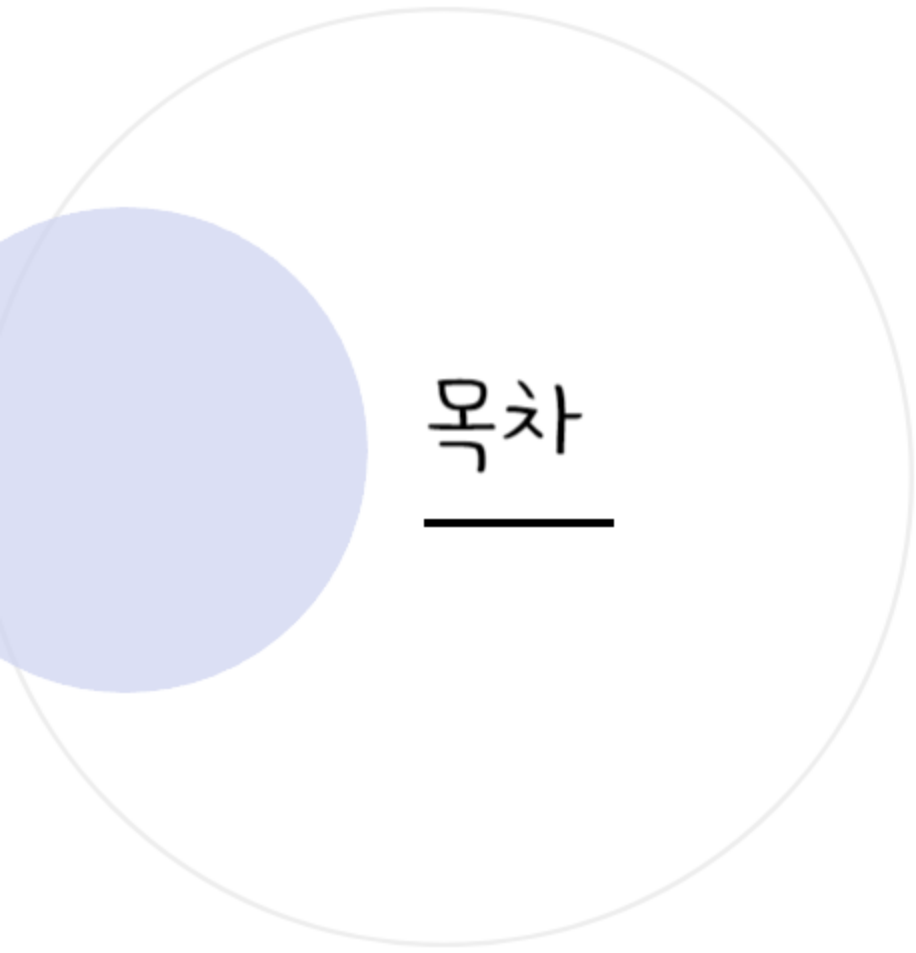


게임프로그래밍

2020904053 윤금비
8_7_1



목차

- 1 게임 설명
- 2 코드 설명
- 3 실행 화면

1. 게임 설명

- 이 코드는 콘솔에서 실행되며, 사용자가 화살표 키를 사용하여 플레이어를 이동시켜 보물을 찾는 간단한 게임을 구현한 것입니다. 게임은 보물을 모두 찾으면 종료되며, 경과 시간과 찾은 보물의 개수가 표시됩니다.

2 . 코드 설명

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <conio.h>
4  #include <windows.h>
5  #include <time.h>
6
7  void display_rule(void);
8  void gotoxy(int x, int y);
9  void make_treasure(int tx[], int ty[]);
10 void display_map(int matrix[][10], int tx[], int ty[]);
11 void basic_map(void);
12 void move_arrow_key(char key, int *x1, int *y1, int x_b, int y_b);
13 void game_control(int tx[], int ty[]);
14
15 int main(void)
16 {
17     int tx[2], ty[2];
18     int matrix[10][10]={0};
19     clock_t start, end;
20     double pst;
21     srand(time(NULL));
22     make_treasure(tx, ty);
23     display_rule();
24     start=clock();
25     system("cls");
26     game_control(tx, ty);
27     end=clock();
28     pst=(double)(end-start)/CLK_TCK;
29     gotoxy(1, 13);
30     printf("경과 시간 : %.1f 초", pst);
31     gotoxy(1, 14);
32     printf("모두 찾았습니다. 게임을 종료합니다. \n");
33     getch();
34     return 0;
35 }
```

2 . 코드 설명

```
37 void display_rule(void)
38 {
39     gotoxy(1, 1);
40     printf("보물 찾기 ");
41     gotoxy(1,3);
42     printf("아래의 바둑판 모양의 격자에는 보물이 \n");
43     gotoxy(1,4);
44     printf("숨겨져 있습니다. \n");
45     gotoxy(1,5);
46     printf("화살표 (↑ ↓ ← →)키를 움직여서 찾습니다. \n");
47     gotoxy(1,7);
48     basic_map();
49     printf("\n아무키나 누르면 시작합니다. \n");
50     getch();
51 }
52
```

```
52
53 void basic_map(void)
54 {
55     int i, j;
56     for(i=0;i<10;i++)
57     {
58         for(j=0;j<10;j++)
59             printf("■ ");
60         printf("\n");
61     }
62 }
63
```

2 . 코드 설명

```
65 void make_treasure(int tx[], int ty[])
66 {
67     int i;
68     for(i=0;i<2;i++)
69     {
70         do
71         {
72             tx[i]=rand()%11; //보물의 x좌표
73             ty[i]=rand()%11; //보물의 y좌표
74             }while((tx[i]<=1) || (ty[i]<=1));
75         }
76     }
77
80 void gotoxy(int x, int y)
81 {
82     COORD Pos = {x - 1, y - 1};
83     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), Pos);
84 }
```

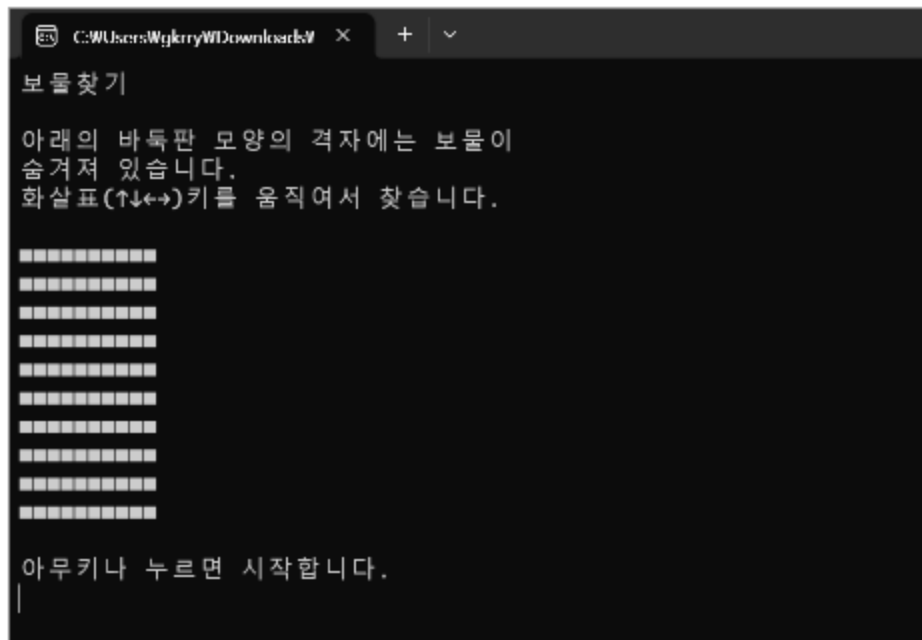
```
86 void display_map(int matrix[][11], int tx[], int ty[])
87 {
88     int i, j;
89     basic_map();
90     for(i=1;i<=10;i++)
91         for(j=1;j<=10;j++)
92             if (matrix[i][j]==1)
93             {
94                 gotoxy(2*i-1, j);
95                 printf("□ ");
96             }
97             else if (matrix[i][j]==2)
98             {
99                 gotoxy(2*i-1, j);
100                 printf("★ ");
101             }
102 }
```

2 . 코드 설명

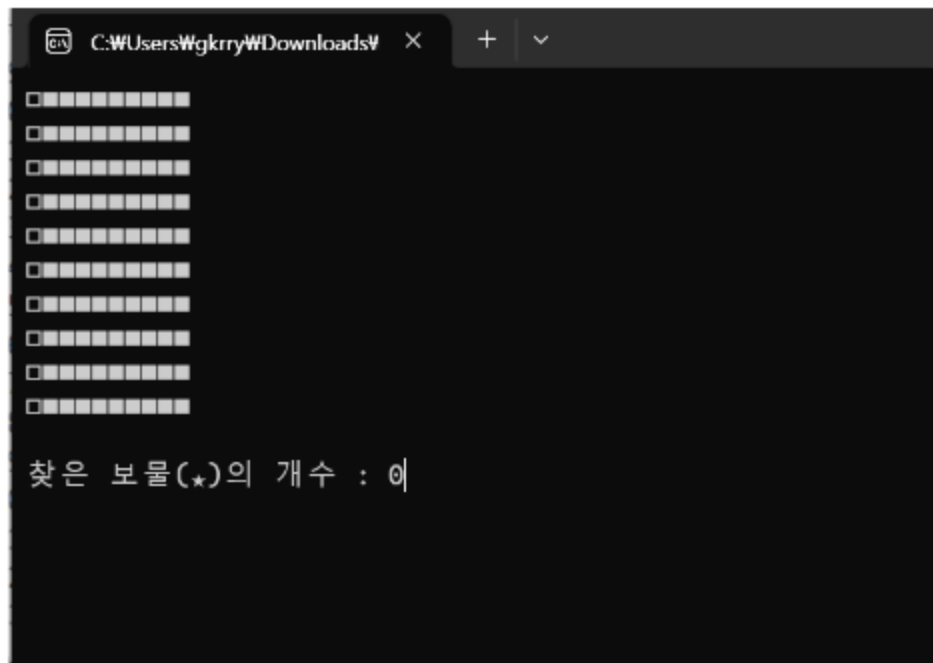
```
103 void move_arrow_key(char key, int *x1, int *y1, int x_b, int y_b)
104 {
105     switch(key)
106     {
107         case 72: //위쪽(상) 방향의 화살표 키 입력
108             *y1=*y1-1;
109             if (*y1<1) *y1=1; //y좌표의 최소값
110             break;
111         case 75: //왼쪽(좌) 방향의 화살표 키 입력
112             *x1=*x1-2; //x좌표의 감소값
113             if (*x1<1) *x1=1; //x좌표의 최소값,
114             break;
115         case 77: //오른쪽(우) 방향의 화살표 키 입력
116             *x1=*x1+2; //x좌표의 증가값
117             if (*x1>x_b) *x1=x_b; //x좌표의 최대값
118             break;
119         case 80: //아래쪽(하) 방향의 화살표 키 입력
120             *y1=*y1+1;
121             if (*y1>y_b) *y1=y_b; //y좌표의 최대값;
122             break;
123         default:
124             return;
125     }
```

2 . 코드 설명

```
128 void game_control(int tx[], int ty[])
129 {
130     char key;
131     int i, j, count=0;
132     int x=1, y=1;
133     int matrix[11][11]={0};
134     do
135     {
136         gotoxy(x, y);
137         printf("□ ");
138         matrix[(x+1)/2][y]=1;
139
140         for(i=0;i<2;i++)
141             if (((x+1)/2==tx[i]) && (y==ty[i]))
142                 matrix[(x+1)/2][y]=2;
143
144         gotoxy(1, 1);
145         display_map(matrix, tx, ty);
146
147         count=0;
148         for(i=1;i<=10;i++)
149             for(j=1;j<=10;j++)
150                 if (matrix[i][j]==2)
151                     count=count+1;
152         gotoxy(1,12);
153         printf("찾은 보물(★)의 개수 : %d", count);
154         key=getch();
155         move_arrow_key(key, &x, &y, 19, 10);
156     }while(count<2);
157 }
```

3. 실행 화면



3. 실행 화면

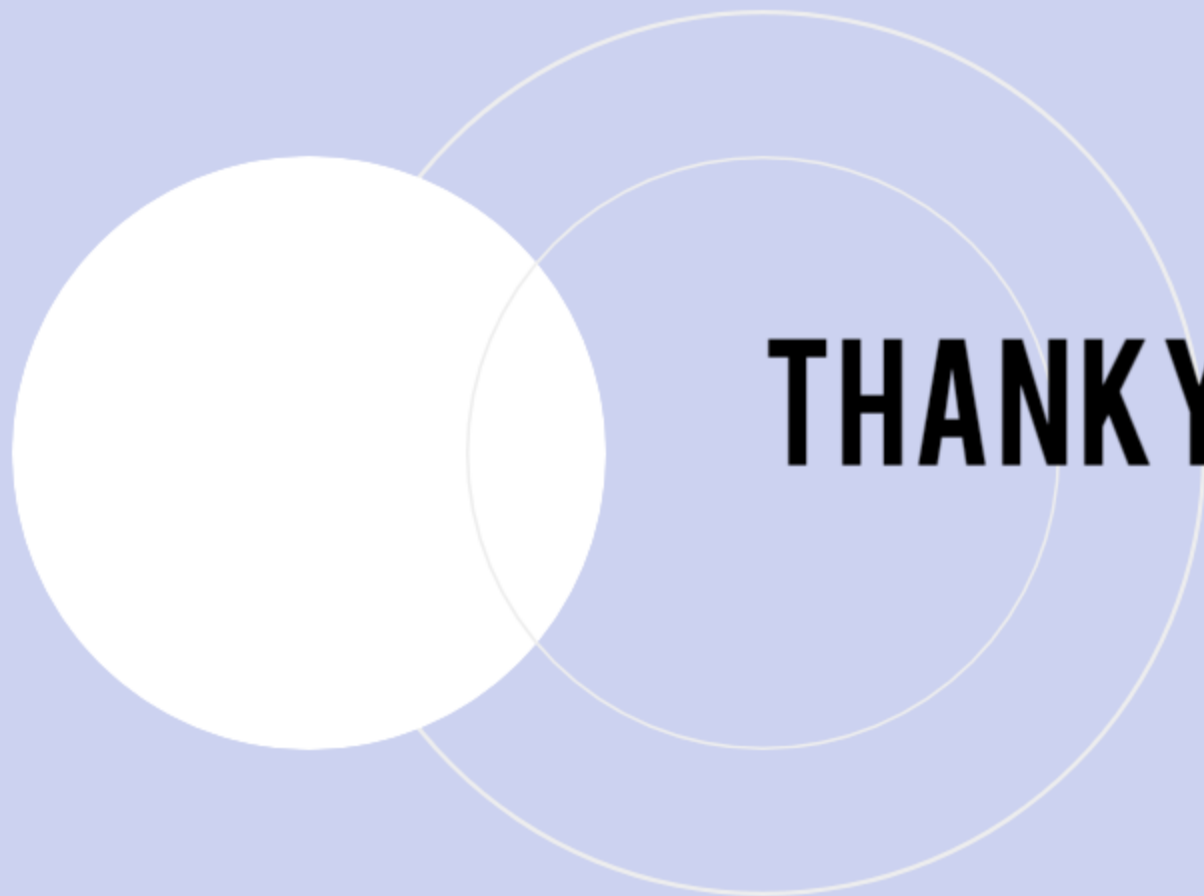
```
C:\Users\Wgkry\Downloads>
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□*□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □

찾은 보물(*)의 개수 : 2
```

```
C:\Users\Wgkry\Downloads>
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□*□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □
□□□□□□□□ □ □ □ □

찾은 보물(*)의 개수 : 2
경과 시간 : 49.0 초
모두 찾았습니다. 게임을 종료합니다.

-----
Process exited after 58.95 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . . |
```



THANK YOU!

