

게임프로그래밍 발표

2021558007 김한준

Contents

01 소스 분석

02 추가 기능 &

02 실행 화면

01 소스 분석

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.h>
4 #include <windows.h>
5
6 #define box_length 15 //게 임 의 영 역 (좌 우 길 이)
7 #define box_height 15 //바 닥 의 높 이 (상 하 길 이)
```

헤더파일

```
9 void intro_game(void);
10 void game_control(void);
11 void gotoxy(int x, int y);
12 int left_right_move(void);
13 void move_down(int x);
14 void draw_rectangle(int c, int r);
15 int max_block(void);
16
17 int block_stack[box_length*2+1]={0};
```

함수, 전역 변수

02 소스 분석

```
19 int main(void)
20 {
21     intro_game();
22     game_control();
23     gotoxy(1, box_height+3);
24     printf("game이 종료되었습니다. \n");
25     return 0;
26 }
```

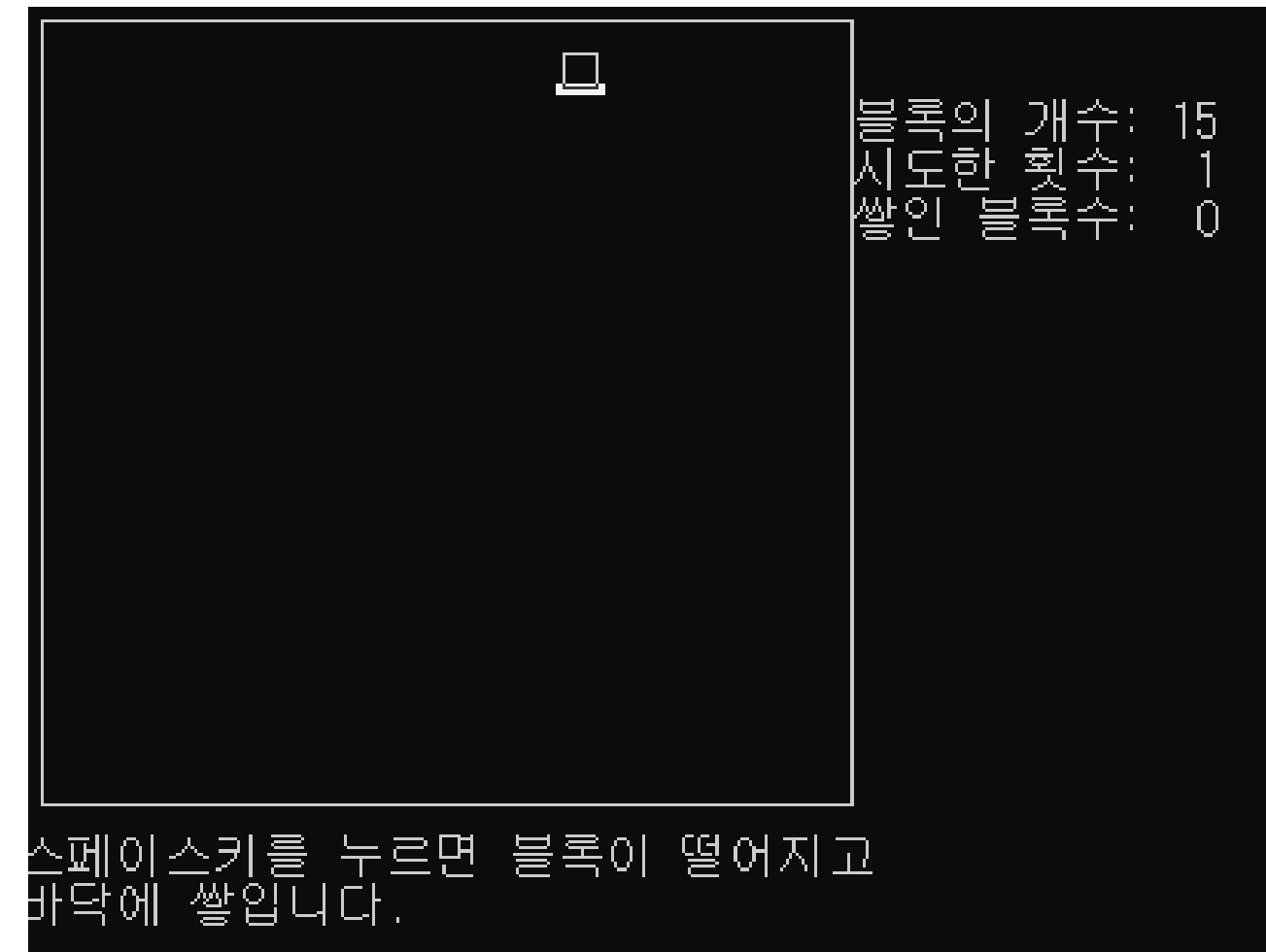
main() 함수

```
28 void intro_game(void)
29 {
30     system("cls");
31     printf("블록 쌓기 \n\n");
32     printf("블록이 좌우로 움직일 때 스페이스키를 누르면 \n");
33     printf("블록이 떨어져 바닥에 쌓입니다. \n\n");
34     printf("아무키나 누르면 게임을 시작합니다. \n");
35     getch();
36 }
```

기본적인 게임 시작 화면

03 소스 분석

```
37 void game_control(void)
38 {
39     int x, count=0;
40     system("cls");
41     draw_rectangle(box_length, box_height);
42     gotoxy(box_length*2+5,3);
43     printf("블록의 개수 : %2d", box_height);
44     gotoxy(1, box_height+3);
45     printf("스페이스키를 누르면 블록이 떨어지고 \n");
46     printf("바닥에 쌓입니다. \n");
47     while(count<box_height)
48     {
49         gotoxy(box_length*2+5,4);
50         printf("시도한 횟수 : %2d", count+1);
51         gotoxy(box_length*2+5,5);
52         printf("쌓인 블록수 : %2d", max_block());
53         x=left_right_move();
54         move_down(x);
55         count++;
56         getch();
57     }
58 }
59 }
```



04 소스 분석

```
void gotoxy(int x, int y)
{
    COORD Pos = {x - 1, y - 1};
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
}
```

gotoxy 함수

```
65 int left_right_move(void)
66 {
67     int x=3, y=2, temp=2;
68     do
69     {
70         x+=temp;
71         if (x>(box_length*2)) //x방향 최대 값 설정
72             temp=-2;
73         if (x<3)
74         {
75             x=3;
76             temp=2;
77         }
78
79         gotoxy(x, y);
80         printf("□ ");
81         Sleep(100); //블록이 좌우로 움직이는 속도를 조절
82         gotoxy(x, y);
83         printf(" ");
84
85     }while(!kbhit());
86     block_stack[x]+=1;
87     return x;
88 }
```

블럭의 좌표 값, 속도 제한

키 입력시 block_stack[x] += 1

05 소스 분석

```
89 void move_down(int x)
90 {
91     int y;
92     for(y=2; y<box_height+2-block_stack[x]; y+=1)
93     {
94         gotoxy(x, y);
95         printf("□ ");
96         Sleep(20);
97         gotoxy(x, y);
98         printf("  ");
99         Sleep(10);
100    }
101    gotoxy(x, box_height+2-block_stack[x]);
102    printf("□ ");
103 }
```

블럭이 내려가는 모습 출력

06 소스 분석

```
104 void draw_rectangle(int c, int r)
105 {
106     int i, j;
107     unsigned char a=0xa6;
108     unsigned char b[7];
109     for(i=1;i<7;i++)
110         b[i]=0xa0+i;
111
112     printf("%c%c",a, b[3]);
113     for(i=0;i<c*2+2;i++)
114         printf("%c%c", a, b[1]);
115     printf("%c%c", a, b[4]);
116     printf("\n");
117     for(i=0;i<r;i++)
118     {
119         printf("%c%c", a, b[2]);
120         for(j=0;j<c+1;j++)
121             printf(" ");
122         printf("%c%c",a, b[2]);
123         printf("\n");
124     }
125     printf("%c%c", a, b[6]);
126     for(i=0;i<c*2+2;i++)
127         printf("%c%c", a, b[1]);
128     printf("%c%c", a, b[5]);
129     printf("\n");
130 }
```

게임판 생성

```
131 int max_block(void)
132 {
133     int i, max=0;
134     for(i=1;i<box_height*2+1;i++)
135     {
136         if (max<=block_stack[i])
137             max=block_stack[i];
138     }
139     return max;
140 }
```

가장 많이 쌓인 블록의 수

01 추가 기능

블록 쌓기

블록이 좌우로 움직일때 스페이스키를 누르면
블록이 떨어져 바닥에 쌓입니다.

20점에 도달하면 게임이 종료됩니다.

아무 키를 누르면 게임을 시작합니다.

```
46 void intro_game(void)
47 {
48     system("cls");
49     printf("블록 쌓기 \n\n");
50     printf("블록이 좌우로 움직일때 스페이스키를 누르면 \n");
51     printf("블록이 떨어져 바닥에 쌓입니다. \n\n");
52     printf("20점에 도달하면 게임이 종료됩니다. \n\n");
53     printf("아무 키를 누르면 게임을 시작합니다. \n");
54     printf("\n");
55     getch();
56 }
```

01 추가 기능

```
int left_right_move(int star)
{
    Sleep(100-(point_block())*5);
```

1. 점수가 높아질수록 속도가 증가

```
int main(void)
{
    srand((unsigned)time(NULL));
    char c;
    do
    {
        intro_game();
        game_control();
        gotoxy(1, box_height+3);
        printf("game이 종료 되었습니다 !! \n");
        printf("다시 시도 하시겠습니까? (Y)");
    }while((c=getch()) == 121 || (c=getch()) == 89);
    return 0;
}
```

2. 재시작

추가 기능



추가 기능

3. 남은 블록으로 변경

```
남은 블록      |      남은 블록
■ : 20개        |      □ : 20개
```

```
94 | ..... printf("남은 블록      |      남은 블록");
95 | ..... gotoxy(box_length*2+10,10);
96 | ..... printf("■ : %2d개      |      □ : %2d개 ", box_height-p1, box_height-p2);
```

```
58 void game_control(void)
59 {
60     int p1=0,p2=0;
```

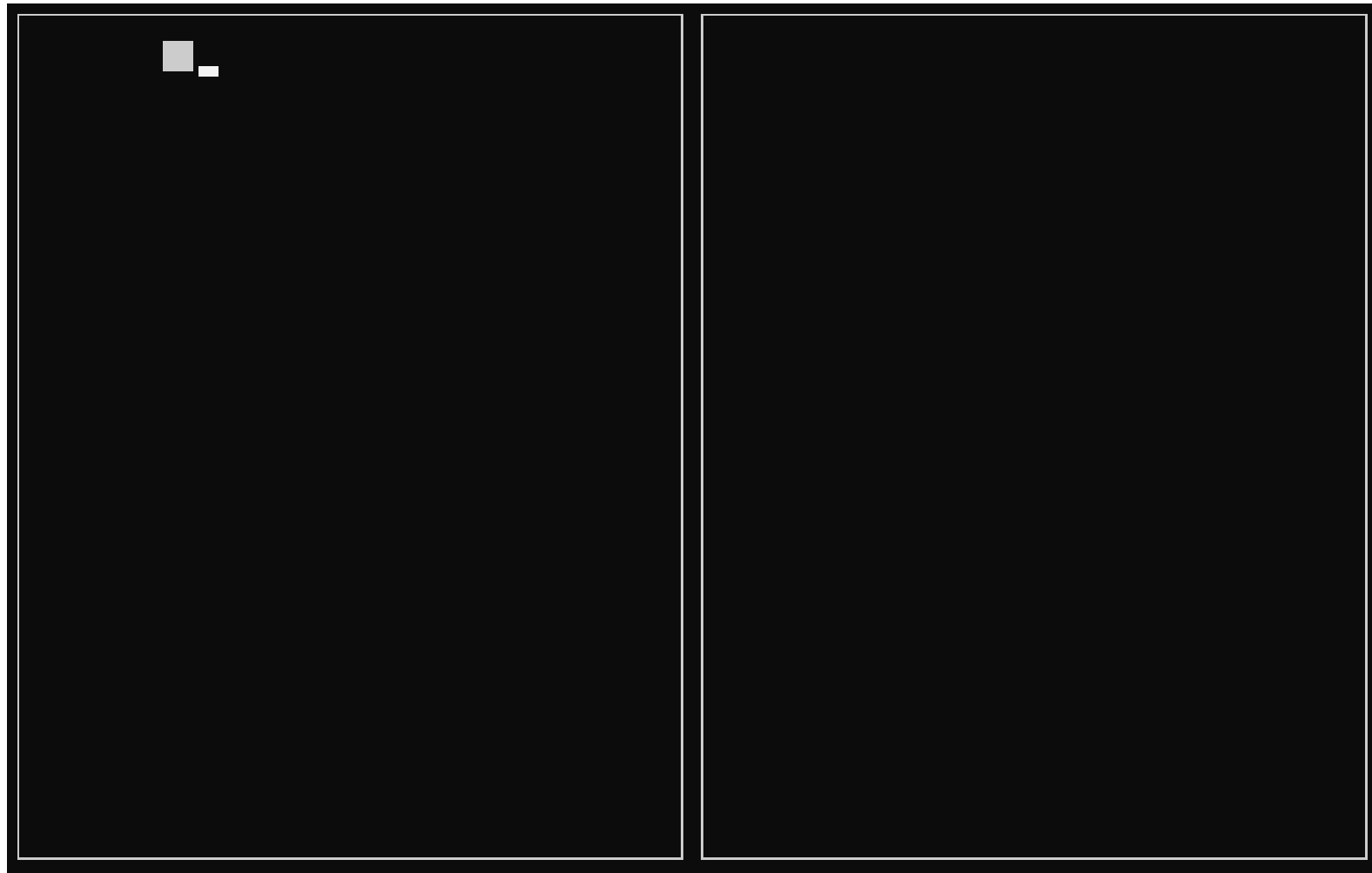
4. 20점을 넘을 시 강제 종료

```
플레이어1      |      플레이어2
■ : 0점         |      □ : 0점
```

```
if(point_block()>=20 || point_block2()>=20)
{
    ..... break;
}
```

추가 기능

5. 2명에서 같이 플레이 할 수 있게 (개인 턴이 존재함)



```
335 // 사각형 그리기
336 void draw_rectangle(int c, int r)
337 {
338     int i, j;
339     unsigned char a=0xa6;
340     unsigned char b[7];
341     for(i=1;i<7;i++)
342         b[i]=0xa0+i;
343
344     printf("%c%c",a, b[3]);
345     for(i=0;i<c*2+2;i++)
346         printf("%c%c", a, b[1]);
347     printf("%c%c", a, b[4]);
348     printf("%c%c",a, b[3]);
349     for(i=0;i<c*2+2;i++)
350         printf("%c%c", a, b[1]);
351     printf("%c%c", a, b[4]);
352     printf("\n");
353     for(i=0;i<r;i++)
354     {
355         printf("%c%c", a, b[2]);
356         for(j=0;j<c+1;j++)
357             printf(" ");
358         printf("%c%c",a, b[2]);
359         printf("%c%c", a, b[2]);
360         for(j=0;j<c+1;j++)
361             printf(" ");
362         printf("%c%c",a, b[2]);
363         printf("\n");
364     }
365     printf("%c%c", a, b[6]);
366     for(i=0;i<c*2+2;i++)
367         printf("%c%c", a, b[1]);
368     printf("%c%c", a, b[5]);
369     printf("%c%c", a, b[6]);
370     for(i=0;i<c*2+2;i++)
371         printf("%c%c", a, b[1]);
372     printf("%c%c", a, b[5]);
373     printf("\n");
374 }
```

추가 기능

2명이서 같이 플레이 할 수 있게 (개인 턴이 존재함)



```
13  int left_right_move(int star);
14  int left_right_move2(int star);
15  void move_down(int star, int x);
16  void move_down2(int star, int x1);
```

```
23  int block_stack[box_length*2+3]={0};
24  int block_stack2[box_length*2+3]={0};
```

```
void game_control(void)
{
    int number = count % 2;
    if(number==0)
    {
        x=left_right_move(star);
        p1++;
    }
    else
    {
        x1=left_right_move2(star);
        p2++;
    }
}
```

추가 기능

2명이서 같이 플레이 할 수 있게 (개인 턴이 존재함)



```
141 int left_right_move(int star)
142 {
143     int x=3, y=2, temp=2;
144     int num;
145     do
146     {
147         x+=temp;
148         if (x>(box_length)) //x방향 최대값 설정
149             temp=-2;
150         if (x<3)
151         {
152             x=3;
153             temp=2;
154         }
155
156         gotoxy(x, y);
157         switch(star)
158         {
159             case 0:
160                 printf("★ ");
161                 break;
162             case 1:
163                 printf("○ ");
164                 break;
165             default:
166                 printf("■ ");
167                 break;
168         }
169         Sleep(100-(point_block()*5)); //블록이 파워트 움직이는 속도를 조절
170         gotoxy(x, y);
171         printf(" ");
172     }
```

추가 기능

2명이서 같이 플레이 할 수 있게 (개인 턴이 존재함)



```
248 void move_down(int star, int x)
249 {
250     int y;
251     for(y=2;y<box_height+2-block_stack[x];y+=1)
252     {
253         gotoxy(x, y);
254         switch(star)
255         {
256             case 0:
257                 printf("★ ");
258                 break;
259             case 1:
260                 printf("○ ");
261                 break;
262             default:
263                 printf("■ ");
264                 break;
265         }
266         Sleep(20);
267         gotoxy(x, y);
268         printf(" ");
269         Sleep(10);
270     }
271     switch(star)
272     {
273         case 0:
274             gotoxy(x, box_height+2-block_stack[x]);
275             printf("★ ");
276             break;
277         case 1:
278             gotoxy(x, box_height+2-block_stack[x]);
279             printf("○ ");
280             break;
281         default:
282             gotoxy(x, box_height+2-block_stack[x]);
283             printf("■ ");
284             break;
285     }
286 }
```


추가 기능

6. 2명의 플레이어가 각각의 점수와 블록을 가짐

```
플레이어1   |   플레이어2
■ : 0점     |   □ : 0점
남은 블록   |   남은 블록
■ : 20개    |   □ : 20개
```

```
gotoxy(box_length*2+10,6);
printf("플레이어1   |   플레이어2");
gotoxy(box_length*2+10,7);
printf("■ : %2d점   |   □ : %2d점 ", point_block(), point_block2());
```

```
375 int point_block(void)
376 {
377     int i, point=0;
378     for(i=1;i<box_height*2+1;i++)
379     {
380         if (point<=block_stack[i]+star_stack[i]-minus_stack[i])
381             point=block_stack[i]+star_stack[i]-minus_stack[i];
382     }
383     return point;
384 }
385
386 int point_block2(void)
387 {
388     int i, point=0;
389     for(i=1;i<box_height*3+8;i++)
390     {
391         if (point<=block_stack2[i]+star_stack2[i]-minus_stack2[i])
392             point=block_stack2[i]+star_stack2[i]-minus_stack2[i];
393     }
394     return point;
395 }
396
```

추가 기능

7. 1등 플레이어 표시

현재 1등: 플레이어 1 (2 점)

플레이어1		플레이어2
■ : 2점		□ : 0점

남은 블록		남은 블록
■ : 19개		□ : 20개

```
gotoxy(box_length*2+10,6);
printf("플레이어 1 | 플레이어 2");
gotoxy(box_length*2+10,7);
printf("■ : %2d점 | □ : %2d점 ", point_block(), point_block2());
```

추가 기능

8. 이벤트 블록

○
■
■
■ ☆

스페이스키를 누르면 블록이 떨어지고 바닥에 쌓입니다.

☆

□□
□□

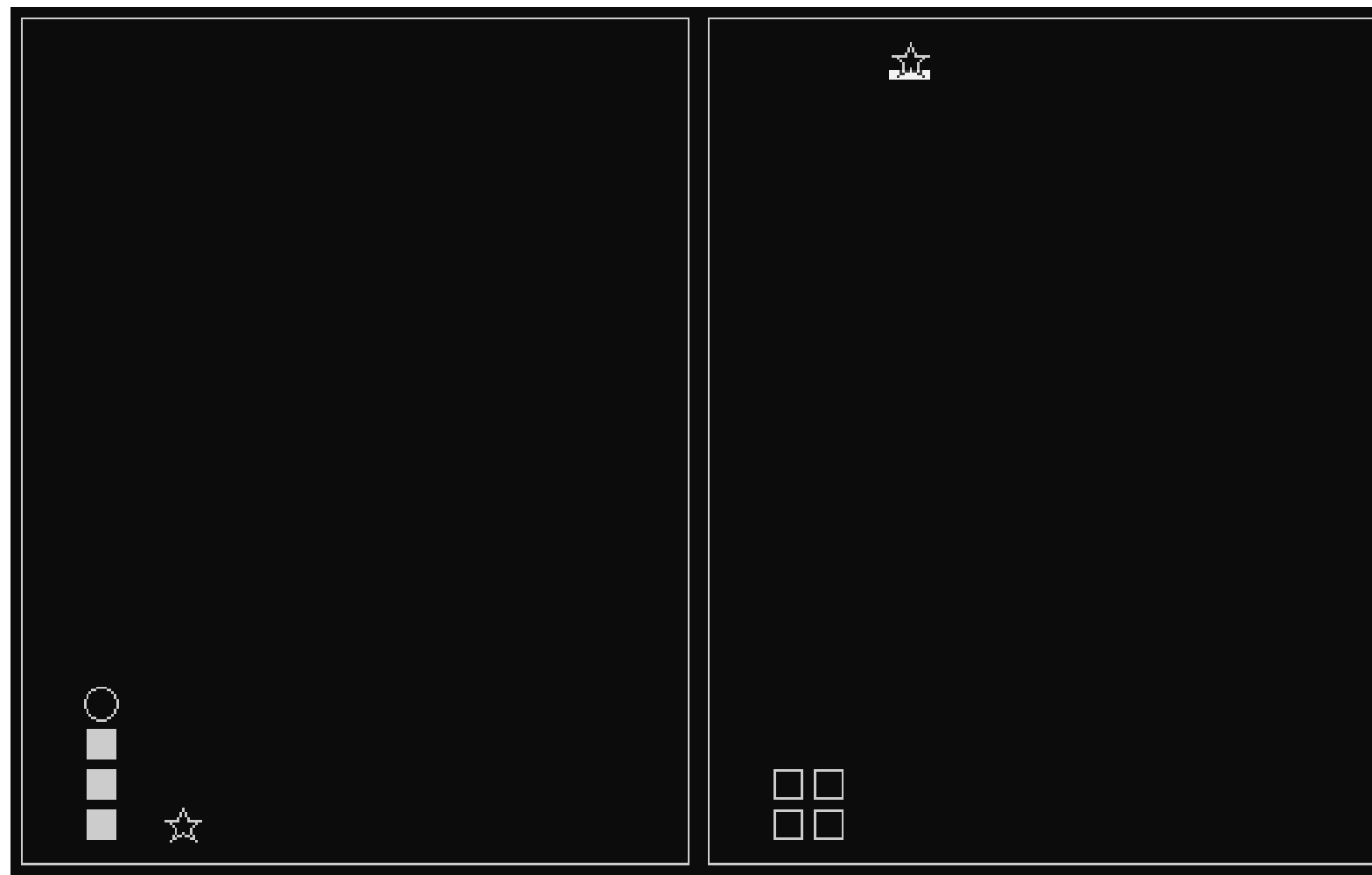
현재 1등: 플레이어 1 (3 점)

플레이어1		플레이어2
■: 2점		□: 2점
남은 블록		남은 블록
■: 15개		□: 16개

■: 1점 | □: 1점 | ☆: 2점 | ○: -1점

추가 기능

이벤트 블록



■: 1점 | □: 1점 | ☆: 2점 | ○: -1점

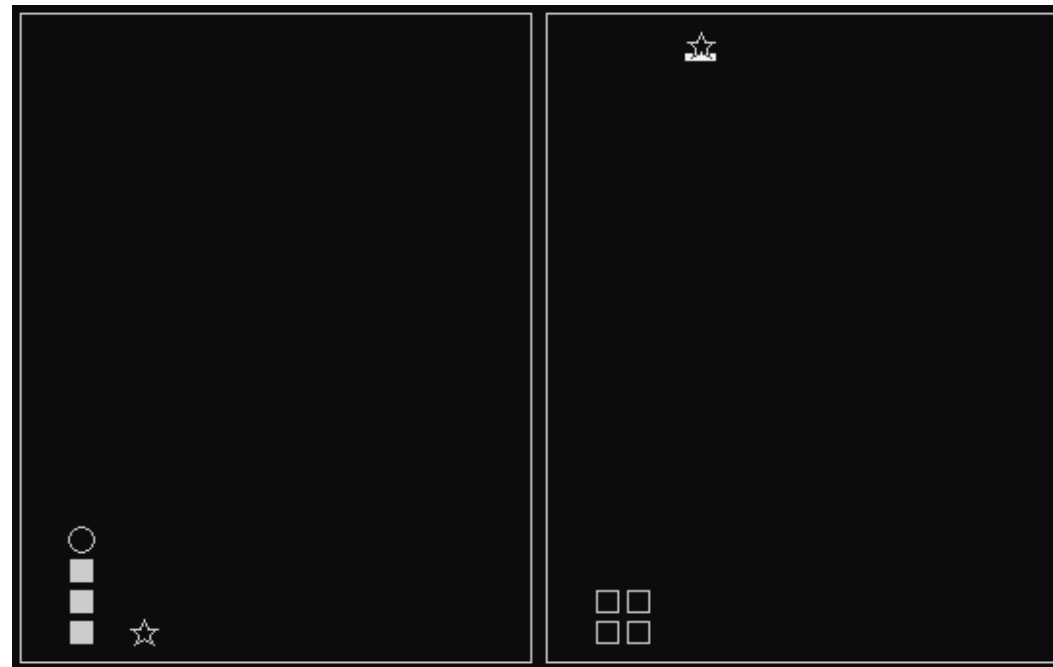
```
5 #include <time.h>
```

```
int main(void)
{
    srand((unsigned)time(NULL)); 난수 생성
    int star = rand() % 9;

    if(number==0)
    {
        x=left_right_move(star);
        p1++;
    }
    else
    {
        x1=left_right_move2(star);
        p2++;
    }
    if(number==0)
        move_down(star, x);
    else
        move_down2(star, x1);
```

추가 기능

이벤트 블록



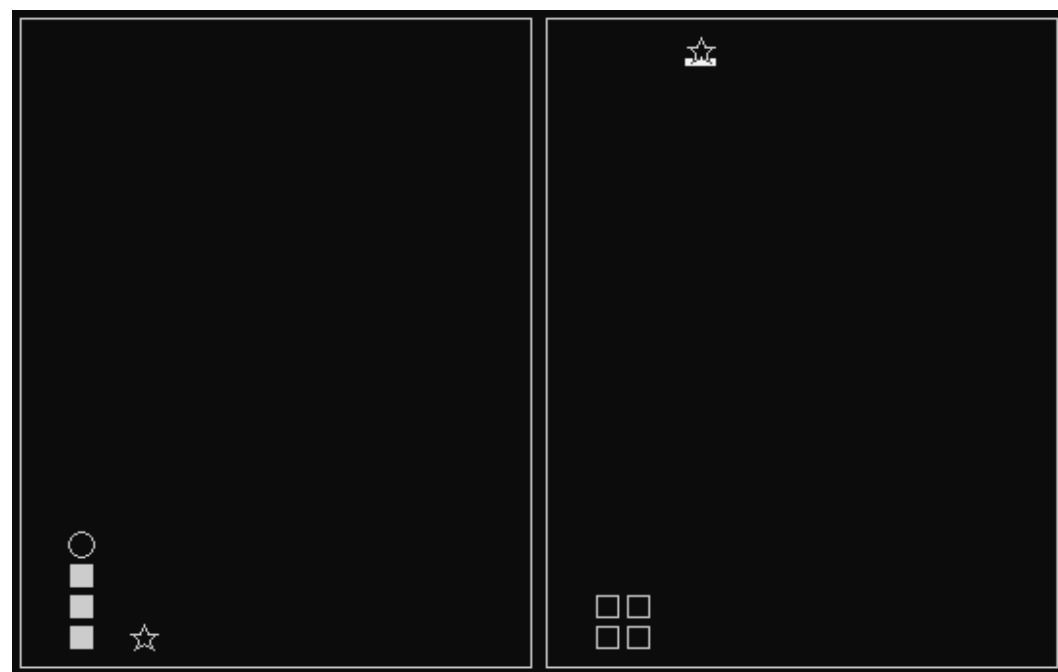
■: 1점 | □: 1점 | ☆: 2점 | ○: -1점

```
139 int left_right_move(int star)
140 {
141     int x=3, y=2, temp=2;
142     int num;
143     do
144     {
145         x+=temp;
146         if (x>(box_length)) //x발 향 최
147             temp=-2;
148         if (x<3)
149         {
150             x=3;
151             temp=2;
152         }
153
154         gotoxy(x, y);
155         switch(star)
156         {
157             case 0:
158                 printf("★ ");
159                 break;
160             case 1:
161                 printf("○ ");
162                 break;
163             default:
164                 printf("■ ");
165                 break;
166         }
```

```
167         Sleep(100-(point_block()*5));
168         gotoxy(x, y);
169         printf(" ");
170
171     }while(!kbhit());
172     switch(star)
173     {
174         case 0:
175             block_stack[x]+=1;
176             star_stack[x]+=1;
177             break;
178         case 1:
179             block_stack[x]+=1;
180             minus_stack[x]+=2;
181             break;
182         default:
183             block_stack[x]+=1;
184             break;
185     }
186     return x;
```

추가 기능

이벤트 블록

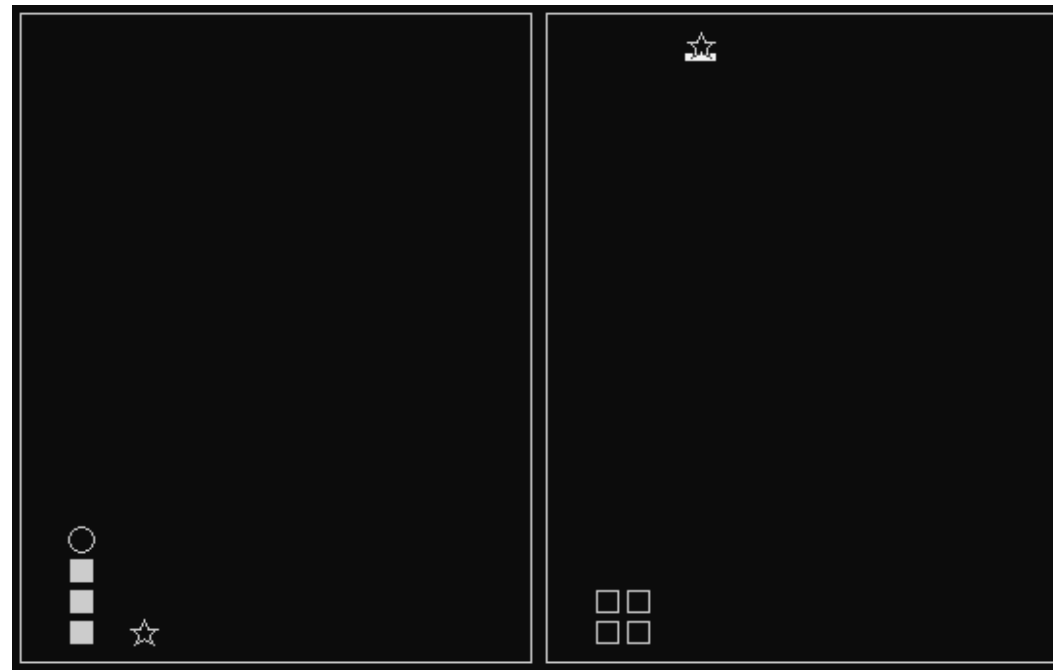


■: 1점 | □: 1점 | ☆: 2점 | ○: -1점

```
246 void move_down(int star, int x)
247 {
248     int y;
249     for(y=2;y<box_height+2-block_stack[x];y+=1)
250     {
251         gotoxy(x, y);
252         switch(star)
253         {
254             case 0:
255                 printf("★ ");
256                 break;
257             case 1:
258                 printf("○ ");
259                 break;
260             default:
261                 printf("■ ");
262                 break;
263         }
264         Sleep(20);
265         gotoxy(x, y);
266         printf(" ");
267         Sleep(10);
268     }
269     switch(star)
270     {
271         case 0:
272             gotoxy(x, box_height+2-block_stack[x]);
273             printf("★ ");
274             break;
275         case 1:
276             gotoxy(x, box_height+2-block_stack[x]);
277             printf("○ ");
278             break;
279         default:
280             gotoxy(x, box_height+2-block_stack[x]);
281             printf("■ ");
282             break;
283     }
284 }
```

추가 기능

이벤트 블록



■: 1점 | □: 1점 | ☆: 2점 | ○: -1점

```
375 int point_block(void)
376 {
377     int i, point=0;
378     for(i=1;i<box_height*2+1;i++)
379     {
380         if (point<=block_stack[i]+star_stack[i]-minus_stack[i])
381             point=block_stack[i]+star_stack[i]-minus_stack[i];
382     }
383     return point;
384 }
```

References

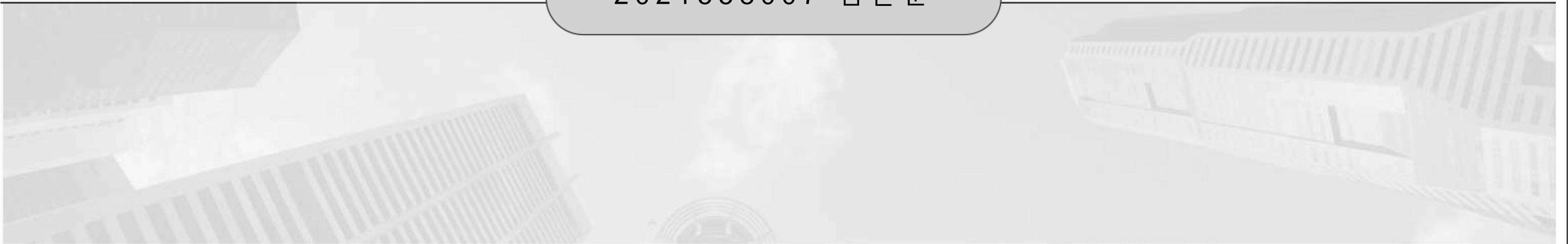


<https://blockdmask.tistory.com/308>

<https://blog.naver.com/sharonichoya/220475095313>

<https://coding-factory.tistory.com/665>

2021558007 김한준



_ □ ×

감사합니다.

2021558007 김한준

