

การทดลองที่ 6 : พื้นฐานการเขียน Text-mode Game ครั้งที่ 2

จุดประสงค์

นักศึกษาสามารถตกแต่ง Text-mode Game ให้มีความน่าสนใจโดยการควบคุมจอภาพแสดงผลให้มีการใช้สี และปรับแต่ง cursor ได้ พร้อมทั้งควบคุมการเคลื่อนไหวของวัตถุต่างๆ ในเกมได้

ข้อกำหนด

- โปรแกรมมีการพัฒนาและทดสอบบน Visual C++ 2019 Edition การปฏิบัติการโดยใช้ compiler ตัวอื่นๆ จะต้องมีการแก้ไขโปรแกรม
- ในการทดลองนี้เป็นการทดลองต่อเนื่องจาก การทดลองที่ 4 ซึ่งจะใช้ source code พื้นฐานเป็นจุดเริ่มต้นของการทดลองนี้ ดังนี้

```
#include<stdio.h>
#include<windows.h>
#include<conio.h>

void gotoxy(int x, int y)
{
    COORD c = { x, y };
    SetConsoleCursorPosition( GetStdHandle(STD_OUTPUT_HANDLE), c);
}

void draw_ship(int x,int y)
{
    gotoxy(x,y); printf("<-0-> ");
}

int main()
{
    char ch='.';
    int x=38,y=20;
    draw_ship(x,y);
    do {
        if(kbhit()){
            ch=getch();
            if(ch=='a'){draw_ship(--x,y);}
            if(ch=='s'){draw_ship(++x,y);}
            fflush(stdin);
        }
    }
```

```

        Sleep(100);

    } while(ch!='x');

return 0;
}

```

โดยเมื่อรันโปรแกรมดังกล่าวโปรแกรมจะรอรับการกดปุ่ม โดยยานจะเคลื่อนที่ไปทางซ้าย,ขวา ที่ละตัวอักษรเมื่อกดปุ่ม a , s ตามลำดับ

ตอนที่ 1 การตกแต่งเกมส์

1. การลบ Cursors

เมื่อนักศึกษารันโปรแกรมบนหน้าจอ console application (console = จอภาพที่มีแต่ text) จะมี cursor ปรากฏอยู่เสมอ ซึ่งเมื่อมีการย้ายตำแหน่งการแสดงผลโดยใช้คำสั่ง gotoxy() จะมี cursor ย้ายไปปรากฏในตำแหน่งอื่นๆ บนหน้าจอ ซึ่งจะดูรกจนสายตาและทำให้โปรแกรมแสดงผลไม่สวยงาม ดังนั้นจึงควรมีการปิดการแสดงผล cursor ก่อนการทำงาน ซึ่งเราสามารถใส่ชุดคำสั่งของฟังก์ชัน setcursor() เพื่อปิด-เปิดการแสดงผล cursor ในหน้าจอได้

```

void setcursor(bool visible)
{
    HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_CURSOR_INFO lpCursor;
    lpCursor.bVisible = visible;
    lpCursor.dwSize = 20;
    SetConsoleCursorInfo(console, &lpCursor);
}

```

โดยเมื่อเรียกใช้คำสั่ง setcursor(0) โปรแกรมจะไม่แสดงผล cursor และเมื่อเรียกคำสั่ง setcursor(1) โปรแกรมจะแสดงผล cursor ที่ตำแหน่งปัจจุบัน จึงควรมีการเรียกใช้ setcursor(0) ใน main() หลังการประกาศตัวแปรเป็นคำสั่งแรกเพื่อให้ cursor จะหายไป

2. การเปลี่ยนสีของ text ต่างๆ

เพื่อความสวยงามของเกมส์จึงควรมีการใช้สีเพื่อตกแต่งเกมส์ให้น่าสนใจ ในการกำหนดค่าสีให้กับตัวอักษรต่างๆ ใน console application สามารถทำได้ทั้งการกำหนดค่าสีของตัวอักษร และการกำหนดค่าสีของพื้นหลัง โดยเราสามารถใส่ชุดคำสั่งของฟังก์ชัน setcolor() เพื่อกำหนดค่าสีก่อนการแสดงผลได้ ซึ่งรายละเอียดของฟังก์ชัน setcolor() มีดังนี้

```

void setcolor(int fg,int bg)

```

```
{
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hConsole, bg*16+fg);
}
```

โดยฟังก์ชัน setcolor() ต้องการพารามิเตอร์ 2 ตัวคือค่าสีของตัวอักษร (foreground) และค่าสีของพื้นหลัง background สำหรับค่ารหัสสีของสีต่างๆ ที่ตั้งค่า foreground, background จะมีค่าดังตาราง

รหัสสี	สี	รหัสสี	สี
0	Black	8	Gray
1	Blue	9	Light Blue
2	Green	A	Light Green
3	Aqua	B	Light Aqua
4	Red	C	Light Red
5	Purple	D	Light Purple
6	Yellow	E	Light Yellow
7	White	F	Bright White

เช่นเมื่อมีการเรียกใช้งาน function setcolor (6,8) แล้ว คำสั่ง printf ที่ตามมาจะมีการแสดงผล ตัวอักษรสีเหลืองบนพื้นหลังสีเทา

สำหรับการลบลวัตถุในเกมนั้น ในกรณีที่ไม่มีค่าสีพื้นหลังสามารถลบลวัตถุได้โดยการพิมพ์ช่องว่าง “ ” แทนตัวอักษรเดิม แต่ในกรณีที่มีการตั้งค่าสีพื้นหลังจะต้องลบลวัตถุโดยการพิมพ์ช่องว่าง พร้อมทั้งกำหนดค่าพื้นหลังเป็นสีดำ (background = 0) จึงจะลบลวัตถุได้ทั้งหมด

Assignment 1 : ให้นักศึกษาปิด cursor และ เปลี่ยนสียานเป็น สีเขียว และ background สีแดง เมื่อยานเคลื่อนที่จะไม่ทิ้งรอยของ background

ตอนที่ 2 การเคลื่อนที่ในเกม

ในการทำให้วัตถุใดๆ ในเกมเคลื่อนไหวนสามารถทำได้ง่ายๆ โดยการลบลวัตถุโดยการพิมพ์ช่องว่างทับตัวอักษรเดิม แล้วไปพิมพ์วัตถุที่ตำแหน่งใหม่ ซึ่งเราจะมี Sleep() ระหว่างการแสดงผลวัตถุกับการลบลวัตถุออกเพื่อให้ภาพติดตาและไม่มีการกะพริบ ใน source code พื้นฐานจะมีการแสดงผลยานเคลื่อนที่ใน Loop หลัก โดยจะสั่งให้ยานเคลื่อนที่เฉพาะตอนที่มีการกดปุ่มเท่านั้น ดัง pseudo code ตัวอย่าง

```
Loop
{
    ตรวจสอบการกดปุ่ม
    {
        ตรวจสอบว่ามีปุ่มเลื่อนซ้าย -> วาดรูปยานเลื่อนไปทางซ้าย 1 ตำแหน่ง
```

```

        ตรวจสอบว่ามีปั้มเลื่อนขวา -> วาดรูปยานเลื่อนไปทางขวา 1 ตำแหน่ง
    }
} จนกว่าจะมีการกดปุ่มออกจาก Loop
    
```

แต่ในกรณีที่จะทำให้วัตถุใดๆ ในเกมส์สามารถเคลื่อนที่ได้อัตโนมัติจะมีการเปลี่ยนแปลงการทำงานเล็กน้อย ดังนี้

- แต่ละวัตถุจะต้องมีตัวแปรเก็บทิศทางของตัวเอง
- เมื่อมีการกดปุ่ม จะตรวจสอบการกดปุ่มแล้วทำการเปลี่ยนค่าตัวแปรทิศทาง แทนการเปลี่ยนตำแหน่งวัตถุ
- เพิ่มการตรวจสอบทิศทางของวัตถุ แล้วแสดงผลวัตถุที่ตำแหน่งใหม่
โดยการทำงานที่เปลี่ยนแปลงไปจะมีรายละเอียด pseudo code ดังนี้

```

ทิศทาง = ไม่เคลื่อนที่
Loop
{
    ตรวจสอบการกดปุ่ม
    {
        ตรวจสอบว่ามีปั้มเลื่อนซ้าย -> ทิศทาง = ซ้าย
        ตรวจสอบว่ามีปั้มเลื่อนขวา -> ทิศทาง = ขวา
        ตรวจสอบว่ามีปั้มหยุด -> ทิศทาง = ไม่เคลื่อนที่
    }
    ตรวจสอบค่าทิศทาง
    {
        ทิศทาง = ซ้าย -> ลบวัตถุตำแหน่งปัจจุบัน , แสดงวัตถุด้านซ้าย
        ทิศทาง = ขวา -> ลบวัตถุตำแหน่งปัจจุบัน , แสดงวัตถุด้านขวา
    }
}
} จนกว่าจะมีการกดปุ่มออกจาก Loop
    
```

Assignment 2 : ให้นักศึกษาเพิ่มเติมโปรแกรม เพื่อให้ยานเคลื่อนที่อัตโนมัติ โดยเมื่อกดปุ่ม a ยานจะเคลื่อนที่ไปทางซ้ายจนสุดจอภาพด้านซ้าย กดปุ่ม d ยานจะเคลื่อนที่ไปทางขวาจนสุดจอภาพด้านขวา และ กด s ยานจะหยุดเคลื่อนที่

ตอนที่ 3 การยิงกระสุน

ภายในเกมส์จะมีวัตถุที่มีลักษณะพิเศษที่ไม่ได้อยู่ภายในหน้าจอของเกมส์อยู่ตลอดเวลา เช่น กระสุน เป้า บิน ศัตรู อุกาบาต ฯลฯ โดยวัตถุดังกล่าวจะมีการแสดงผลแล้วหายไปตามเงื่อนไขเช่นเกิดขึ้นเมื่อมีการกดปุ่ม และหายไปเมื่อออกนอกขอบเขตของจอภาพ หรือ ถูกทำลายจากการชนกัน เป็นต้น สำหรับการเขียน

ชื่อ-นามสกุล รหัสประจำตัวนักศึกษา

โปรแกรมให้วัตถุต่างๆ มีการเคลื่อนที่ได้โดยอัตโนมัติตามเงื่อนไขการแสดงผลเช่นเมื่อมีการกดปุ่มจะมีขั้นตอนดังนี้

1. กำหนดสถานะของวัตถุว่าอยู่ในสถานะอะไร (มีหรือไม่มีการแสดงผล)
2. ภายใน Loop การทำงานหลัก...
 - a. ตรวจสอบว่ามีการกดปุ่มให้วัตถุแสดงผล -> เปลี่ยนสถานะของวัตถุเป็นแสดงผล , กำหนดค่าตำแหน่งเริ่มต้นของวัตถุ
 - b. ตรวจสอบว่าวัตถุอยู่ในสถานะแสดงผล และตำแหน่งอยู่ในขอบเขตหรือไม่ -> ให้เคลื่อนที่ตามทิศทางที่กำหนด
(ถ้าไม่ใช่) -> ให้เปลี่ยนสถานะของวัตถุเป็นไม่มีการแสดงผล

สำหรับวัตถุเช่นกระสุน จะมีการเคลื่อนที่ในทิศทางเดียว ไปจนกว่าจะพ้นขอบเขตแสดงผลของจอภาพ จึงลบวัตถุออกจากจอภาพ pseudo code ของการแสดงผลกระสุนเมื่อมีการกดปุ่มมีดังนี้

กำหนด สถานะกระสุน = OFF

Loop

{ ตรวจสอบการกดปุ่ม

 {

ตรวจสอบว่ามีปุ่มกดยิง ที่สถานะกระสุนเป็น OFF -> เปลี่ยน สถานะกระสุน = ON .

กำหนดตำแหน่ง x,y ของกระสุน

 }

ตรวจสอบว่าสถานะของกระสุน = ON {

 { ลบกระสุนตำแหน่งปัจจุบัน

ตรวจสอบว่ากระสุนยังอยู่ในกรอบการแสดงผล -> แสดงกระสุนที่ตำแหน่งใหม่

(ถ้าไม่ใช่) -> Status ยิง = OFF

 }

} จนกว่าจะมีการกดปุ่มออกจาก Loop

Assignment 3 : ให้นักศึกษาเพิ่มเติมโปรแกรมให้เมื่อมีการกดปุ่ม ‘ ‘ จะมีการยิงกระสุนจากยานและเคลื่อนที่ขึ้นจนสุดจอภาพด้านบน

Assignment 4 : ให้นักศึกษาแก้ไขโปรแกรมให้ยานสามารถยิงกระสุนได้ถึง 5 นัด

การส่งงาน : ให้นักศึกษาส่งโปรแกรมที่สมบูรณ์ของ Assignment 4 เพียงโปรแกรมเดียวเท่านั้น