

EIE3810 Microprocessor System Design Laboratory

Laboratory Report #6

Name: Tengfei Ma

Student ID: 121090406

Date: Dec 8, 2023

The Chinese University of Hong Kong, Shenzhen

- Experiment 1:Realize the bouncing ball game
- Experiment 2: Improve the bouncing ball game

1. Experiment 1

1.1 Design

1.1.1 Explanation of source code

1.1.1.1 I use a few global variables to record the status of game as follows.

```

u8 level;//0 for easy, 1 for hard
u8 status;// 0 for pause,
           // 1 for selecting difficulty level,
           // 2 for initialize direction,
           // 3 for counting down,
           // 4 for playing,
           // 5 for game over
u8 direction;//the random number for direction
u8 winner = 0;//A is 0; B is 1

u32 time = 0;
u32 bounceU = 0;//upper play's bounce
u32 bounceD = 0;//lower play's bounce
u32 bounceT = 0;//two play's bounce

ul6 x_playerB = 200;//joypad, user
ul6 x_playerA = 200;//key, user
ul6 ballx = 240;//x position of the ball
ul6 bally = 400;//y position of the ball
int x_speed = 2;//x axis speed of the ball
int y_speed = 2;//y axis speed of the ball

```

1.1.1.2 I mainly use TIM3 to realize pause and moving ball and boards, namely, check the status of JOYPAD and keys periodically. And TIM4 is used to record the time and the number of collisions.

```

EIE3810_TIM3_Init(99,7199);//set the counter number + 1 to be 99

while(1)
{
    if (status==0)//status is pause
    {
        while(status==0);//wait pause over
    }
    else if (status==4)//status is playing
        EIE3810_TIM4_Init(9999,7199);//the frequency is 1Hz
    else if (status == 5)//status is game over
    {
        EIE3810_TFTLCD_FILLRectangle(0,480,0,800,WHITE);//clear the whole screen

        EIE3810_TFTLCD_ShowS2412(10,200,"The Winner is: ",15, BLACK, WHITE);//show the detail of the information
        if (winner == 0)//winner is A
            EIE3810_TFTLCD_ShowS2412(10 + 15 * 12,200,"A",1, BLACK, WHITE);//show the name of A
        else //winner is B
            EIE3810_TFTLCD_ShowS2412(10 + 15 * 12,200,"B",1, BLACK, WHITE);//show the name of B
        Delay(3000000);
        break;
    }
}

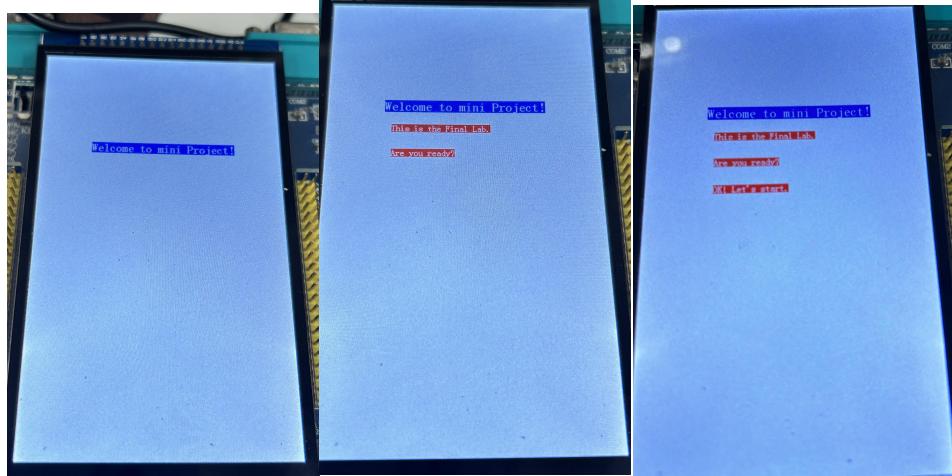
```

1.1.1.3 I will check whether the ball collide the up or down boundary(not the board) to judge who wins when moving ball.

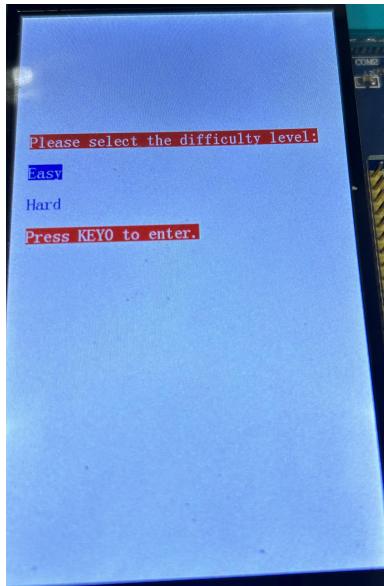
```
void moveBall(void)
{
    if (status==0) return;//status is pause
    EIE3810_TFTLCD_DrawCircle(ballx, bally, 10, 1, WHITE); // clear ball
    if (ballx >= 480-10 || ballx<=10)//hit the left and right boundary
    {
        x_speed = - x_speed;
        Buzzer_Toggle();
    }
    else if (isHitPad())//hit the pad
    {
        y_speed = - y_speed;
        Buzzer_Toggle();
    }
    else if (bally <= 10+1 || bally>=800-10-1)//hit the up and down boundary
    {
        status = 5;//status is game over
        if (bally <= 10 + 1) winner = 0;//A win
        else if (bally >= 800 - 10 - 1) winner = 1;//B win
    }
    ballx = ballx + x_speed;//update position
    bally = bally + y_speed;//update position
    EIE3810_TFTLCD_DrawCircle(ballx, bally, 10, 1, RED);//draw the ball
}
```

1.2 Result

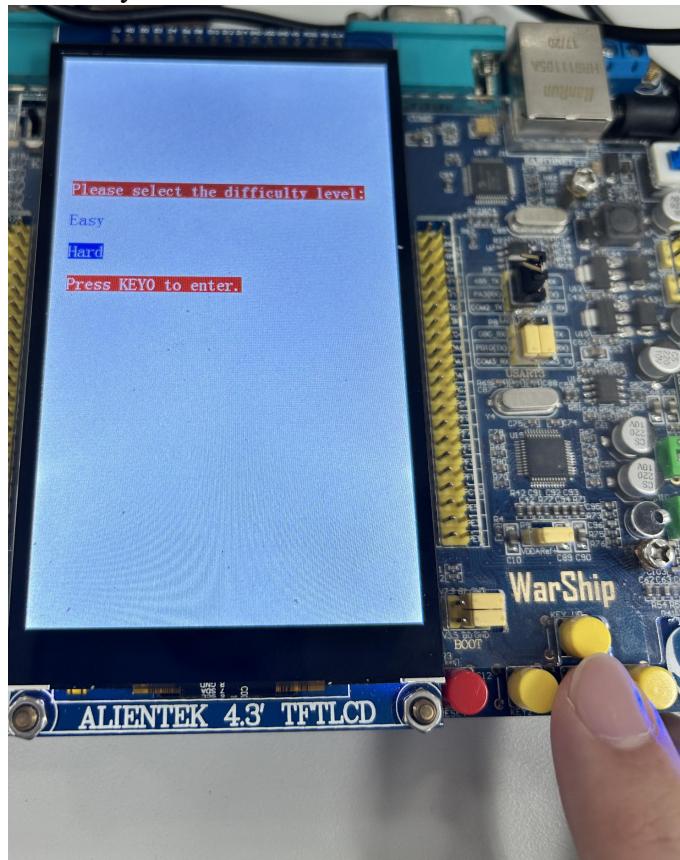
1.2.1 Welcome page:



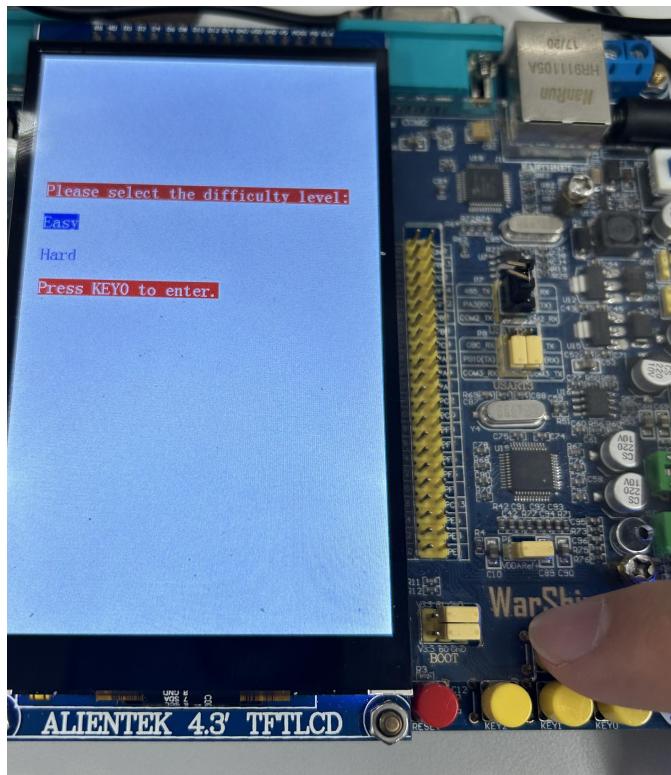
1.2.2 Set difficulty level:



1.2.3 Push Key1:



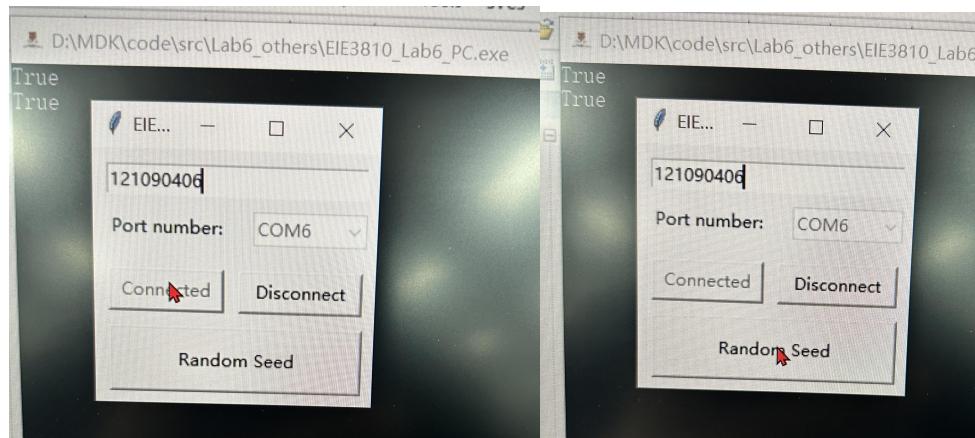
1.2.4 And push Key_up



1.2.5 Push Key0 and enter the page of direction selecting



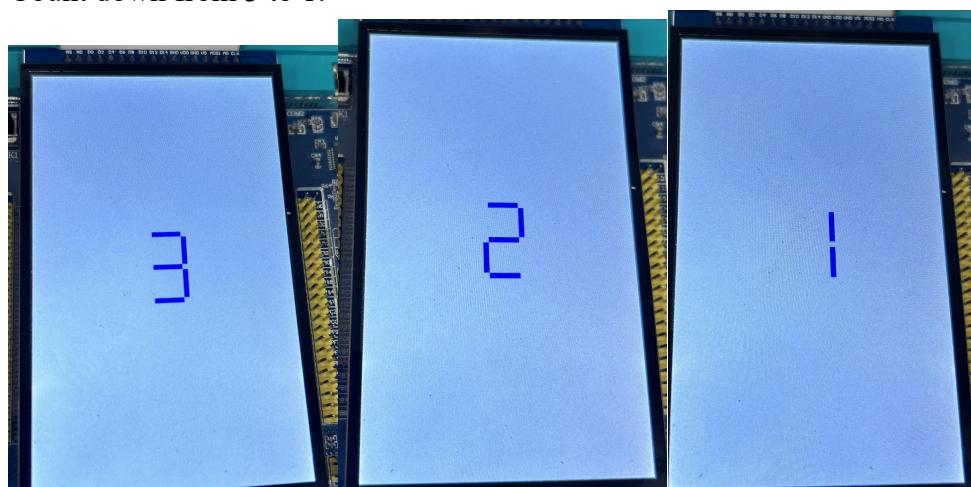
1.2.6 Use USART to select direction:



1.2.7 Get the direction:

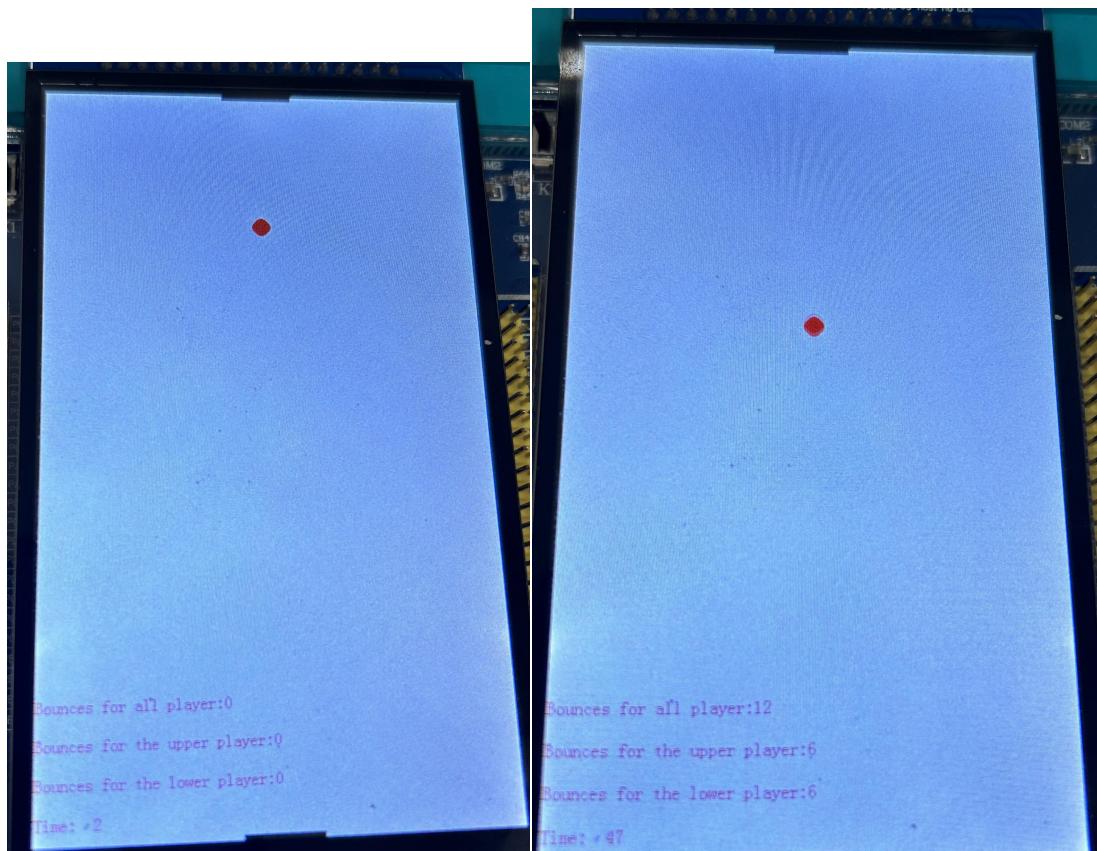


1.2.8 Count down from 3 to 1:

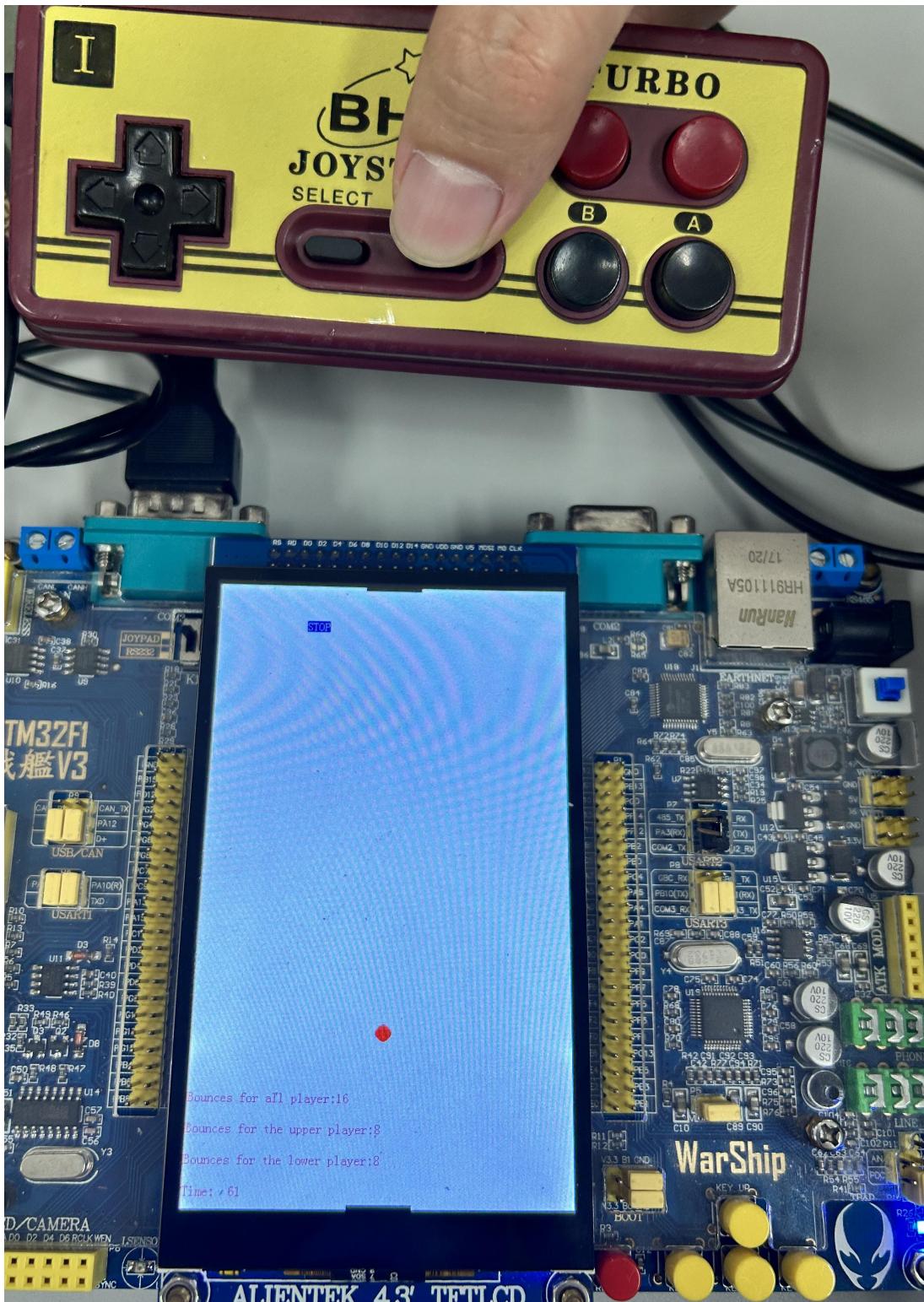


1.2.9

1.2.10 Game starts:



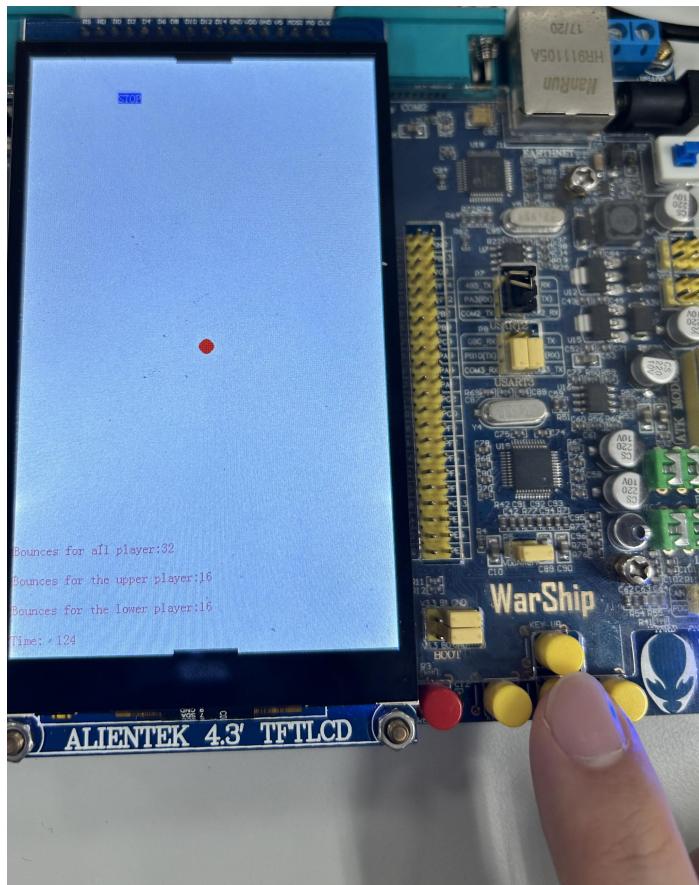
1.2.11 Push “Start” to pause:



1.2.12 Push “start” again to restore:



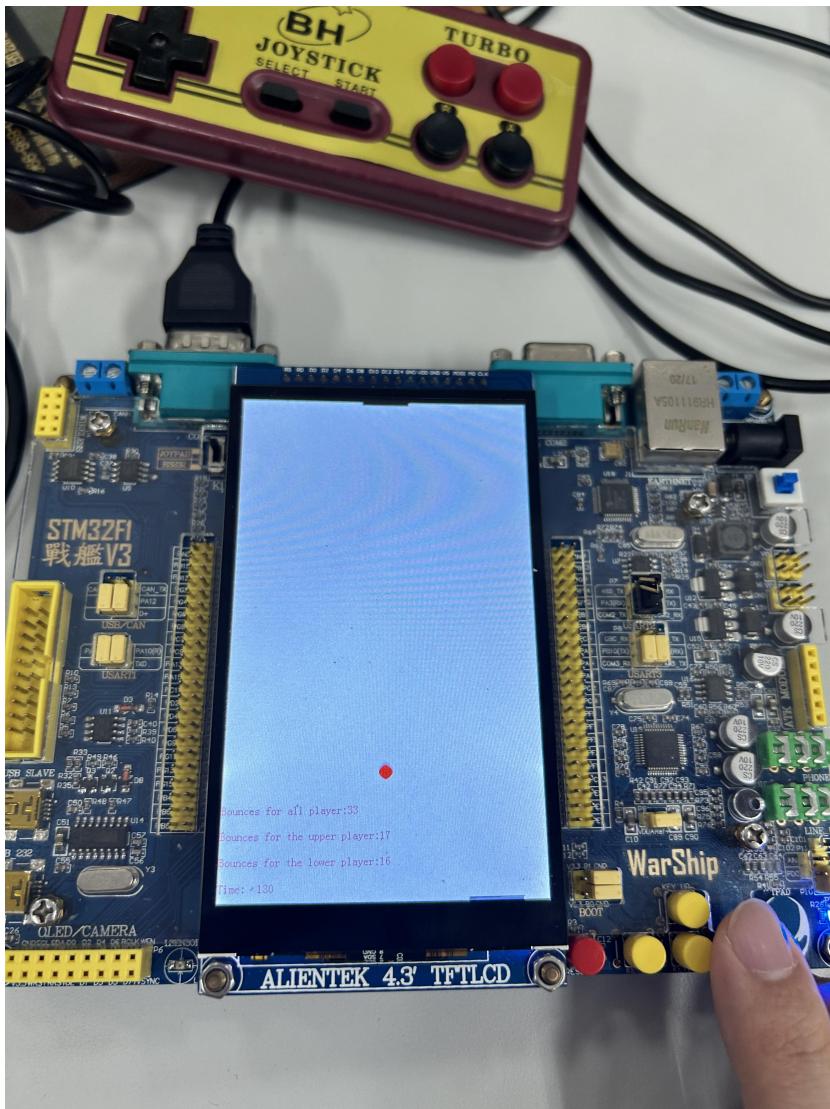
1.2.13 Push Key_up to pause:



1.2.14 Push Key_up again to restore:



1.2.15 Change position of board:



1.2.16 Player B wins



2. Experiment 2

2.1 Design

2.1.1 Improvement

2.1.1.1 Every collision of ball causes the color of ball to change.(color range:BLACK,RED,YELLOW). And Player A can use Key_up to change the color of lower board, while Player B can use “up” to change the color of upper board. Only if the board and the ball have the same color, the collision between ball and board is effective. Otherwise, you will lose even if your board touch the ball but the ball have the different color from your board.

2.1.2 Explanation of source code

2.1.2.1 I use global variable to record the color of boards and ball.

```
u32 ballColor = BLACK;
u32 boardColorA = BLACK;
u32 boardColorB = BLACK;
```

2.1.2.2 I implement functions to change color of boards and ball.

(BLACK->RED->YELLOW->BLACK)

```
void changeBoardColorA(void)
{
    if (boardColorA == BLACK) boardColorA = RED;
    else if (boardColorA == RED) boardColorA = YELLOW;
    else if (boardColorA == YELLOW) boardColorA = BLACK;
}

void changeBoardColorB(void)
{
    if (boardColorB == BLACK) boardColorB = RED;
    else if (boardColorB == RED) boardColorB = YELLOW;
    else if (boardColorB == YELLOW) boardColorB = BLACK;
}

void changeBallColor(void)
{
    if (ballColor == BLACK) ballColor = RED;
    else if (ballColor == RED) ballColor = YELLOW;
    else if (ballColor == YELLOW) ballColor = BLACK;
}
```

2.1.2.3 Only the same color collision is effective:

```
// check if the ball hit the pad
u8 isHitPad(void)
{
    int comp;
    if (bally >= 800-5-10-6)//hit user A's pad
    {
        comp = ballx - x_playerA;//the difference between the ball and pad's left side
        bounceD ++;//user point ++
        bounceT ++;//total point ++
        return (ballColor == boardColorA)&(comp <= 80 && comp >= 0);//the difference less than the length
    }
    if (bally<=0+10+6)//hit user B's pad
    {
        comp = ballx - x_playerB;//the difference between the ball and pad's left side
        bounceU ++; //user point ++
        bounceT ++;//total point ++
        return (ballColor == boardColorB)&(comp <= 80 && comp >= 0);//the difference less than the length
    }
    return 0;
}
```

2.1.2.4 Ball color will change if hit left and right boundary and boards. And if there is no effective collision between ball and board, the game will be over.

```

void moveBall(void)
{
    if (status==0) return;//status is pause
    EIE3810_TFTLCD_DrawCircle(ballx, bally, 10, 1, WHITE); // clear ball
    if (ballx >= 480-10 || ballx<=10)//hit the left and right boundary
    {
        x_speed = - x_speed;
        Buzzer_Toggle();
        changeBallColor();
    }
    else if (isHitPad())//hit the pad
    {
        y_speed = - y_speed;
        Buzzer_Toggle();
        changeBallColor();
    }
    else if (bally <= 10+1 || bally>=800-10-1)//hit the up and down boundary
    {
        status = 5;//status is game over
        if (bally <= 10 + 1) winner = 0;//A win
        else if (bally >= 800 - 10 - 1) winner = 1;//B win
    }
    ballx = ballx + x_speed;//update position
    bally = bally + y_speed;//update position
    EIE3810_TFTLCD_DrawCircle(ballx, bally, 10, 1, ballColor);//draw the ball
}

```

2.1.2.5

2.1.2.6 Push “up” of JOYPAD to change color of upper board.

```

else if (((joypad>>4)&0x1) == 0x01)//push up, change board color
{
    changeBoardColorB();
}

```

2.1.2.7 Push Key_up to change color of lower board.

```

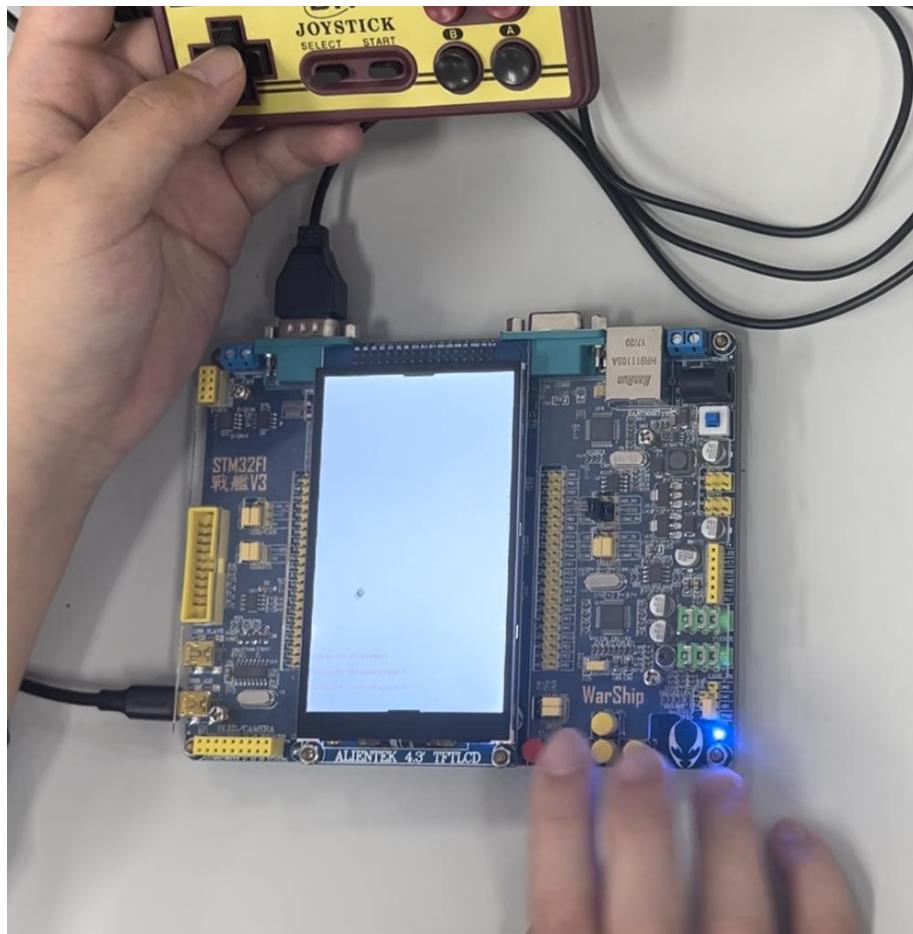
else if (Keyup == 1)//change color of board
{
    changeBoardColorA();
}

```

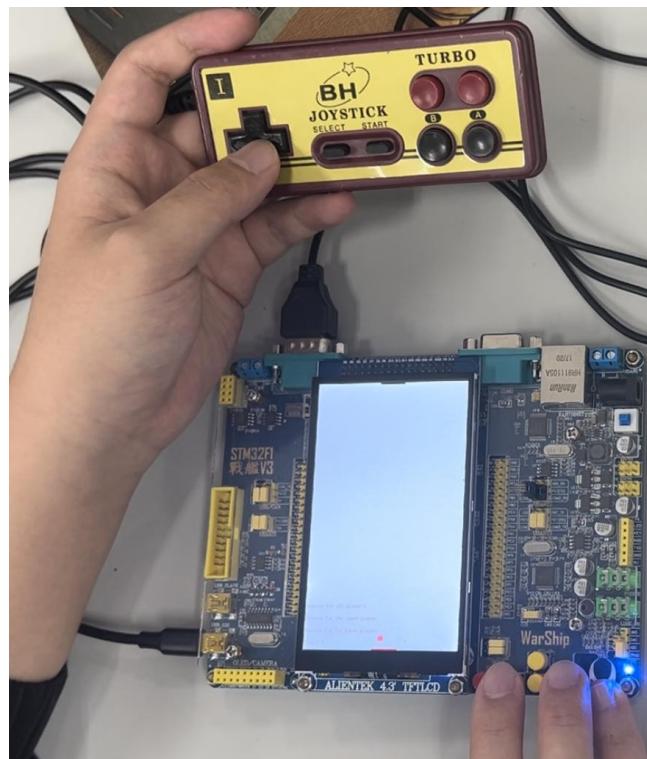
2.2 Result

2.2.1 I show some screen shot of my video, if you want details, you can watch my video.

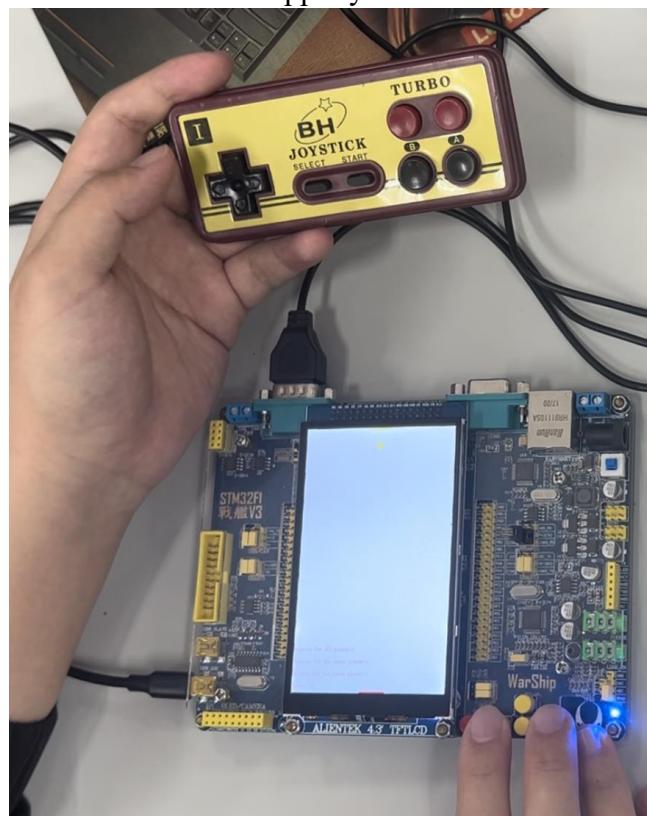
Black ball rushes to black lower board:



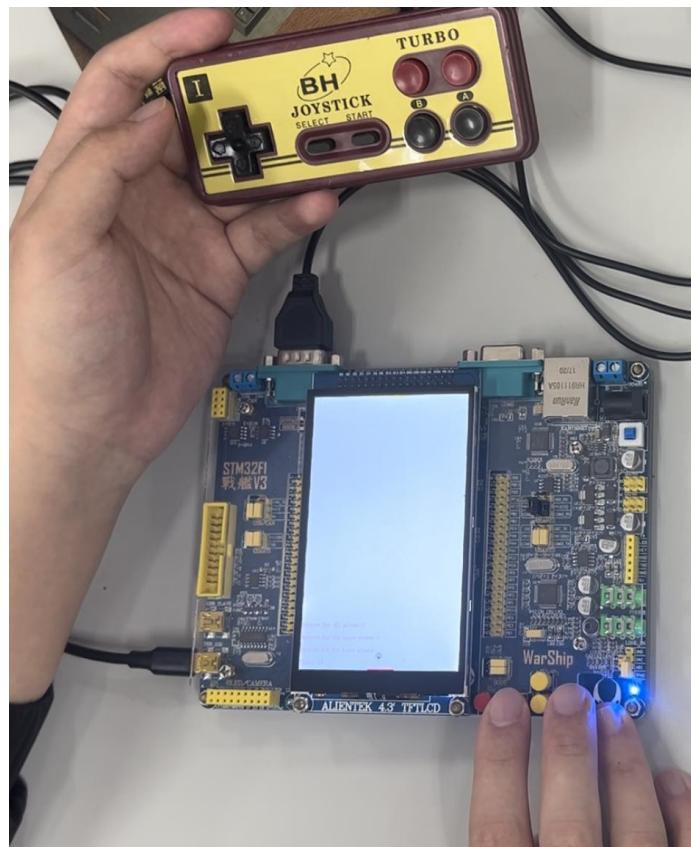
Red ball hits red lower board:



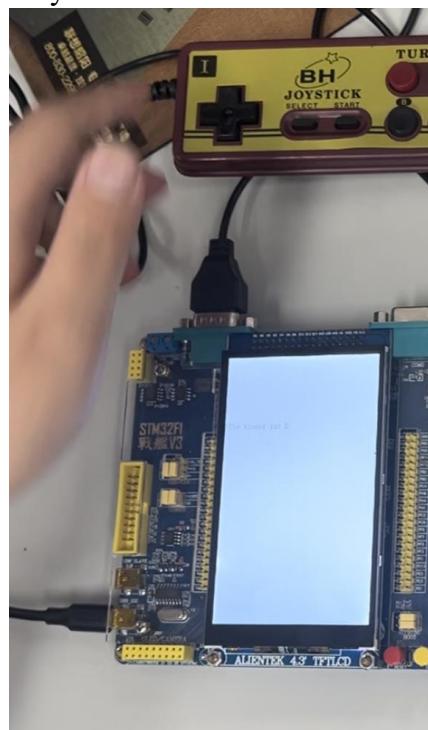
Yellow ball hits the upper yellow board:



Black ball hits red lower board:



Player B wins:



3. Conclusion

I have designed and built up a 2-player bouncing ball game based on the knowledge learned through Lab 1-5. And propose and improve the game based on my design. I

have practiced GPIO, USART, LCD, External interrupt, Timer, which makes have a deep understanding of STM32F103ZET6. Thanks for all the TAs and instructor.