

การทดลองที่ 7 : พื้นฐานการเขียน Text-mode Game ครั้งที่ 3

จุดประสงค์

นักศึกษาสามารถพัฒนาเกมในส่วนของการสุ่มตัวเลข การสร้างเสียง และการตรวจสอบการชนกันของวัตถุ

ข้อกำหนด

- โปรแกรมมีการพัฒนาและทดสอบบน Visual C++ 2019 Edition การปฏิบัติการโดยใช้ compiler ตัวอื่นๆ จะต้องมีการแก้ไขโปรแกรม
- ในการทดลองนี้เป็นการทดลองต่อเนื่องจาก การทดลองที่ 5 ซึ่งจะใช้ source code พื้นฐานเป็นจุดเริ่มต้นของการทดลองนี้ ดังนี้

```
#include<stdio.h>
#include<windows.h>
#include<time.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-> ");
}

void draw_bullet(int x,int y)
{
    gotoxy(x,y); printf("^");
}

void clear_bullet(int x,int y)
{
    gotoxy(x,y); printf("");
}

int main()
{
```

```

char ch='.';
int x=38,y=20;
int bx,by,i;
int bullet =0;
draw_ship(x,y);
do {
    if(_kbhit){
        ch=_getch();
        if(ch=='a'){draw_ship(-x,y);}
        if(ch=='s'){draw_ship(++x,y);}
        if(bullet!=1 && ch==' '){bullet=1; bx=x+3; by=y-1;}
        fflush(stdin);
    }
    if(bullet=1){
        clear_bullet(bx,by);
        if(by=2){ bullet=0;} else { draw_bullet(bx,-by); }
    }

    Sleep(100);

} while(ch!='x');

return 0;
}

```

โดยเมื่อรันโปรแกรมดังกล่าวโปรแกรมจะรอรับการกดปุ่ม โดยยานจะเคลื่อนที่ไปทางซ้าย,ขวา เมื่อกดปุ่ม a , s ตามลำดับ สามารถยิงกระสุนได้ทีละ 1 นัดโดยกระสุนจะเคลื่อนที่ไปยังด้านบนของจอภาพ และจะหยุดโปรแกรมเมื่อกดปุ่ม x

ตอนที่ 1 การสุ่มค่า

การสุ่มตัวเลขเป็นฟังก์ชันหลักที่มีการใช้งานบ่อยครั้งในเกม เพื่อให้เกมมีความน่าสนใจมากขึ้น เช่น การสุ่มจุดที่เกิดเป้าหมาย สุ่มเหตุการณ์ต่างๆ ในเกม สุ่มทิศทางการเคลื่อนที่ ในการสุ่มตัวเลขจะเกี่ยวข้องกับฟังก์ชัน 2 ฟังก์ชันได้แก่ ฟังก์ชัน rand() และฟังก์ชัน srand() สำหรับฟังก์ชัน rand() เป็นฟังก์ชันที่ใช้ในการสุ่มค่า ซึ่งจะต้องมีการประกาศ #include<windows.h> ก่อนใช้งาน ผลลัพธ์ของฟังก์ชัน rand() จะได้ตัวเลขสุ่มที่ได้จากอัลกอริทึมสุ่มเทียม (Pseudo Random Algorithm) แต่มีข้อจำกัดเนื่องจากการเป็นค่าวนจากอัลกอริทึมการสุ่มเทียม ทำให้การรันโปรแกรมทุกครั้งจะได้ลำดับของเลขสุ่มเหมือนเดิมทุกครั้ง ยกตัวอย่างเช่น

```

#include<stdio.h>
#include<windows.h>
int main()
{
    int i;
    for(i=0;i<10;i++)printf("%d\n",rand());
    return 0;
}

```

เมื่อมีการรันโปรแกรมตัวอย่างจะมีการแสดงผลตัวเลขสุ่มจำนวน 10 ตัวซึ่งเกิดจากเลขสุ่มจำนวน 10 รอบ แต่ละรอบมีการแสดงผลค่าที่ได้จากฟังก์ชัน rand() ซึ่งเราจะพบว่าฟังก์ชัน rand() นี้ให้ผลเป็นเลขสุ่มได้จริง แต่สิ่งที่โปรแกรมเมอร์ต้องการคือการรันโปรแกรมแต่ละครั้งต้องได้ผลลัพธ์ที่แตกต่างกัน โปรแกรมจึงน่าสนใจ ในการทำให้ฟังก์ชัน rand() สุ่มค่าไม่เหมือนกันทุกครั้งที่รันโปรแกรมจะต้องมีการกำหนดค่า seed ที่แตกต่างกันทุกครั้งที่รันโปรแกรม ซึ่ง seed เป็นพารามิเตอร์ใน pseudo random algorithm ถ้ามีค่า seed ต่างกัน จะทำให้ค่าเริ่มต้นของการสุ่มของ rand() แตกต่างกัน การกำหนดค่า seed สามารถทำได้โดยใช้ฟังก์ชัน srand() อย่างไรก็ตามถ้าเรากำหนดค่า seed โดยใช้ค่าคงที่ เราก็จะได้ลำดับตัวเลขสุ่มที่เหมือนเดิมทุกครั้งเหมือนเดิม เพื่อให้การรันโปรแกรมแต่ละครั้งมีค่าชุดตัวเลขสุ่มไม่เหมือนกัน จะต้องตั้งค่า seed ที่แตกต่างกัน ค่าที่เหมาะสมที่สุดที่จะใช้เป็น seed จึงเป็นค่าเวลาของเครื่อง เพราะการรันแต่ละครั้งเป็นการรันโปรแกรมคนละเวลากัน เราจึงใช้ฟังก์ชัน time() เพื่อดึงค่าเวลาของระบบมาเป็น seed (ฟังก์ชัน time() ต้อง #include<time.h>) ยกตัวอย่างการสุ่มตัวเลขที่ทำให้ได้ค่าการสุ่มแตกต่างกันทุกครั้งที่รันโปรแกรกดังนี้

```
#include<stdio.h>
#include<windows.h>
#include<time.h>
int main()
{
    int i;
    srand(time(NULL));
    for(i=0;i<10;i++)printf("%d\n",rand());
    return 0;
}
```

ให้นักศึกษาลองรันโปรแกรมนี้อีกหลายๆ ครั้ง จะพบว่าค่าตัวเลขสุ่มมีค่าแตกต่างกันทุกครั้ง ในการใช้ฟังก์ชัน srand() ในการกำหนดค่าเริ่มต้นของการสุ่มให้การทำงานของฟังก์ชัน rand() ได้ผลลัพธ์แตกต่างกันทุกครั้งที่รันโปรแกรม จะทำเพียงครั้งเดียวก่อนการเรียก rand() และไม่มีการเรียก srand() ในฟังก์ชันย่อย เราจึงมักเห็นว่าฟังก์ชัน srand() จะอยู่บรรทัดแรกๆ ของฟังก์ชัน main() เสมอ

Assignment 1 : ให้นักศึกษาเพิ่มเติม source code พื้นฐานโดย สร้างดาว “*” แบบสุ่ม 20 ดวง โดยแสดงผลในช่วงบรรทัดที่ 2 ถึง 5 และตัวอักษรที่ 10-70 ในแต่ละครั้งที่รันโปรแกรมจะให้ตำแหน่งของดาวที่แตกต่างกัน

ตอนที่ 2 การเล่นเกมเสียง

speaker เป็นอุปกรณ์พื้นฐานของคอมพิวเตอร์ที่สามารถส่งเสียงได้ แต่การส่งเสียงออกจาก speaker จะสามารถทำได้เพียงส่งคลื่นความถี่ในช่วงที่มนุษย์สามารถได้ยินได้เท่านั้น ในการส่งให้มีเสียงออกจาก speaker สามารถทำได้โดยใช้ฟังก์ชัน Beep() โดยฟังก์ชันนี้ ต้องการพารามิเตอร์ 2 ตัวคือความถี่ที่จะสร้าง

ชื่อ-นามสกุล รหัสประจำตัวนักศึกษา
ออกที่ลำโพง (Hz) และระยะเวลาที่จะให้ออกเสียง (ms) ยกตัวอย่างเช่น Beep(700,100) จะส่งคลื่นความถี่ 700 Hz ออกทาง Speaker เป็นเวลา 100 ms ในการใช้คำสั่งนี้จะมีข้อกำหนดดังนี้

- เปิดลำโพง
- speaker ของแต่ละคนคุณภาพไม่เท่ากัน บางช่วงความถี่อาจไม่สามารถออกเสียงได้
- ถ้าช่วงเวลาที่ให้เสียงออกน้อยเกินไป เสียงจะไม่ออก
- ต้อง #include<windows.h>

Assignment 3 : ให้นักศึกษาเพิ่มเติมโปรแกรมให้มีเสียงเมื่อกระสุนเคลื่อนที่

ตอนที่ 3 การตรวจสอบการชนกันของวัตถุ

ในการตรวจสอบการชนกันของวัตถุนั้น สามารถทำได้โดยการตรวจสอบว่าตำแหน่งที่วัตถุจะเคลื่อนที่ไปนั้น มีวัตถุอื่นอยู่หรือไม่ ซึ่งเราสามารถเรียกดูค่าตัวอักษรในตำแหน่งที่ระบุโดยใช้ฟังก์ชัน cursor() ดังตัวอย่าง

```
char cursor(int x, int y){
    HANDLE hStd = GetStdHandle(STD_OUTPUT_HANDLE);
    char buf[2]; COORD c = {x,y}; DWORD num_read;
    if(
!ReadConsoleOutputCharacter(hStd,(LPTSTR)buf,1,c,(LPDWORD)&num_read))
        return '\0';
    else
        return buf[0];
}
```

เมื่อมีการเรียกฟังก์ชัน cursor() จะมีการระบุตำแหน่ง (x,y) บนหน้าจอ โดย y คือค่าบรรทัด และ x คือค่าตำแหน่งตัวอักษรในบรรทัด แล้วจะส่งค่ากลับคือตัวอักษร (char) ถ้าบนจอภาพ บรรทัดแรกมีการแสดงผล “ABCDEF” จะได้ผลลัพธ์ของการเรียก cursor(0,0) คือ ‘A’ , ผลลัพธ์ของการเรียก cursor(2,0) คือ ‘C’ จาก source code พื้นฐาน หากต้องการตรวจสอบว่ากระสุนยิงถูกดาวหรือไม่ สามารถทำได้ในส่วน of โปรแกรมที่ควบคุมการเคลื่อนที่ของกระสุนดังนี้

```
if(bullet=1){
    clear_bullet(bx,by);
    if(by=2){ bullet=0;} else { draw_bullet(bx,-by); }
```

ชื่อ-นามสกุล รหัสประจำตัวนักศึกษา

โดยเพิ่มเติมการตรวจสอบก่อน draw_bullet(bx,-by); ว่า cursor(bx,by-1) เป็น * หรือไม่ ถ้าเป็น * แสดงว่ามีการชนกันเกิดขึ้น นักศึกษาสามารถเขียนโปรแกรมให้กระสุนและดาวหายไปเมื่อมีการชนกันพร้อมเสียง Beep ได้

Assignment 3 : ให้นักศึกษาเพิ่มเติมโปรแกรมให้มีการแสดงผลคะแนนที่มุมบนขวาของจอภาพ โดยคะแนนจะเพิ่มขึ้น เมื่อมีการยิงกระสุนไปทำลายดาว และจะมีการสลับตำแหน่งการเกิดดาวใหม่ทุกครั้งที่ถูกทำลาย พร้อมทั้งมีเสียง Beep ถ้ามีการยิงกระสุนโดนดาว

การส่งงาน : ให้นักศึกษาส่งโปรแกรมที่สมบูรณ์ของ Assignment 3 เพียงโปรแกรมเดียวเท่านั้น