

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

GitHub URL : <https://github.com/nkd3v/programming-fundamental/tree/master/labs/lab6>

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

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

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

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

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

```
void draw_ship(int x, int y) {
    setcolor(2, 4);
    gotoxy(x, y);
    printf(" <-0-> ");
    setcolor(7, 0);
}

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

int main() {
    setcursor(0);
    char ch = '.';
    int x = 38, y = 20;
    draw_ship(x, y);

    do {
        if (kbhit()) {
            ch = getch();
            if (ch == 'a') {
                erase_ship(x, y);
                draw_ship(--x, y);
            }
            if (ch == 's') {
                erase_ship(x, y);
                draw_ship(++x, y);
            }
            fflush(stdin);
        }
        Sleep(100);
    } while (ch != 'x');

    return 0;
}
```

Assignment 2 : ให้นักศึกษาเพิ่มเติมโปรแกรม เพื่อให้ยานเคลื่อนที่อัตโนมัติโดยเมื่อกดปุ่ม a ยานจะเคลื่อนที่ไปทางซ้ายจนสุดจอภาพด้านซ้าย กดปุ่ม d ยานจะเคลื่อนที่ไปทางขวาจนสุดจอภาพด้านขวา และ กดr ยานจะหยุดเคลื่อนที่

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

#define SHIP_HEIGHT 1
#define SHIP_WIDTH 7
```

```
#define SCREEN_HEIGHT 23
#define SCREEN_WIDTH 80

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

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

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

void draw_ship(int x, int y) {
    setcolor(2, 4);
    gotoxy(x, y);
    printf(" <-0-> ");
    setcolor(7, 0);
}

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

int main() {
    setcursor(0);
    char ch = '.';
    int d = 0;
    int x = 0, y = 20;
    draw_ship(x, y);

    do {
        if (kbhit()) {
            ch = getch();
            if (ch == 'a') d = -1;
            if (ch == 's') d = 0;
            if (ch == 'd') d = 1;
            fflush(stdin);
        }
    }
```

```
        if (d != 0) {
            erase_ship(x, y);
            if (x + d >= 0 && (x + SHIP_WIDTH - 1) + d < SCREEN_WIDTH)
                x += d;
            draw_ship(x, y);
        }

        Sleep(100);
    } while (ch != 'x');

    return 0;
}
```

Assignment 3 : ให้นักศึกษาเพิ่มเติมโปรแกรมให้เมื่อมีการกดปุ่ม ‘ ‘ จะมีการยิงกระสุนจากยานและเคลื่อนที่ขึ้นจนสุดจอภาพด้านบน

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

#define SHIP_HEIGHT 1
#define SHIP_WIDTH 7
#define SCREEN_HEIGHT 23
#define SCREEN_WIDTH 80

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

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

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

void draw_ship(int x, int y) {
    setcolor(2, 4);
    gotoxy(x, y);
```

```
    printf(" <-0-> ");
    setcolor(7, 0);
}

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

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

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

int main() {
    setcursor(0);
    char ch = '.';
    int d = 0;
    int x = 38, y = 20;
    draw_ship(x, y);

    int bx = 0, by = 0;
    bool bVisible = false;

    do {
        if (kbhit()) {
            ch = getch();
            if (ch == 'a') d = -1;
            if (ch == 's') d = 0;
            if (ch == 'd') d = 1;

            if (ch == ' ' && !bVisible) {
                bx = x + SHIP_WIDTH / 2;
                by = y - 1;
                bVisible = true;
                draw_bullet(bx, by);
            }

            fflush(stdin);
        }

        if (bVisible) {
            erase_bullet(bx, by);
        }
    }
```

```
        if (by - 1 >= 0) draw_bullet(bx, --by);
        else bVisible = false;
    }

    if (d != 0) {
        erase_ship(x, y);
        if (x + d >= 0 && (x + SHIP_WIDTH - 1) + d < SCREEN_WIDTH)
            x += d;
        draw_ship(x, y);
    }

    Sleep(100);

} while (ch != 'x');

return 0;
}
```

Assignment 4 : ให้นักศึกษาแก้ไขโปรแกรมให้ยานสามารถยิงกระสุนได้ถึง 5 นัด

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

#define SHIP_HEIGHT 1
#define SHIP_WIDTH 7
#define SCREEN_HEIGHT 23
#define SCREEN_WIDTH 80
#define BULLET_COUNT 5

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

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

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

void draw_ship(int x, int y) {
```

```
        setcolor(2, 4);
        gotoxy(x, y);
        printf(" <-0-> ");
        setcolor(7, 0);
    }

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

struct Bullet {
    bool active = false;
    int x, y;
};

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

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

int main() {
    setcursor(0);
    char ch = '.';
    int d = 0;
    int x = 38, y = 20;
    draw_ship(x, y);

    Bullet bullets[BULLET_COUNT];

    do {
        if (kbhit()) {
            ch = getch();
            if (ch == 'a') d = -1;
            if (ch == 's') d = 0;
            if (ch == 'd') d = 1;

            if (ch == ' ') {
                for (int i = 0; i < BULLET_COUNT; i++) {
                    if (!bullets[i].active) {
                        bullets[i].active = true;
                        bullets[i].x = x + SHIP_WIDTH / 2;
                        bullets[i].y = y - 1;
                    }
                }
            }
        }
    } while (ch != 'q');
```

```
                draw_bullet(bullets[i].x, bullets[i].y);
                break;
            }
        }
    }

    fflush(stdin);
}

for (int i = 0; i < BULLET_COUNT; i++) {
    if (!bullets[i].active) continue;
    erase_bullet(bullets[i].x, bullets[i].y);
    if (bullets[i].y - 1 >= 0)
        draw_bullet(bullets[i].x, --bullets[i].y);
    else
        bullets[i].active = false;
}

if (d != 0) {
    erase_ship(x, y);
    if (x + d >= 0 && (x + SHIP_WIDTH - 1) + d < SCREEN_WIDTH)
        x += d;
    draw_ship(x, y);
}

Sleep(100);

} while (ch != 'x');

return 0;
}
```