

응용 8_11_1

게임 프로그래밍

응용 프로그램 발표

영상애니메이션학부

2021978029

윤정화



목차

1. 게임 설명

2. 소스 분석

3. 추가 기능

청기백기

플레이어 수를 정하여 각각 청기백기 게임을 한 다음, 정답 수와 속도에 따라 승리하는 게임

청기/백기 게임

게임에 참가할 인원수를 입력하고 Enter>1

문제의 개수를 입력하고 Enter>5

1번 참가자의 이름을 입력하고 Enter>참가자1

만약 한글을 사용하였다면 영문키로 바꾸십시오.

아무키나 누르시오.

게임 진행

1: 청기 올려!

정해진 문제 수 만큼 진행



5: 청기 올리지말고 청기 내려!

모든 게임이 끝났습니다.
아무키나 누르면 결과를 표시합니다.

게임 결과

```
참가자1
1번 : X 5.29
2번 : X 1.69
3번 : X 5.07
4번 : X 1.31
5번 : O 1.00
```

-> 정답과 오답
-> 문제 당 걸린 시간

```
맞은개수: 1
소요시간: 1.00
```

-> 총 결과

```
-----
Process exited after 48.37 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

소스 분석 함수 선언

-헤더 파일

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <conio.h>
4  #include <time.h>
5  #include <windows.h>
```

-함수 선언

```
7  void gotoxy(int x, int y);
8  void game_control(int order, char name[7], char sol[][10], double check[][10], int quest);
9  void display_result(int user, int order[], double check[][10], char name[][7], char sol[][10], int quest);
10 void display_rule(void);
```

소스 분석 메인

```
12 int main(void)
13 {
14     int i, user_number, quest_number, rnd, temp;
15     int order[4]={0, 1, 2, 3};
16     char solution[4][10]={0};
17     double time_check[4][10]={0};
18     char user_name[4][7];
19     srand(time(NULL));
20
21     printf("청기/백기 게임 \n\n");
22     printf("게임에 참가할 인원수를 입력하고 Enter>");
23     scanf("%d", &user_number);
24     printf("\n문제의 개수를 입력하고 Enter>");
25     scanf("%d", &quest_number);
26
27     for(i=0; i<user_number; i++)
28     {
29         printf("%d번 참가자의 이름을 입력하고 Enter>", i+1);
30         scanf("%s", &user_name[i]);
31     }
32     printf("만약 한글을 사용하였다면 영문키로 바꾸십시오.\n");
33     printf("아무키나 누르시오.\n");
34     getch();
35     printf("\n게임의 순서는 다음과 같습니다. \n");
36     for(i=0; i<user_number; i++)
37     {
38         rnd=rand()%user_number;
39         temp=order[i];
40         order[i]=order[rnd];
41         order[rnd]=temp;
42     }
```

->참가자 순서를 저장하고, 무작위로 섞기

- 참가자 수와 문제 수 입력

- rand로 참가자 무작위로 섞기

- 게임 결과 표시

```
43     for(i=0; i<user_number; i++)
44     {
45         printf("%d번 : %s\n", i+1, user_name[order[i]]);
46     }
47     printf("아무키나 누르시오.\n");
48     getch();
49
50     for(i=0; i<user_number; i++)
51         game_control(order[i], user_name[order[i]], solution,
52                       time_check, quest_number);
53
54     printf("\n\n");
55     printf("모든 게임이 끝났습니다. \n");
56     printf("아무키나 누르면 결과를 표시합니다. \n");
57     getch();
58     display_result(user_number, order, time_check, user_name,
59                   solution, quest_number);
60     getch();
61     return 0;
62 }
```

소스 분석 게임 진행

```
70 void game_control(int order, char name[7], char sol[][10], double check[][10], int quest)
71 {
72     char *flag[]={"청 기 ", "백 기 "};
73     char *question[]={"내 러 !", "올 러 !", "내 리 지 말 고 ", "올 리 지 말 고 "};
74     clock_t start, end;
75     int i, rnd[4];
76     char key;
77     system("cls");
78     gotoxy(1, 3);
79     printf("%s님 순 서 입 니 다 . 키 조 작 은 다 음 과 같 습 니 다 .", name);
80     display_rule();
81     for(i=0;i<quest;i++)
82     {
83         system("cls");
84         rnd[0]=rand()%2;
85         rnd[1]=rand()%4;
86         start=clock();
87         gotoxy(5, 10);
88         if (rnd[1]>1)
89         {
90             rnd[2]=rand()%2;
91             rnd[3]=rand()%2;
92             printf("%d: %s %s %s %s", i+1, flag[rnd[0]], question[rnd[1]], flag[rnd[2]], question[rnd[3]]);
93         }
```

- getch로 참가자의 키 입력받기

- 게임 진행 중 시간 측정

- 주어진 명령 랜덤하게 조합

```
94     else
95     printf("%d: %s %s", i+1, flag[rnd[0]], question[rnd[1]]);
96     key=getch();
97     end=clock();
98     check[order][i]=(double)(end-start)/CLK_TCK;
99     if (rnd[1]>1)
100     {
101         if ((rnd[2]==0) && (rnd[3]==0) && (key=='z'))
102             sol[order][i]=1;
103         else if ((rnd[2]==0) && (rnd[3]==1) && (key=='a'))
104             sol[order][i]=1;
105         else if ((rnd[2]==1) && (rnd[3]==0) && (key=='m'))
106             sol[order][i]=1;
107         else if ((rnd[2]==1) && (rnd[3]==1) && (key=='k'))
108             sol[order][i]=1;
109         else
110             sol[order][i]=0;
111     }
112     else
113     {
114         if ((rnd[0]==0) && (rnd[1]==0) && (key=='z'))
115             sol[order][i]=1;
116         else if ((rnd[0]==0) && (rnd[1]==1) && (key=='a'))
117             sol[order][i]=1;
118         else if ((rnd[0]==1) && (rnd[1]==0) && (key=='m'))
119             sol[order][i]=1;
120         else if ((rnd[0]==1) && (rnd[1]==1) && (key=='k'))
121             sol[order][i]=1;
122         else
123             sol[order][i]=0;
124     }
125 }
126 }
```

-> 키 입력

-> 시간 측정

소스 분석 게임 결과

```
128 void display_result(int user, int order[], double check[][10], char name[][7], char sol[][10], int quest)
129 {
130     int i, j, result[4]={0};
131     double min_time[4]={0};
132     system("cls");
133     printf("      ");
134     for(i=0;i<user;i++)
135         printf("%8s ", name[order[i]]);
136     printf("\n");
137     for(j=0;j<quest;j++)
138     {
139         printf("    %d번 : ", j+1);
140         for(i=0;i<user;i++)
141         {
142             if (sol[order[i]][j] && check[order[i]][j]<1)
143             {
144                 printf("O");
145                 result[order[i]]+=1;
146             }
147             else
148                 printf("X");
149             printf("%5.2f ", check[order[i]][j]);
150             if ((sol[order[i]][j]==1) && (check[order[i]][j]<1))
151                 min_time[order[i]]+=check[order[i]][j];
152         }
153         printf("\n");
154     }
155     printf("\n");
156     printf("맞 은 개 수 :");
157     for(i=0;i<user;i++)
158         printf("%7d ", result[order[i]]);
159     printf("\n");
160     printf("소 요 시 간 :");
161     for(i=0;i<user;i++)
162         printf("%7.2f ", min_time[order[i]]);
163     printf("\n");
164 }
```

- 게임이 끝난 후 결과 표시

- 참가자가 맞춘 문제 수와 소요시간을
정리 후 출력

소스 분석 게임 규칙

```
166 void display_rule(void)
167 {
168     gotoxy(1,4);
169     printf("게 임 시 작 전 까 지 키 를 누 르 지 마 세 요 !");
170     gotoxy(1, 6);
171     printf("청 기 올 려 : a      백 기 올 려 : k");
172     gotoxy(1, 7);
173     printf("청 기 내 려 : z      백 기 내 려 : m");
174     Sleep(2000);
175     gotoxy(1, 8);
176     printf("5초 후 에 게 임 을 시 작 합 니 다 . 준 비 하 세 요 .");
177     Sleep(5000);
178
179 }
```

- 게임 규칙 표시

- 참가자가 시작 전에 키를 누르지 않도록 안내

- 청기, 백기 키 안내

추가 기능 1. 색상

```
12 int main(void)
13 {
14     system("color F1"); //메인 색상 변경
15
16     int i, user_number, quest_number, rnd, temp;
17     int order[4]={0, 1, 2, 3};
18     char solution[4][10]={0};
19     double time_check[4][10]={0};
20     char user_name[4][7];
21     srand(time(NULL));
```

- 메인 색상 변경

- 정답일때 초록색 화면과 소리

- 오답일때 빨간색 화면과 소리

```
122     else if ((rnd[0]==1) && (rnd[1]==0) && (key=='m'))
123         sol[order][i]=1;
124     else if ((rnd[0]==1) && (rnd[1]==1) && (key=='k'))
125         sol[order][i]=1; //정답
126     else
127         sol[order][i]=0; //오답
128 }
129
130 // 각 문제당 결과 출력
131 if (sol[order][i] == 1)
132 {
133     // 정답 : 초록색 배경과 소리
134     system("color 2F");
135     printf("정답입니다!\n");
136     Beep(1000, 300); // 정답 소리
137 }
138 else
139 {
140     // 오답 : 빨간색 배경과 소리
141     system("color 4F");
142     printf("오답입니다!\n");
143     Beep(500, 300); // 오답 소리
144 }
145
146 Sleep(1000);
147
148 // 화면을 다시 원래대로 돌려놓기
149 system("color F1"); // 기본 색상
150 }
151 }
```

추가 기능 1. 색상

참가자1: PA

참가자2: PB

참가자3: PC

청기/백기 게임

게임에 참가할 인원수를 입력하고 Enter>3

문제의 개수를 입력하고 Enter>5

1번 참가자의 이름을 입력하고 Enter>PA

2번 참가자의 이름을 입력하고 Enter>PB

3번 참가자의 이름을 입력하고 Enter>PC

만약 한글을 사용하였다면 영문키로 바꾸십시오.
아무키나 누르시오.

게임의 순서는 다음과 같습니다.

1번 : PC

2번 : PA

3번 : PB

아무키나 누르시오.

4: 백기 올리지말고 백기 내려!정답입니다!

2: 백기 내려!오답입니다!

추가 기능 2. 시작 전 키 입력 방지

- 시작 전 키 입력시 첫 문제 오답 버그
- 입력 버퍼를 비워 키 입력 방지

```
214 void display_rule(void)
215 {
216     gotoxy(1,4);
217     printf("게 임 시 작 전 까 지 키 를 누 르 지 마 세 요 !");
218     gotoxy(1, 6);
219     printf("청 기 올 려 : a      백 기 올 려 : k");
220     gotoxy(1, 7);
221     printf("청 기 내 려 : z      백 기 내 려 : m");
222     Sleep(2000);
223     gotoxy(1, 8);
224     printf("5초 후 에 게 임 을 시 작 합 니 다 . 준 비 하 세 요 .");
225     Sleep(5000);
226
227     // 콘솔 입력 버퍼를 비우기
228     FlushConsoleInputBuffer(GetStdHandle(STD_INPUT_HANDLE));
229 }
```

추가 기능 3. 승자 발표

- 통계만으로 승자를 빨리 알기 어려움
- 승자 발표로 빠르게 결과를 알 수 있음

```
153 void display_result(int user, int order[], double check[][10], char name[][7], char sol[][10], int quest)
154 {
155     int i, j, result[4]={0};
156     double min_time[4]={0};
157     int winner_index = 0; // 승자의 인덱스
158     system("cls");
```

```
201 {
202     // 동점일 경우, 소요 시간이 적은 사람이 승리
203     if (min_time[order[i]] < min_time[order[winner_index]])
204     {
205         winner_index = i;
206     }
207 }
208 }
209
210 // 승자 발표
211 printf("\n승자는 %s입니다. 종료하려면 아무키나 누르시오.\n", name[order[winner_index]]);
212 }
```

추가 기능 3. 승자 발표

```
참가자1
1번 : X 5.29
2번 : X 1.69
3번 : X 5.07
4번 : X 1.31
5번 : O 1.00

맞은개수: 1
소요시간: 1.00
```

Process exited after 48.37 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .



	PC	PA	PB
1번	: X 1.82	X 0.77	O 0.85
2번	: X 3.36	X 0.61	O 0.76
3번	: X 1.17	X 0.54	O 0.57
4번	: X 1.71	X 0.00	X 1.01
5번	: O 0.78	X 0.00	O 0.76

맞은개수:	1	0	4
소요시간:	0.78	0.00	2.94

승자는 PB입니다. 종료하려면 아무키나 누르시오.

Process exited after 104.8 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .