01076104 Programming Project
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การทดลองที่ 5 : พื้นฐานการเขียน Text-mode Game ครั้งที่ 1

จุดประสงค์

นักศึกษาสามารถพัฒนา Game ในส่วนของการควบคุม และการแสดงผลต่างๆ ขั้นพื้นฐานได้

ข้อกำหนด

โปรแกรมมีการพัฒนาและทดสอบบน Visual C++ 2019 Edition การปฏิบัติการโดยใช้ Compiler ตัวอื่นๆ จะต้องมีการแก้ไขโปรแกรม

ตอนที่ 1 การวาดรูปยานบนจอภาพ

1. การสร้าง function เพื่อวาดรูป

ในการวาดรูปวัตถุต่างๆ ใน Text-mode Game มักจะมีการสร้างฟังก์ชั่นเฉพาะเพื่อวาดรูปวัตถุนั้นๆ แล้วมีการเรียกใช้งานฟังก์ชั่นนั้น ใน main() ยกตัวอย่างใน Example 1

```
#include<stdio.h>

void draw_ship()
{
     printf("<-0-> ");
}

int main()
{
     draw_ship();
     return 0;
}
```

Example 1

ในการทำงานของโปรแกรม จะเริ่มต้นที่ฟังก์ชั่น main() โดยภายในฟังก์ชั่น main() จะทำงานเพียงการเรียกฟังก์ชั่น draw_ship() ซึ่งฟังก์ชั่น draw_ship() ที่เรียกใช้งานจะทำงานเพียงการแสดงผลรูปยาน <-0-> เท่านั้น

2. การวาดรูปแบบกำหนดตำแหน่ง

การวาดรูปแบบกำหนดตำแหน่งสามารถทำได้โดยใช้คำสั่งในการเลื่อน curser บนจอภาพไปยังตำแหน่งที่ต้องการสำหรับคำสั่งที่ใช้งานคือคำสั่ง gotoxy()

```
Declaration :void gotoxy(int x, int y);
 where (x, y)is the position where we want to place the
cursor.
```

Example 2

สำหรับคำสั่ง gotoxy() เป็นคำสั่งที่ใช้ใน compiler มาตรฐานในอดีตเช่น turbo c compiler แต่ไม่มีใน Visual C++ 2019 Edition หรือ Dev-C++ เนื่องจากในระบบปฏิบัติการตระกูล Windows นั้นหน้าจอ Command Line ไม่ใช่จอภาพจริง แต่เป็นการสร้างจอภาพเสมือนขึ้นภายในระบบ Windows หากใช้ Visual C++ 2019 Edition จะต้องใช้ฟังก์ชั่นชื่อ SetConsoleCursorPosition() แทน gotoxy() ซึ่งเป็นฟังก์ชั่นที่ทำงานเหมือนกันแต่รูปแบบการเขียนยุ่งยากกว่า ดังนั้นเพื่อให้เขียนโปรแกรมทำได้ง่ายขึ้น จึงทำการสร้างฟังก์ชั่น gotoxy() ขึ้นใหม่จาก SetConsoleCursorPosition() ดังนี้

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

Example 3

*เมื่อมีการใช้งาน SetConsoleCursorPosition() ต้อง #include<windows.h>

โดยเราสามารถวาดรูปยาน <-0-> ที่บรรทัดที่ 5 ตัวอักษรที่ 10 โดยเรียกฟังก์ชั่น gotoxy(10,5) เพื่อย้าย cursor ที่เป็นตัวกำหนดตำแหน่งการแสดงผล ไปที่ตำแหน่ง (10,5) บนจอภาพก่อน แล้วจึงเรียกฟังก์ชั่น draw_ship() ให้วาดรูปยาน

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

void gotoxy(int x, int y)
{    … }

void draw_ship()
{    … }
```

```
int main()
{
     gotoxy(10,5);
     draw_ship();
     return 0;
}
```

Example 4

* รายละเอียดในฟังก์ชั่น gotoxy() และ draw_ship() คือโค้ดในกรอบ Example ก่อนหน้า

3. คำสั่ง Sleep() เป็นคำสั่งที่ใช้ชลอการทำงานของโปรแกรมจะใช้ในการชลอไม่ให้โปรแกรมทำงานรวดเร็วเกินไปจนไม่สามารถเล่นเกมได้

```
Declaration :bool Sleep(Delay)
Delay -The time interval for which execution is to
be suspended, in milliseconds.
```

Example 4

*เมื่อมีการใช้งาน Sleep() ต้อง #include<windows.h>

เช่น

```
for(x=1;x<10;x++)
{
printf("x");
Sleep(1000);
}
```

Example 5

โปรแกรมนี้จะมีการค่อยๆ แสดงผลตัว x โดยจะแสดงผลทุกๆ 1 วินาที (1000 มิลลิวินาที) แต่หากไม่มีคำสั่ง Sleep จะมีการแสดงผลอย่างรวดเร็ว

Assignment 1 : คำสั่ง draw_ship() จะทำงานเพียงวาดรูปยานที่จุด curser ปัจจุบันเท่านั้น ให้นักศึกษาแก้ไขฟังก์ชั่น draw_ship() ให้เป็น draw_ship (int x,int y) ซึ่งจะแสดงผลยานที่บรรทัดที่ y ตัวอักษรที่ x แล้วทดลองใช้งาน

Assignment 2 : ให้นักศึกษาลองแก้โปรแกรมใน Example 4 ให้ยานเคลื่อนที่ในบรรทัดที่ 20 จากทางซ้ายของจอภาพไปยังด้านขวาของจอภาพ (กำหนดให้จอภาพมี 80 ตัวอักษรต่อบรรทัด) โดยยานจะเคลื่อนที่อย่างช้าๆ ตำแหน่งละ 500 มิลลิวินาที

ตอนที่ 2 การควบคุมให้ยานเคลื่อนที่

4. ให้นักศึกษาเพิ่ม #include<conio.h> เพื่อให้ compiler รู้จักคำสั่ง _getch() แล้วแก้ไขฟังก์ชั่น main() ตาม Example 6 เพื่อสร้างลูปการทำงานหลักของเกม

```
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=='d'){draw_ship(++x,y);}
              fflush(stdin);
      }
     Sleep(100);
     } while (ch!='x');
     return 0;
}
```

Example 6

ในการทำงานของ main()จะมี do-while loop เป็น loop หลักในการทำงาน ภายใน loop จะมีการทำงานดังนี้

1. มีการตรวจสอบว่ามีการกดคีย์บอร์ดหรือไม่ (ในการทำงานของระบบคอมพิวเตอร์ เมื่อมีการกด keyboard จะมีการเก็บข้อมูลลงใน keyboard buffer ซึ่งเป็นหน่วยความจำส่วนหนึ่งในระบบ และฟังก์ชั่น _kbhit() เป็นฟังก์ชั่นมาตรฐานของภาษา C ใช้ตรวจสอบว่ามีการกดคีย์บอร์ดหรือไม่)
2. ถ้ามีการกดคีย์บอร์ด จะตรวจสอบว่ามีการกดปุ่มใด (ch=_getch(); เป็นคำสั่งใช้ดึงตัวอักษรใน keyboard buffer มาเก็บไว้ใน ch แล้วใช้คำสั่ง if ในการตรวจสอบว่า ch เป็นตัวอักษรใด)

3. ในกรณีที่มีการกดปุ่ม ‘a’ จะมีการวาดรูปยานด้านซ้ายของตำแหน่ง (x,y) และ ในกรณีที่มีการกดปุ่ม ‘d’ จะมีการวาดรูปยานด้านขวาของตำแหน่ง (x,y) ซึ่งมีการกำหนดค่าเริ่มต้น (x,y) คือ (38,20) ในการ draw_ship(x,y); ครั้งแรกจะได้ตำแหน่งยานที่บรรทัดที่ 20 ตัวอักษรที่ 38
4. เมื่อมีการตรวจสอบการกดปุ่มโดยคำสั่ง if แล้วจะมีการล้างข้อมูลใน keyboard buffer โดยใช้คำสั่ง fflush(stdin); เพื่อให้ keyboard buffer ว่างรอรับการกดปุ่มถัดไป
5. do-while loop จะสิ้นสุดการทำงานเมื่อมีการกดปุ่ม ‘x’

Assignment 3 : เนื่องจากหน้าจอปกติจะมีจำนวนตัวอักษรต่อบรรทัดคือ 80 และโปรแกรมปัจจุบันมี bug ที่ยอมให้ยานจะเคลื่อนที่ไปนอกพื้นที่จอภาพ ให้นักศึกษาแก้ไขโปรแกรมให้ยานเคลื่อนที่ได้ในช่วงของจอภาพเท่านั้น

Assignment 4 : เนื่องจากขอบของรูปยานในฟังก์ชั่น draw_ship() จะมีช่องว่างด้านข้างปีกทั้งซ้ายและขวา ทำให้การเคลื่อนที่ไปทางซ้ายหรือขวาจะมีการลบภาพเดิมโดยอัตโนมัติ แต่ในกรณีที่มีการเลื่อนยานขึ้นหรือลง ต้องมีการลบภาพยานที่ตำแหน่งเดิม แล้วแสดงผลภาพยานที่ตำแหน่งใหม่ ให้นักศึกษาสร้าง function erase_ship(int x,int y) สำหรับลบรูปยานที่ตำแหน่ง (x,y) พร้อมทั้งแก้ไขโปรแกรมให้ยานสามารถเคลื่อนที่ได้ทั่วจอภาพ โดยใช้ปุ่ม ‘w’ สำหรับการเคลื่อนที่ขึ้น และปุ่ม ‘s’ สำหรับการเคลื่อนที่ลง โดยควบคุมให้ยานเคลื่อนที่ภายในจอภาพเท่านั้น

การส่งงาน : ให้นักศึกษาส่งโปรแกรมที่สมบูรณ์ที่ทำงานใน Assignment 1-4 เพียงโปรแกรมเดียวเท่านั้น