print(1, sc_str); // output to the pc
7
8
9     IERG3810_TFTLCD_FillRectangle(red, 0, 250, 0, 280);
10    IERG3810_ShowSID("Lose!",150, black, red, 100);
11    IERG3810_ShowSID("Press 1 to restart.",130, black, red, 20);
12    GPIOB ->BRR = 1 << 5; // red led light
13    while(1){
14        if(ps2count >= 11){
15            if(ps2key == 0x69){
16                if(ps2key == 0x69){
17                    IERG3810_TFTLCD_FillRectangle(white, 0, 240, 0, 320);
18                    game_mode = 0;
19                    IERG3810_ShowPig(250,white, black);
20                }
21            }
22        }
23    }
24 }
```


VII. Difficulties

The game can introduce more characters such as a flying bird on the sky and background environment image. But the drawing of matrix dot is very time consuming and using may not be accuracy. An Improvement could be using some pixel map drawing software to make the character and output the array.

VIII. Conclusion

This project combines the lab1-5 materials. The LED, key configuration in lab1. The USART configuration in lab2. The LCD screen configuration in lab3. The interrupt configuration in lab4. And the timer configuration in lab5.

IX. DIVISION OF WORK

<Lab work: Jesse & Derek, Report writing: Jesse>

X. REFERENCES