C:\Users\pakuton\Desktop\ce\pro_fund\lab8\lab8.cpp 1

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

#define scount 80
#define screen_x 80
#define screen_y 25

HANDLE wHnd;
HANDLE rHnd;
DWORD fdwMode;
CHAR_INFO consoleBuffer[screen_x * screen_y];
COORD bufferSize = { screen_x,screen_y };
COORD characterPos = { 0,0 };
SMALL_RECT windowSize = { 0,0,screen_x - 1,screen_y - 1 };
COORD star[scount];

int setConsole(int x, int y)
{
    wHnd = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleWindowInfo(wHnd, TRUE, &windowSize);
    SetConsoleScreenBufferSize(wHnd, bufferSize);
    return 0;
}
void setcursor(bool visible)
{
    HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_CURSOR_INFO lpCursor;
    lpCursor.bVisible = visible;
    lpCursor.dwSize = 20;
    SetConsoleCursorInfo(console, &lpCursor);
}
int setMode()
{
    rHnd = GetStdHandle(STD_INPUT_HANDLE);
    fdwMode = ENABLE_EXTENDED_FLAGS | ENABLE_WINDOW_INPUT |
        ENABLE_MOUSE_INPUT;
    SetConsoleMode(rHnd, fdwMode);
    return 0;
}
void clear_buffer()
{
    for (int y = 0; y < screen_y; ++y) {
        for (int x = 0; x < screen_x; ++x) {
            consoleBuffer[x + screen_x * y].Char.AsciiChar = ' ';
            consoleBuffer[x + screen_x * y].Attributes = 7;
        }
    }
```

```
50  }
51  void fill_buffer_to_console()
52  {
53      WriteConsoleOutputA(wHnd, consoleBuffer, bufferSize, characterPos,
          &windowSize);
54  }
55  void init_star()
56  {
57      for (int i = 0; i < scount; i++) {
58          star[i].X = rand() % 80;
59          star[i].Y = rand() % 25;
60      }
61  }
62  void star_fall()
63  {
64      int i;
65      for (i = 0; i < scount; i++) {
66          if (star[i].Y >= screen_y - 1) {
67              star[i] = { (rand() % screen_x),1 };
68          }
69          else {
70              star[i] = { star[i].X, star[i].Y + 1 };
71          }
72      }
73  }
74  void fill_star_to_buffer()
75  {
76      for (int i = 0; i < scount; i++) {
77              consoleBuffer[star[i].X + screen_x * star[i].Y].Char.AsciiChar
                  = '*';
78              consoleBuffer[star[i].X + screen_x * star[i].Y].Attributes =
                  7;
79      }
80  }
81  void draw_ship(int xmouse, int ymouse,int color)
82  {
83      if (xmouse >= 0 && xmouse <= 80 && ymouse >= 0 && ymouse <= 25) {
84          consoleBuffer[xmouse - 1 + screen_x * ymouse].Char.AsciiChar =
              '<';
85          consoleBuffer[xmouse - 1 + screen_x * ymouse].Attributes = color;
86          consoleBuffer[xmouse + screen_x * ymouse].Char.AsciiChar = '-';
87          consoleBuffer[xmouse + screen_x * ymouse].Attributes = color;
88          consoleBuffer[xmouse + 1 + screen_x * ymouse].Char.AsciiChar =
              '>';
89          consoleBuffer[xmouse + 1 + screen_x * ymouse].Attributes = color;
90          fill_buffer_to_console();
91      }
92  }
93  int main()
```

C:\Users\pakuton\Desktop\ce\pro_fund\lab8\lab8.cpp 3

```
 94  {
 95      bool play = true;
 96      DWORD numEvents = 0;
 97      DWORD numEventsRead = 0;
 98      int posx = 0, posy = 0;
 99      int* xmouse = &posx, *ymouse = &posy;
100      int ship_color = 5, crash = 10;
101
102      srand(time(NULL));
103      setConsole(screen_x, screen_y);
104      setMode();
105      clear_buffer();
106      init_star();
107      setConsole(screen_x, screen_y);
108      setcursor(0);
109      setMode();
110      while (play)
111      {
112          clear_buffer();
113          star_fall();
114          fill_star_to_buffer();
115          fill_buffer_to_console();
116
117          GetNumberOfConsoleInputEvents(rHnd, &numEvents);
118          if (numEvents != 0) {
119              INPUT_RECORD* eventBuffer = new INPUT_RECORD[numEvents];
120              ReadConsoleInput(rHnd, eventBuffer, numEvents,
                  &numEventsRead);
121              for (DWORD i = 0; i < numEventsRead; ++i) {
122                  if (eventBuffer[i].EventType == KEY_EVENT &&
123                      eventBuffer[i].Event.KeyEvent.bKeyDown == true) {
124                      if (eventBuffer[i].Event.KeyEvent.wVirtualKeyCode ==
                        VK_ESCAPE) {
125                          play = false;
126                      }
127                      if (eventBuffer[i].Event.KeyEvent.uChar.AsciiChar ==
                        'c') {
128                          ship_color = rand() % 15 + 1;
129                      }
130                  }
131                  else if (eventBuffer[i].EventType == MOUSE_EVENT) {
132                      posx = eventBuffer
                        [i].Event.MouseEvent.dwMousePosition.X;
133                      posy = eventBuffer
                        [i].Event.MouseEvent.dwMousePosition.Y;
134                      xmouse = &posx;
135                      ymouse = &posy;
136                      if (eventBuffer[i].Event.MouseEvent.dwButtonState &
                        FROM_LEFT_1ST_BUTTON_PRESSED) {
```

```
137                         ship_color = rand() % 15 + 1;
138                     }
139                     else if (eventBuffer[i].Event.MouseEvent.dwButtonState
                        & RIGHTMOST_BUTTON_PRESSED) {
140                         printf("right click\n");
141                     }
142                     else if (eventBuffer[i].Event.MouseEvent.dwEventFlags
                        & MOUSE_MOVED) {
143                         draw_ship(posx, posy, ship_color);
144                     }
145                 }
146             }
147             delete[] eventBuffer;
148         }
149         else {
150             draw_ship(*xmouse, *ymouse, ship_color);
151         }
152         for (int i = 0; i < scount; i++) {
153             if (*xmouse == star[i].X && *ymouse == star[i].Y ||
154                 *xmouse == star[i].X - 1 && *ymouse == star[i].Y ||
155                 *xmouse == star[i].X + 1 && *ymouse == star[i].Y) {
156                 star[i].X = rand() % 80;
157                 star[i].Y = 0;
158                 crash--;
159             }
160         }
161         if (crash <= 0) play = false;
162         Sleep(100);
163     }
164     return 0;
165 }
```