

รายงานโครงการ  
วิชาการสร้างโปรแกรมคอมพิวเตอร์  
“เกมส์ Othello”

โดย

1.นายรเมศ ย้งดำ	ID: 57070098	SEC : 3
2.นายวรเวทย์ ศิริพัฒน์พงศ์	ID: 57070104	SEC : 3
3.นายวัชรกร มะลิขาว	ID: 57070106	SEC : 3
4.นายศาสตรา ตันท์ศุภศิริ	ID: 57070111	SEC : 3
5.สัฟวัน แปเราะ	ID: 57070125	SEC : 1

เสนอ

อาจารย์ ดร.กิตติ์สุชาติ พสุภา

อาจารย์ ดร.ณัฐพล พันธุ์วงศ์

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา

06016206 COMPUTER PROGRAMMING

สาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2557

หัวข้อโครงการ	เกมส์ Othello		
วิชา	COMPUTER PROGRAMMING		
	06016206		
จัดทำโดย	นายรเมศ ยังดำ	ID: 57070098	Section: 3
	นายวรเวทน์ ศิริพัฒน์พงศ์	ID: 57070104	Section: 3
	นายวัชรกร มะลิขาว	ID: 57070106	Section: 3
	นายศาสตรา ตัณตศุภศิริ	ID: 57070111	Section: 3
	นายสัฟวัน แปเราะ	ID: 57070125	Section: 1
ปีการศึกษา	2557		
ส่ง	อาจารย์ ดร.กิต์สุชาติ พสุภา		
	อาจารย์ ดร.ณัฐพล พันธุ์วงศ์		

### บทคัดย่อ

โอเทลโล่ เป็นเกมหมากระดานที่ถูกคิดค้นในราวปี ค.ศ.1880 โดยชาวอังกฤษ นามว่า ลูอิส วอเตอร์แมน โดยในแรกเริ่มมีชื่อว่า รีเวอร์ซี่(Reversi) ต่อมาในปี ค.ศ.1970 ชาวญี่ปุ่น โกะโร ฮาเซกาว่า ได้ปรับเปลี่ยนกฎและกติกาใหม่ โดยได้แรงบันดาลใจจากหมากล้อม แล้วเปลี่ยนชื่อเป็น"โอเทลโล่" ตามบทประพันธ์ของวิลเลียม เชกสเปียร์ ที่มีเนื้อหาเกี่ยวกับโศกนาฏกรรมของคนดำและคนขาว

โดยกลุ่มของผู้พัฒนาต้องการที่จะเขียนโปรแกรมเกม"โอเทลโล่" เพื่อใช้เล่นในยามว่าง และผ่อนคลาย จนถึงแบ่งปันให้ผู้อื่นได้ทดลองเล่น เพื่อความสนุกสนาน

## สารบัญ

บทคัดย่อ.....	ii
สารบัญ.....	iii
<b>บทที่ 1 : บทนำ.....</b>	<b>1</b>
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 ประโยชน์ที่ได้รับ.....	2
<b>บทที่ 2 : ทฤษฎีที่เกี่ยวข้อง.....</b>	<b>3</b>
2.1 เครื่องมือที่ใช้ในการเขียน.....	3
2.2 โครงสร้างของโปรแกรม.....	3
2.2.1 ฟังก์ชันที่ใช้เป็นส่วนประกอบในโปรแกรม.....	3
2.2.2 การเขียนผังงาน.....	5
2.2.3 วิธีการใช้โปรแกรม และหน้าต่างของโปรแกรม.....	6
2.2.4 วิธีการเล่น.....	12
<b>บทที่ 3 : บทสรุป.....</b>	<b>14</b>
3.1 ผลที่ได้รับ.....	14
3.2 ข้อดีและข้อเสีย.....	14
<b>ภาคผนวก.....</b>	<b>15</b>

## บทที่ 1

### บทนำ

รายงานนี้จัดทำขึ้นเพื่อชี้แจงที่มาที่ไป แนวทางการทำโปรเจกต์ หลักการทำงานของโปรแกรม รวบรวมเนื้อหาแล้วจัดส่งเป็นรายงานของวิชา COMPUTER

PROGRAMING 06016206

### 1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบันการเรียนรู้และศึกษาภายในห้องเรียนเพียงอย่างเดียว อาจไม่พอต่อการพัฒนาความรู้ในการเขียนโปรแกรม โปรเจกต์นี้จึงเกิดขึ้นเพื่อนำวิชาความรู้ที่ได้จากวิชา COMPUTER PROGRAMMING 06016206 มาเขียนโปรแกรมเกมส์ Othello ซึ่งเป็นเกมส์ที่คนทั่วไปน่าจะรู้จัก และเคยเล่นกันมาบ้างแล้ว โดยมีลักษณะคล้ายหมากล้อม

### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อทำเป็นโปรเจกต์ในวิชา COMPUTER PROGRAMING 06016206
2. เพื่อศึกษาและฝึกฝนทักษะภาษาซีนอกตำราเรียน
3. เพื่อฝึกฝนการทำงานร่วมกับผู้อื่น

### 1.3 ขอบเขตของโครงการ

1. นำความรู้ภาษาซีจากวิชา COMPUTER PROGRAMMING 06016206 มาพัฒนาเกมส์ Othello
2. ศึกษาคำสั่งต่างๆจากเว็บไซต์นอกเหนือจากภายในเนื้อหาที่เรียน

### 1.4 ประโยชน์ที่ได้รับ

1. ฝึกทักษะในการเขียนโปรแกรม
2. ได้ฝึกการทำงานร่วมกับผู้อื่น

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 เครื่องมือที่ใช้ในการเขียน

1. Sublime Text
2. Notepad

#### 2.2 โครงสร้างของโปรแกรม

##### 2.2.1 ฟังก์ชันที่ใช้เป็นส่วนประกอบในโปรแกรม

void othello()

- เป็นฟังก์ชันที่ใช้แสดงคำว่า Othello

void menu()

- เป็นฟังก์ชันที่ใช้แสดงหน้าต่างเมนู

void gotoxy()

- เป็นฟังก์ชันที่ใช้เลื่อนเคอร์เซอร์ไปยังตำแหน่งที่ต้องการบนจอภาพในเทกซ์โหมด (Text mode) รูปแบบของฟังก์ชัน gotoxy เป็นดังนี้ gotoxy(x,y); เมื่อ x เป็นคอลัมน์ที่จะให้เคอร์เซอร์เลื่อนไป และ y เป็นแถวแนวนอนที่จะให้เคอร์เซอร์เลื่อนไป

void credit()

- เป็นฟังก์ชันที่ใช้แสดงหน้า Credit ผู้จัดทำ

void http()

- เป็นฟังก์ชันที่ใช้แสดงหน้า How to play

void black\_win()

- เป็นฟังก์ชันที่ใช้แสดงหน้าต่าง Black Win เมื่อหมากสีดำชนะ

void white\_win()

- เป็นฟังก์ชันที่ใช้แสดงหน้าต่าง White Win เมื่อหมากสีขาวชนะ

void draw()

- เป็นฟังก์ชันที่ใช้แสดงหน้าต่าง Draw เมื่อหมากสีดำและสีขาวทั้งคู่เสมอกัน

void table\_8x8()

- เป็นฟังก์ชันที่ใช้แสดงตาราง 8\*8 ช่อง

void flip()

- เป็นฟังก์ชันที่ใช้พลิกหมากของฝ่ายตรงข้าม

void play()

- เป็นคำสั่งให้เริ่มเล่นเกม

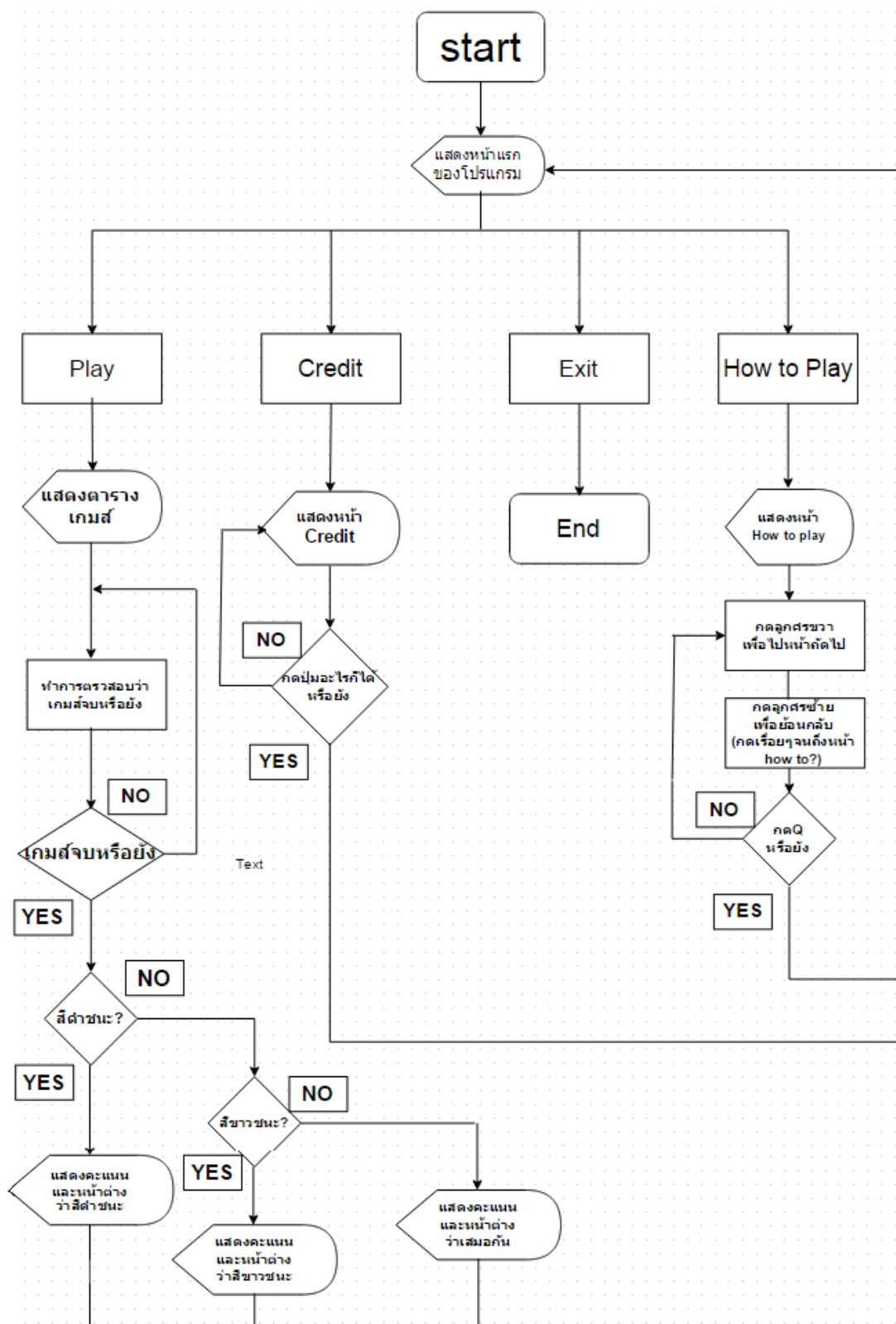
void page()

- เป็นการวนลูปเพื่อรับค่า เมื่อได้ที่ต้องการ จะแสดง How to play

void http\_table()

- เป็นฟังก์ชันที่ใช้แสดงการสอนวิธีเล่น Othello ต่อจากฟังก์ชัน void http()

## 2.2.2 การเขียนผังงาน





## 2.2.3 วิธีการใช้โปรแกรม และหน้าต่างของโปรแกรม

หน้าต่างของโปรแกรม

รูปที่ 1



หน้าจอแสดงหน้าแรกของโปรแกรม มีแถบป้ายบอกชื่อของโปรแกรม OTHELLO และเมนูให้เลือก 4 เมนู ซึ่งได้แก่

เมนู 1 – Play                      เป็นการเข้าสู่ตัวเกมส์

เมนู 2 - How to Playเป็นการอธิบายวิธีการเล่น

เมนู 3 – Credit                    เป็นการแสดงคณะผู้จัดทำโปรแกรม

เมนู 4 – Exit                      เป็นการจบการทำงานของโปรแกรม

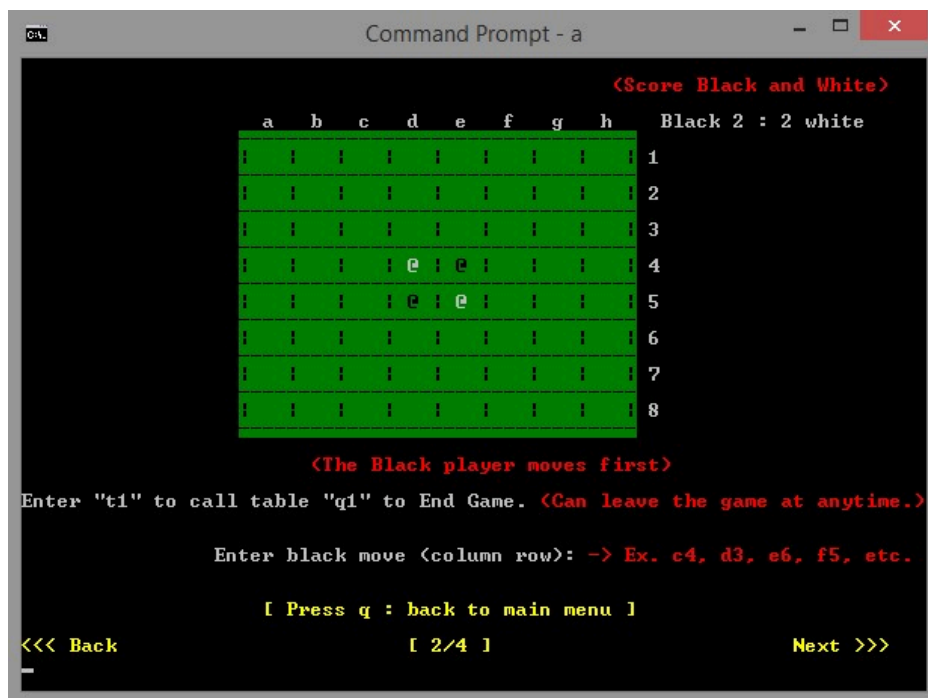
ซึ่งจะมีการแสดงข้อความ Please You Chose เพื่อให้พิมพ์ตัวเลขตามที่เมนูกำหนดไว้เข้าไป เช่น 1, 2 เป็นต้น

รูปที่ 2



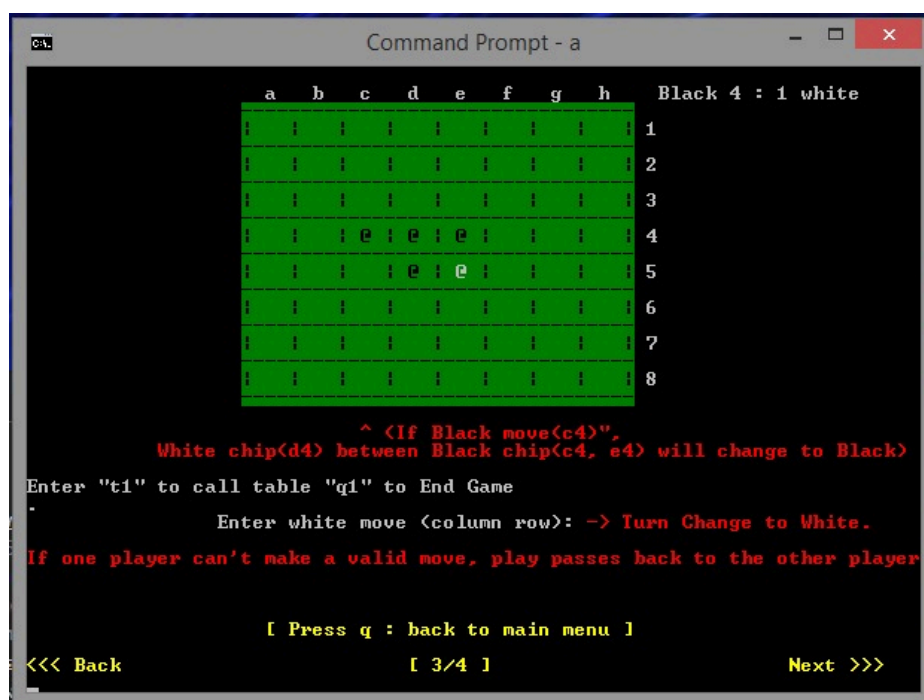
หน้าจอแสดง HOW TO? กล่าวคือ ถ้าหากมีการกดคีย์บอร์ดไปขวา หน้าจอก็เปลี่ยนไปอีกหน้า เพื่อแสดงคำอธิบายในการเล่นเกมส์ และถ้าหากต้องการออกจากหน้านี้อีกไปสู่นำเมนูหลักก็ทำการกดคีย์บอร์ด Q โปรแกรมก็จะย้อนกลับไปหน้าแรกเหมือนเดิม

รูปที่ 3



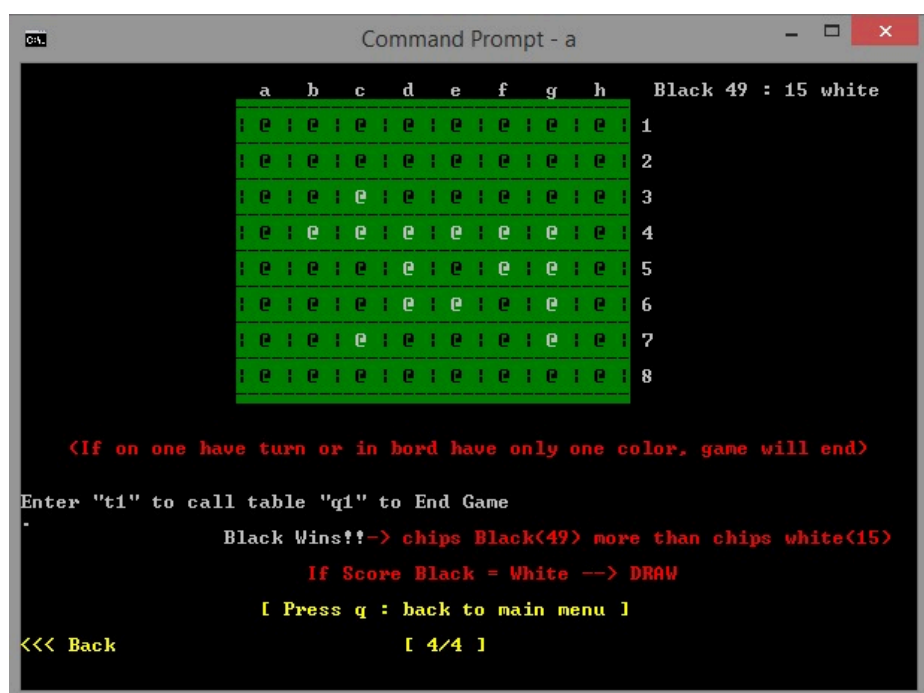
หน้าจอแสดงคำอธิบายการเล่นเกมส์หน้าแรก เมื่อต้องการอ่านอธิบายเพิ่มเติมก็กดคีย์บอร์ดไปขวาเมื่อเดิม หรือถ้าหากต้องการย้อนกลับก็ทำก็กดคีย์บอร์ดไปซ้าย

รูปที่ 4



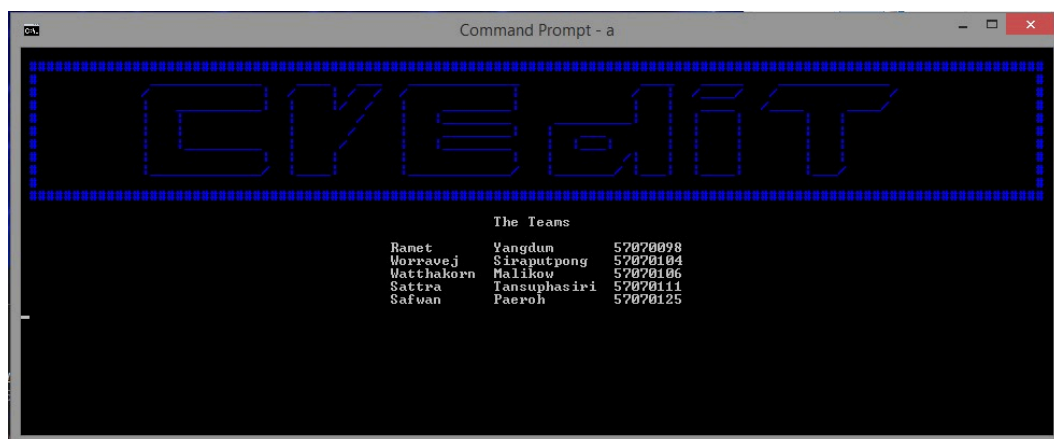
หน้าจอแสดงคำอธิบายการเล่นเกมส์หน้าที่สอง เมื่อต้องการอ่านอธิบายเพิ่มเติมก็กดคีย์บอร์ดไปขวาเมื่อเต็ม หรือถ้าหากต้องการย้อนกลับก็ทำก็กดคีย์บอร์ดไปซ้าย

รูปที่ 5



หน้าจอแสดงคำอธิบายการเล่นเกมส์หน้าสุดท้าย เมื่ออ่านคำอธิบายเสร็จแล้วต้องการออกก็ทำการกดคีย์บอร์ดไปซ้ายเรื่อยๆจนถึงหน้า HOW TO? แล้วกดคีย์บอร์ด Q เพื่อออกไปยังหน้าเมนูหลัก

รูปที่ 6



หน้าจอแสดง Credit ของทีมผู้พัฒนาโปรแกรม

รูปที่ 7



หน้าจอแสดงตารางเกมส์ ขนาด 8\*8 ด้านบนตารางมีอักขรภาษาอังกฤษเพื่อบอกคอลัมน์ และด้านขวาของตารางมีตัวเลข เพื่อบอกหมายเลขแต่ละแถว

รูปที่ 8



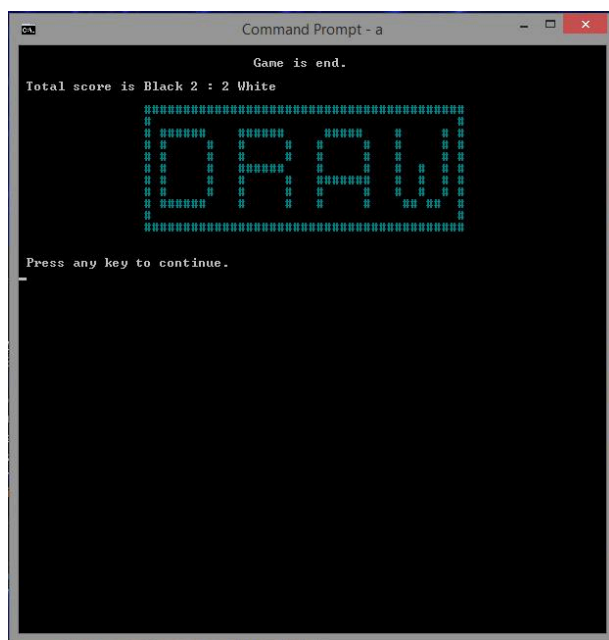
หน้าจอแสดงการจบเกมส์ BLACK WIN โดยที่ฝ่ายที่ชนะ คือ หมากสีดำ  
และมีการแสดงแต้มที่ชนะว่าอยู่ที่เท่าไรด้วย

รูปที่ 9



หน้าจอแสดงการจบเกมส์ WHILE WIN โดยที่ฝ่ายที่ชนะ คือ หมากสีขาว  
และมีการแสดงแต้มที่ชนะว่าอยู่ที่เท่าไรด้วย

รูปที่ 10



หน้าจอแสดงการจบเกมส์ DRAW โดยเสมอที่หมากทั้งสองสีนั้นเสมอกัน  
และมีการแสดงแต้มที่เสมอว่าอยู่ที่เท่าไรด้วย

## 2.2.4 วิธีการเล่น

รูปที่ 11



- หน้าจอแสดงตารางเกมส์ ขนาด 8\*8 ด้านบนตารางมีอักขระภาษาอังกฤษ เพื่อบอกคอลัมน์ และด้านขวาของตารางมีตัวเลข เพื่อบอกหมายเลขแต่ละแถว และ Enter black move หรือ Enter while move เพื่อบอกว่าครั้งต่อไปสีไหนเป็นผู้เริ่มหรือวางหมากลงก่อน
- วิธีการเล่นให้พิมพ์ตัวอักษรของคอลัมน์และเลขของแถวในการวางหมากในตำแหน่งที่ต้องการ เช่น ถ้าต้องการลงหมากในคอลัมน์ที่ C แถวที่ 4 ก็ให้พิมพ์คำว่า C1 ลงไป แล้วกด Enter
- การวางหมากจะเป็นการสลับกันป้อนตำแหน่งกันระหว่างสองสี ดังนั้นควรระมัดระวังในการลงหมากแต่ละครั้ง
- ถ้าหากป้อนตำแหน่งคอลัมน์และแถวผิดโปรแกรมก็จะรอรับค่าใหม่จนกว่าจะเป็นตำแหน่งที่ถูกต้อง
- และถ้าตำแหน่งคอลัมน์และแถวที่ป้อนลงไปนั้นไม่สามารถล้อมหมากของอีกฝั่งได้ โปรแกรมก็จะรอรับค่าใหม่จนกว่าตำแหน่งที่ป้อนจะสามารถล้อมหมากของอีกสีได้

- และถ้าตำแหน่งคอลัมน์และแถวที่ป้อนลงไปนั้นมีหมากลงอยู่แล้วก็จะไม่สามารถลงหมากในตำแหน่งนั้นได้อีก โปรแกรมก็จะรอรับค่าใหม่จนกว่าตำแหน่งที่ป้อนจะเป็นตำแหน่งที่ว่าง
- ถ้าหากต้องการออกจากเกมส์ก็ให้พิมพ์คำว่า Q1 และ Enter ตัวโปรแกรมก็จะออกเกมส์ แล้วแสดงผลแต้มและแสดงหน้าตาต่างผลการแข่งขันว่าสีใดเป็นผู้ชนะหรือเสมอกันทั้งสองสี
- เกมส์จะจบลงเมื่อตารางไม่มีที่ว่างหมากเพิ่ม แล้วแสดงผลแต้มและแสดงหน้าตาต่างผลการแข่งขันว่าสีใดเป็นผู้ชนะหรือเสมอกันทั้งสองสี



## บทที่3

### บทสรุป

#### 3.1 ผลที่ได้รับ

การที่ได้ทำโปรเจกต์เกมส์Othello ได้ช่วยฝึกการทำงานร่วมกับผู้อื่น ได้ฝึกการเขียนภาษาซีเพื่อพัฒนาเกมส์ และได้รับความรู้เพิ่มเติมจากนอกห้องเรียน

#### 3.2 ข้อดีและข้อเสีย

- ข้อดี**
1. ฝึกสมาธิและไหวพริบในการแก้ปัญหา
  2. ฝึกการคิดให้รอบคอบก่อนตัดสินใจ

- ข้อเสีย**
1. กราฟฟิกไม่สวยงามเนื่องจากใช้ระบบ Command line
  2. ประโยคภาษาอังกฤษต่างๆอาจไม่ถูกต้องเนื่องจากความรู้ด้านภาษาอังกฤษของผู้จัดทำยังไม่ดีพอ

### ภาคผนวก

```
#include <stdio.h>

#include <windows.h>

void othello();// Othello label

void menu();// menu 1 2 3 4

void gotoxy();// fix axis x & y

void credit();// credit

void htp();// how to play

void black_win();// black win label

void white_win();// white win label

void draw();// draw label

void table_8x8();// print table

void flip(int col, int row, int bow);// flip opponent stone

void play();// play game

void page();// page hoe to play

void htp_table();// table how to play

int checka(int col, int row, int bow);// check around of selected position can
land a stone

int checks(int col, int row, int bow, int x, int y);// check straight of selected
position can land a stone

int can_land(int bow);// check that turn can land a stone

int check_score_black();// check score of black
```

```
int check_score_white(); // check score of white
```

```
int game_end(); // check game is end?
```

```
int table[8][8], p=1; // array of table 8*8
```

```
/* main function */
```

```
int main()
```

```
{
```

```
    system("MODE 120,30");
```

```
    othello();
```

```
    menu();
```

```
    return 0;
```

```
}
```

```
void othello()
```

```
{
```

```
    system("cls");
```

```
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x0A);
```

```
    printf("\n");
```

```

printf("
#####
#####
##### ");

printf(" # _____ _# ");

printf(" # / ____ | / / / | / | / | / | /
| / ____ | # ");

printf(" # | | | | / ____ ____ / | | | | | ____ | | |
| | | | | | # ");

printf(" # | | | | | | | | ____ | | | ____ | | |
| | | | | # ");

printf(" # | | | | | | | | ____ | | | ____ | | |
| | | | | # ");

printf(" # | | | | | | | | ____ | | | ____ |
| | | | | # ");

printf(" # | ____ | | | | | | | | | ____ |
| ____ | | ____ | | # ");

printf(" # | ____ / | ____ / | ____ / | ____ /
| ____ / | ____ / | ____ / | ____ / # ");

printf(" #
# ");

printf("
#####
#####
##### ");

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x07);

```

```
}

void menu()

{
    int number;

    printf("\n\n\n");

    printf("\t\t\t\t\t1 - Play\n");

    printf("\t\t\t\t\t2 - How to Play\n");

    printf("\t\t\t\t\t3 - Credit\n");

    printf("\t\t\t\t\t4 - Exit\n");

    printf("\n\n\n");

    printf("\t\t\t\t\t Please You Choose : ");

    scanf("%d",&number);

    system("cls");

    system("MODE 75,50");

    if(number==1)

        play();

    else

        if(number==2)

        {

            http();

            page();

            main();
```

```
    }

    else

        if(number==3)

            {

                credit();

                getch();

                main();

            }

        else

            if(number!=4)

                main();

}

COORD coord = {0,0};

void gotoxy(int x, int y)

{

    coord.X = x; coord.Y = y;

    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);

}

void credit()

{

    system("MODE 120,30");
```

```

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x09);

printf("\n");

printf("
#####
#####
##### ");

printf(" #
# ");

printf(" # / | / | / / | / | /___/ /
/ # ");

printf(" # | | / / | |
___ /___ ___/ # ");

printf(" # | | / | |___ | | |
| | # ");

printf(" # | | / | |___ | | |
| | # ");

printf(" # | |___ | / | |___ | |
| | | # ");

printf(" # | | | | | / | | | |
# ");

printf(" # |___/ |___ |___/ |___/
|___ |___ |___/ # ");

printf(" #
# ");

printf("
#####

```

```
#####  
##### ");
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x07);
```

```
gotoxy(55,13);
```

```
printf("The Teams\n");
```

```
gotoxy(43,15);
```

```
printf("Ramet      Yangdum      57070098\n");
```

```
gotoxy(43,16);
```

```
printf("Worravej   Siraputpong  57070104\n");
```

```
gotoxy(43,17);
```

```
printf("Watthakorn  Malikow      57070106\n");
```

```
gotoxy(43,18);
```

```
printf("Sattra     Tansuphasiri 57070111\n");
```

```
gotoxy(43,19);
```

```
printf("Safwan     Paeroh       57070125\n");
```

```
}
```

```
void htp()
```

```
{
```

```
system("cls");
```

```
system("MODE 120,30");
```

```
p=1;
```

```
SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x09);
```



```

printf("\n");

printf("
#####
#####
##### ");

printf(" # _____
_____ # ");

printf(" # / | / | / _____ | / | _____ / | / / /
_____ | / _____ | # ");

printf(" # | | | | | | | | / | | | / _____ / | |
| | | | | _____ | | # ");

printf(" # | _____ | | | | | | | | | | | | | | |
| | | _____ | | # ");

printf(" # | _____ | | | | | | | | | | | | | | |
| | _____ | | # ");

printf(" # | | | | | | | | _____ | _____ | | | |
| _____ | # ");

printf(" # | | | | | | _____ | | | | | |
| _____ | # ");

printf(" # | _____ / | _____ / | _____ /
| _____ / | _____ / | _____ # ");

printf(" #
# ");

printf("
#####
#####
##### ");

```

```

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x0E);

gotoxy(55,15);

printf("HOW TO PLAY");

gotoxy(43,17);

printf("[ Press q : back to main menu ]\t\t\t\t\tNext >>>\n");

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x07);

}

```

```

void page()
{
    int n;

    do

    {        do

            {

                    do

                    {

                            n = getch(); // left=75   right=77

                    }while(n!=75 && n!=77 && n!='q');

            }while((n==75&&p==1) || (n==77&&p==4));

            if (n==77&&p!=4)

            {

                    p++;

                    system("cls");

```

```

        http_table();

        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x0E);

        printf("\n\n\t\t [ Press q : back to main menu
]n");

        if (p==4)

        {

            printf("\n<<< Back\t\t\t[ %d/3 ]n", p-1);

        }

        else

        {

            printf("\n<<< Back\t\t\t[ %d/3 ]\t\t\t\tNext
>>>n", p-1);

        }

        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x07);

        }

        if(n==75&&p!=1)

        {

            p--;

            system("cls");

            http_table();

            SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x0E);

```

```

printf("\n\n\t\t [ Press q : back to main menu
]\n");

printf("\n<<< Back\t\t\t[ %d/3 ]\t\t\t\tNext >>>\n",
p-1);

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x07);

    }

    if(n==75&& p<2)

    {

        htp();

    }

}while(n!='q');
}

void htp_table()

{

    int i,j;

    for(i=0;i<=7;i++)// set first table

        for(j=0;j<=7;j++)

            table[i][j] = 0;

    table[3][4] = 1;// set black

    table[4][3] = 1;// set black

    table[3][3] = 2;// set white

```

```

table[4][4] = 2;// set white

if(p==2)
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x0C);

    printf("\n\t\t\t\t\t (Score Black and White)\n");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    table_8x8();

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x0C);

    printf("\t\t\t(The Black player moves first)\n\n");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    printf("Enter \"t1\" to call table \"q1\" to End Game.");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x0C);

    printf(" (Can leave the game at anytime.)\n");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    printf("\n\t\tEnter black move (column row): ");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x0C);

    printf("-> Ex. c4, d3, e6, f5, etc.\n");

```

```

        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    }

    else if(p==3)

    {

        table[2][3] = 1;

        table[3][3] = 1;

        table_8x8();

        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x0C);

        printf("\t\t\t ^ (If Black move(c4)\", \n\t White chip(d4)
between Black chip(c4, e4) will change to Black");

        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

        printf("\n\nEnter \"t1\" to call table \"q1\" to End Game\n.");

        printf("\n\t\tEnter black move (column row): ");

        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x0C);

        printf("-> Turn Change to White.\n");

        printf("\nIf one player can't make a valid move, play passes back
to the other player\n");

        SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    }

    else if (p==4)

```

```

{

    int i,j;

    for (i=0;i<8;i++)

    {

        for (j=0;j<8;j++)

            table[i][j]=1;

    }

    for (i=1; i<7; i++)

        table[i][3]=2;

    table[2][2]=2;table[2][6]=2;

    table[3][4]=2;table[3][5]=2;

    table[4][5]=2;

    table[5][4]=2;

    table[6][4]=2;table[6][5]=2;table[6][6]=2;

    table_8x8();

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x0C);

    printf("\n    (If on one have turn or in bord have only one color,
game will end)\n");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    printf("\n\nEnter \"t1\" to call table \"q1\" to End Game\n.");

    printf("\n\t\t Black Wins!!");

```







```

printf("\t# # # # ##### # # # # #\n");

printf("\t# # # # # # # # # #\n");

printf("\t# # # # # # # #\n");

printf("\t# # # # ##### # # #####\n");

printf("\t# # # # # # # # #\n");

printf("\t# # # # # # # # #\n");

printf("\t# ## ## # # ##### ## #####\n");

printf("\t# #\n");

printf("\t#####\n");

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x0E);

printf("\t\t\t\t\t#####\n");

printf("\t\t\t\t\t#\n");

printf("\t\t\t\t\t# # ##### # #\n");

printf("\t\t\t\t\t# # # ## #\n");

printf("\t\t\t\t\t# # # # #\n");

printf("\t\t\t\t\t# # # # #\n");

printf("\t\t\t\t\t# # # # #\n");

printf("\t\t\t\t\t# # # # #\n");

printf("\t\t\t\t\t# # # # #\n");

printf("\t\t\t\t\t# # # # #\n");

printf("\t\t\t\t\t# # # # #\n");

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x07);

```

```

}

void draw()
{
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x03);

    printf("\t\t#####\n");

    printf("\t\t#                               #\n");

    printf("\t\t# #####      #####      #####      #      # #\n");

    printf("\t\t# #      # #      # #      # #      # #\n");

    printf("\t\t# #      # #      # #      # #      # #\n");

    printf("\t\t# #      # #####      #      # #      # #\n");

    printf("\t\t# #      # #####      # #      # #\n");

    printf("\t\t# #      # #      # #      # #      # #\n");

    printf("\t\t# #####      #      # #      # #\n");

    printf("\t\t#                               #\n");

    printf("\t\t#####\n");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x07);
}

/* function print table */

void table_8x8()
{

```

```

printf("\n\t\t");

char abc[9]="abcdefgh";

int i,j;

printf(" ");

for (i=0; i<8; i++)

{

    printf("  %c", abc[i]);

}

printf("    Black %d : %d white", check_score_black(),
check_score_white());

printf("\n ");

for (i=0;i<8;i++)

{

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    printf("\t\t ");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x20);

    printf("-----\n");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    printf("\t\t ");

    for(j=0;j<8;j++)

    {

```

```

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x20);

    printf("|");

    if (table[j][i]==1)

    {

        printf(" @ ");

    }

    else if (table[j][i]==2)

    {

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x27);

        printf(" @ ", table[j][i]);

    }

    else

        printf("  ");

    }

    printf("|");

    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE),
0x07);

    printf(" %d \n", i+1);

}

printf(" \t\t
");SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x20);

```

```

printf("-----\n\n");

SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), 0x07);

}

/* flip opponent stone */

void flip(int col, int row, int bow)

{
    int i,j,a,b;

    for(i=-1;i<=1;i++)

    {
        for(j=-1;j<=1;j++)

        {
            if((i!=0||j!=0) && checks(col, row, bow, i, j))

            {
                a=col;

                b=row;

                while(table[a+i][b+j]!=bow) // flip opponent target
stone
            {

                table[a+i][b+j]=bow;

                a+=i;

                b+=j;

            }

```

```

        }

    }

}

}

/* play */

void play()

{

    int check=1,col,row,turn=1,again=0,bow,a,b,i,j; // bow is black or white
    stone that turn, a and b is observer of col and row

    char cols[2]; // string of column

    for(i=0;i<=7;i++)// set first table

        for(j=0;j<=7;j++)

            table[i][j] = 0;

    table[3][4] = 1;// set black

    table[4][3] = 1;// set black

    table[3][3] = 2;// set white

    table[4][4] = 2;// set white

    printf("\n\n");

```

```

table_8x8();// first table

do{

    if((cols[0]==113&&row==0)||game_end())// check game is end or
exit input

    {

        system("cls");

        printf("\n\t\t\t\t\t Game is end. \n\n");

        table_8x8();

        printf("\n\n Press any key to continue.\n");

        getch();

        system("cls");

        printf("\n\t\t\t\t\t Game is end. \n\n");

        printf(" Total score is Black %d : %d White\n\n",
check_score_black(), check_score_white());

        if(check_score_black() == check_score_white())

            draw();

        else

            if (check_score_black() > check_score_white())

                black_win();

            else

                white_win();

```



```

        printf("\n\n Press any key to continue.\n");

        getch();

        main();

        check=0;

    }

    else

    {

        if(turn==1||can_land(bow))

        {

            if(again==0)

            {

                printf(" Enter \"t1\" to call table \"q1\" to
End Game.\n\n\t\t Enter %s move (column row): ",
turn%2==0?"white":"black"); // print presenter of input

            }

            else

            {

                printf("\t\t Enter %s again (column row): ",
turn%2==0?"white":"black"); // if input is wrong it will be work

            }

            scanf("%1s%1d", cols,&row); // input column in
string form and row in integer form

```

```

        col = cols[0]-96; //change char to integer

        again++; // set again more than 0 for check this turn
have to input again or not?

        col--; // change integer to index

        row--; // change integer to index

        bow=turn%2==0?2:1; // turn of black or white

    }

}

if(can_land(bow)) // check this turn can play?

{

    if(table[col][row]==0 && 0<=col&&col<=7 &&
0<=row&&row<=7 && checka(col, row, bow)) // check selected position can land
a stone

    {

        // land and flip //

        table[col][row] = bow; // land a stone

        flip(col, row, bow); // flip opponent stone

        again=0; // set again to 0

    }

    if(again==0) // if this turn worked true will go next turn

    {

        system("cls");

```

```

        printf("\n\t\t\t\t %s move: %s%d\n",
turn%2==0?"white":"black", cols, row+1); // print position of this turn

        table_8x8();

        turn++; // set turn to next turn

    }

}

else

{

    turn++; // set turn to next turn

}

if(cols[0]==116&&row==0)

{

    system("cls");

    printf("\n\n");

    table_8x8();

    printf(" Enter \"t1\" to call table \"q1\" to End Game.\n\n");

}

bow=turn%2==0?2:1;

}while(check==1);

}

```

```

/* check around of selected position can land a stone */

int checka(int col, int row, int bow)

{ // check around

    int flag=0,i,j;

    for(i=-1;i<=1;i++)

    {

        for(j=-1;j<=1;j++)

        {

            if((i!=0||j!=0) && checks(col, row, bow, i, j))

            {

                if(checks(col, row, bow, i, j))

                    flag=1;

            }

        }

    }

    return flag;

}

```

```

/* check straight of selected position can land a stone */

int checks(int col, int row, int bow, int x, int y)

{

    int flag=0,fst=0;

```

```

while(flag==0 && col<=7&&row<=7 && col>=0&&row>=0)

{

    if(fst++==0)

    {

        if(table[col+x][row+y]==bow || table[col+x][row+y]==0)// if
next step is same or space will break while loob and return flag = 0

        {

            break;

        }

    }

    else

    {

        if(table[col+x][row+y]==0)// while function is working if
next step is space will break while loob and return flag = 0

        {

            break;

        }

        else

        {

            if(table[col+x][row+y]==bow&&col+x<=7&&row+y<=7&&col+x>=0&&row+
y>=0) // check next step untill find same stone and return flag = 1

            {

                flag=1;

```

```

        }

    }

    }

    col+=x;

    row+=y;

}

return flag;// flag=1 is can land stone in that space

}

```

/\* check this turn can land a stone \*/

```

int can_land(int bow)
{
    int flag=0,i,j;

    for(i=0;i<=7;i++)
    {
        for(j=0;j<=7;j++)
        {
            if(table[i][j]==0&&checka(i, j, bow))
                flag=1;
        }
    }

    return flag;
}

```

```
}

/* check score of black*/

int check_score_black()
{
    int black=0,q,p;

    for(q=0;q<=7;q++)
    {
        for(p=0;p<=7;p++)
        {
            if(table[q][p] == 1)
                black++;
        }
    }

    return black;
}
```

```
/* check score of white */

int check_score_white()
{
    int white=0,q,p;

    for(q=0;q<=7;q++)
    {
```

```
        for(p=0;p<=7;p++)  
        {  
            if(table[q][p] == 2)  
                white++;  
        }  
    }  
    return white;  
}
```

```
/* check game is end? */
```

```
int game_end()  
{  
    int flag=1;  
    if(can_land(1) || can_land(2)) // if both stone can't land return flag = 0  
        flag=0;  
    return flag;  
}
```