

미니게임 월드

```
=====Mini Game World=====
                                           Made by 백종욱

    *미니게임월드 에 오신 것을 환영합니다*
*한 개의 미니게임을 클리어 할 때마다 메달이 주어집니다*
* 3개의 메달을 모두 모으시면 마지막 스테이지가 열립니다*
*마지막 스테이지를 클리어 하시면 게임이 클리어됩니다*
    *종료하시려면 0을 눌러주세요*
    *계속하시려면 1을 눌러주세요*

=====Mini Game World=====
```

2021. 7. 19 / 로봇학부 / 백종욱

제작 동기



게임 종류

01



가위바위보

02



연타게임

03



흑과 백

04



같은 그림 찾기

사용 함수

main

Main menu

가위바위보

연타게임

흑과 백

유저 턴

컴퓨터 턴

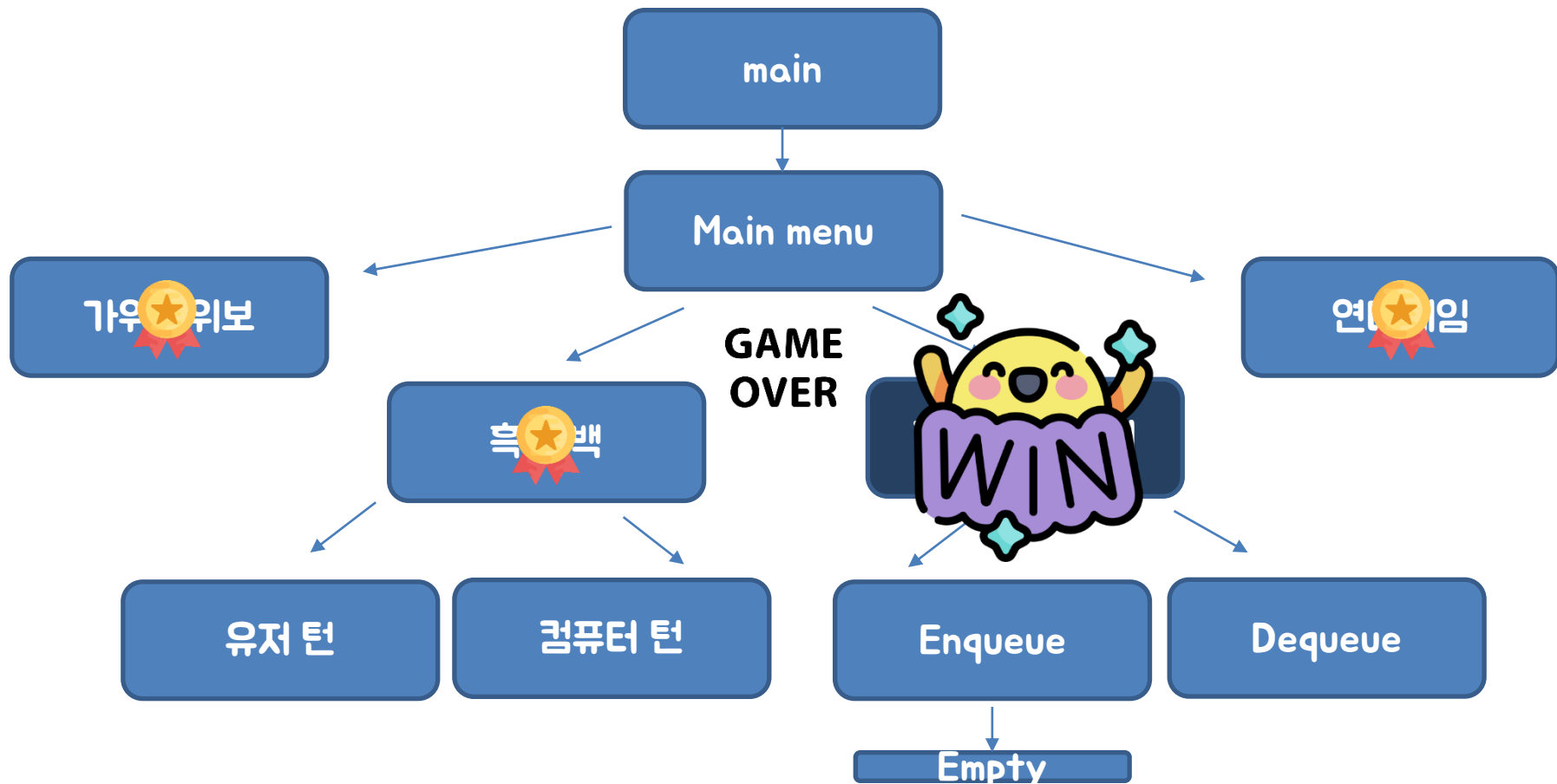
같은 그림 찾기

Enqueue

Dequeue

empty

프로그램 작동 과정



코드 설명

메인 함수

를 안내

```
40 {
41     int startgame;
42
43     printf("=====Mini Game World=====\\n");
44     printf("                Made by 백종욱\\n");
45     printf("\\n        *미니게임월드에 오신 것을 환영합니다*        \\n");
46     printf("**한 개의 미니게임을 클리어 할 때마다 메달이 주어집니다*\\n");
47     printf("** 3개의 메달을 모두 모으시면 마지막 스테이지가 열립니다*\\n");
48     printf(" *마지막 스테이지를 클리어 하시면 게임이 클리어됩니다* \\n");
49     printf("        *종료하시려면 0을 눌러주세요*        \\n");
50     printf("        *계속하시려면 1을 눌러주세요*        \\n");
51     printf("\\n=====Mini Game World=====\\n");
52
53     do
54     {
55         scanf_s("%d", &startgame);
56
57         if (startgame == 0)
58         {
59             printf("게임을 종료합니다\\n");
60             break;
61         }
62
63         else if (startgame == 1)
64         {
65             system("cls");
66             mainmenu();
67         }
68
69         else
70             printf("올바른 번호를 입력해주세요!\\n");
71     } while (startgame != 1);
72
73     return 0;
74 }
```

스타트게임 변수에 값을 입력받아
0이면 종료,
1이면 메인 메뉴 실행,
그 외의 숫자라면 예외 처리

화면 출력을 지우는 함수

메인 메뉴

```
76 void mainmenu()
77 {
78     int select;
79     int gameover = 0;
80
81     printf("=====게임 선택=====\n");
82     printf("1.가위바위보\n");
83     printf("2.연타게임\n");
84     printf("3.흑과 백\n");
85     printf("4.FINAL STAGE\n");
86     printf("5.지금까지의 메달 확인\n");
87     printf("0.게임 종료\n");
88     printf("=====게임 선택=====\n");
89
90     while (gameover != 1)
91     {
92         scanf_s("%d", &select);
93
94         switch (select)
95         {
96             case 0:
97                 printf("게임을 종료합니다\n");
98                 gameover++;
99
100                break;
101
102                case 1:
103                    if (clear_RSP == 1)
104                        printf("이 게임은 이미 클리어하셨습니다\n");
105
106                    else
107                    {
108                        RSP();
109                        gameover++;
110                    }
111                }
```

게임 선택 = 셀렉트 변수에 값 입력

셀렉트 변수의 값에 따라 각 게임 함수 실행
이때, 게임을 클리어해야 1증가하는 클리어 변수가
1일 때(즉, 게임을 클리어한 경우) 는 예외 문구를
출력한 후 다른 명령을 기다림

게임을 클리어하지 못했을 경우, 게임오버 변수를
증가시켜 반복문을 빠져나와 프로그램 종료

가위바위보

```
181 while ((win != 2) && (lose != 2))
182 {
183     printf("\n현재 이긴 횟수 : %d\n", win);
184     printf("현재 진 횟수 : %d\n", lose);
```

```
185     printf("1.가위\n");
186     printf("2.바위\n");
187     printf("3.보\n");
```

```
188
189 while (1)
190 {
191     scanf_s("%d", &user);
192     system("cls");
```

```
193     printf("당신이 낸 것 : ");
194     if (user == 1)
195     {
196         printf("가위\n");
197         break;
198     }
199     else if (user == 2)
200     {
201         printf("바위\n");
202         break;
203     }
204     else if (user == 3)
205     {
206         printf("보\n");
207         break;
208     }
209 }
```

사용자는 가위, 바위, 보에 해당하는
번호를 입력, 그것을 user 변수에 저장

가위바위보

```
220 com = rand() % 3 + 1;
221
222 printf("컴퓨터가 낸 것 : ");
223 if (com == 1)
224     printf("가위\n");
225
226 else if (com == 2)
227     printf("바위\n");
228
229 else if (com == 3)
230     printf("보\n");
```

컴퓨터 또한 1부터 3까지의 난수를 생성
하여 com변수에 저장 후,
그 숫자에 맞는 행동 출력

```
232 if (((user == 1) && (com == 3)) || ((user == 2) && (com == 1)) || ((user == 3) && (com == 2)))
233 {
234     printf("컴퓨터와의 승부에서 승리하였습니다!\n");
235     win++;
236 }
237
238 else if (((user == 1) && (com == 2)) || ((user == 2) && (com == 3)) || ((user == 3) && (com == 1)))
239 {
240     printf("컴퓨터와의 승부에서 패배하였습니다!\n");
241     lose++;
242 }
243
244 else
245     printf("비겼습니다!\n");
246 }
247
```

각 경우의 수를 따져 사용자가
이겼을 경우 win변수를 증가시키고,
졌을 경우 lose변수를 증가시킴
비겼을 경우 변수의 증가 없음

가위바위보

게임에서 2판 이겨 win변수가 2가 되었을 경우, 클리어 메시지를 출력하고 변수를 증가시키며, 메달을 하나 얻음. 1을 누르면 메인 화면으로 돌아감.

```
247 if (win == 2)
248 {
249     printf("\n축하합니다! 가위바위보 게임을 클리어하셨습니다!\n");
250     clear_RSP++;
251
252     printf("보상으로 메달 1개를 드립니다\n");
253     medal++;
254
255     printf("계속하시려면 1을 입력해주세요\n");
256     printf("종료하시려면 아무 키나 입력해주세요\n");
257     scanf_s("%d", &select);
258
259     if (select == 1)
260     {
261         system("cls");
262         mainmenu();
263     }
264 }
265 }
```

```
266 else
267     printf("\nGAME OVER!\n");
268 }
269 }
```

두 번 졌을 경우 게임 오버

연타게임



연타게임

```
clock_t start, end;  
int win = 0, lose = 0;  
int select, com_time;  
char space;
```

시간 변수 설정(시간 측정용)

```
if (win == 0)  
    com_time = 150;  
  
else  
    com_time = 130;
```

컴퓨터를 한 번 이겼을 때, 컴퓨터의
속도는 더 빨라짐

```
for (int i = 1; i <= 30; i++)  
{
```

```
    space = _getch();
```

개행 문자 없이 입력받기

```
    if (space == 32)  
    {  
        printf("%d ", i);
```

```
        if (i == 1)  
            start = clock();
```

```
    }  
}  
  
end = clock();  
  
double time_gap = (double)(end - start) / CLOCKS_PER_SEC;  
printf("\n%g초 걸렸습니다\n", time_gap);
```

처음 누른 순간부터 30회 다 눌렀을 때까
지 시간 측정 후, 타이п 갭 변수에 저장

연타게임

```
printf("\n컴퓨터 : ");
```

```
for (int i = 1; i <= 30; i++)
```

```
{
```

컴퓨터 또한 30번의 입력을 받음

```
    if (i == 1)
```

```
        start = clock();
```

```
    printf("%d ", i);
```

```
    Sleep(com_time);
```

0.001초 기다리는 함수(딜레이)

```
}
```

```
end = clock();
```

```
double com_time_gap = (double)(end - start) / CLOCKS_PER_SEC;
```

```
printf("\n%g초 걸렸습니다\n", com_time_gap);
```

처음 누른 순간부터 30회 다 눌렀을
때까지 시간 측정 후,
컴퓨터 타이밍 변수에 저장

연타게임

```
else if (time_gap < com_time_gap)
{
    win++;
    printf("\n컴퓨터와의 승부에서 승리하였습니다!\n");
    printf("준비가 되었다면 1을 입력해주세요\n");

    while (1)
    {
        scanf_s("%d", &select);

        if (select == 1)
            break;
        else
            printf("올바른 값을 입력해주세요!\n");
    }

    system("cls");
}

else
{
    lose++;
    printf("\n컴퓨터와의 승부에서 패배하였습니다!\n");
    printf("준비가 되었다면 1을 입력해주세요\n");

    while (1)
    {
        scanf_s("%d", &select);

        if (select == 1)
            break;
        else
            printf("올바른 값을 입력해주세요!\n");
    }
}
```

사용자의 시간차가 컴퓨터의 시간차
보다 작은 경우 승리, win변수 증가

아닌 경우, 패배, lose함수 증가

게임에서 2판 이겨 win변수가 2가 되
었을 경우, 클리어 메시지를 출력하고
변수를 증가시키며, 메달을 하나 얻음.
1을 누르면 메인 화면으로 돌아감.

두 번 졌을 경우 게임 오버

흑과 백

흑과백



1. 데스매치대상자 2명은 0~8까지 9장의 숫자타일을 지급받는다.

흑과백



2. 9개의 숫자타일은 흑색, 백색으로 나뉘며 0, 2, 4, 6, 8은 흑색, 1, 3, 5, 7은 백색타일로 구성되어 있다.

흑과백

2 Round



3. 1라운드의 선플레이어는 가넷이 많은 플레이어가 결정하며, 2라운드부터는 전 라운드에 승리한 플레이어가 선플레이어가 된다.

흑과백



4. 선플레이어가 0~8까지의 숫자타일 중 1개를 뒷면이 보이도록 제시한 뒤, 후 플레이어가 타일을 제시한다.

흑과 백

```
422 int i;  
423 int user, com, select;  
424 int turn = 1;  
425  
426 int user_tile[9] = { 0, 1, 2, 3, 4, 5, 6, 7, 8 };  
427 int com_tile[9] = { 0, 1, 2, 3, 4, 5, 6, 7, 8 };  
428 int used_user_tile[9];  
429 int used_com_tile[9];  
430 int win = 0, lose = 0;  
431  
432 srand((unsigned)time(NULL));  
433
```

타일 배열 선언

```
449  
450 if (turn % 2 != 0)  
451 {  
452     printf("당신이 먼저 제시할 차례입니다\n");  
453     useturn(&user, user_tile, used_user_tile);  
454     comturn(&com, com_tile, used_com_tile);  
455 }  
456  
457 else  
458 {  
459     printf("컴퓨터가 먼저 제시할 차례입니다\n");  
460     comturn(&com, com_tile, used_com_tile);  
461     useturn(&user, user_tile, used_user_tile);  
462 }
```

턴제로 돌아가며 진행

흑과 백

사용자가 제시한 값을 포인터 변수로 받고, 타일 배열과 사용한 타일 배열을 매개변수로 받음

만약 사용자가 유효한 숫자의 타일을 제시했다면, 그 타일을 사용한 타일 배열에 복사함. 다음에 사용한 타일을 다시 입력받았을 경우, 메시지 출력

사용자가 제시한 타일은 타일 배열에서 -1로 표기되도록 설정

제시한 타일이 홀수면 백색, 짝수면 흑색이라고 알림

컴퓨터도 똑같은 과정을 거침

```
544 void userturn(int *user, int user_tile[], int used_user_tile[])
545 {
546     printf("\n현재 당신의 타일 : \n");
547
548     for (int j = 0; j < 9; j++)
549         printf("%d ", user_tile[j]);
550
551     printf("\n\n널 타일의 숫자를 입력해주시시오 : ");
552
553     while (1)
554     {
555         scanf_s("%d", &(*user));
556
557         if (*user == used_user_tile[*user])
558             printf("그 타일은 이미 사용했습니다!\n");
559
560         else if (*user >= 0 && *user <= 8)
561         {
562             used_user_tile[*user] = *user;
563             break;
564         }
565
566         else
567             printf("올바른 숫자를 입력해주세요!\n");
568     }
569
570     for (int j = 0; j < 9; j++)
571     {
572         if (*user == user_tile[j])
573         {
574             user_tile[j] = -1;
575             break;
576         }
577     }
578
579     if (*user % 2 != 0)
580         printf("\n당신이 제시한 타일의 색은 *백색*입니다\n");
581
582     else
583         printf("\n당신이 제시한 타일의 색은 *흑색*입니다\n");
584 }
```

흑과 백

```
478 if (user > com)
479 {
480     printf("당신의 승리\n\n");
481     win++;
482 }
483
484 else if (user < com)
485 {
486     printf("컴퓨터의 승리\n\n");
487     lose++;
488 }
489
490 else
491     printf("비겼습니다\n\n");
492
493 turn++;
494 }
```

사용자가 제시한 타일이 컴퓨터가 제시한 타일보다 클 경우, 승리

아니면 패배

경기 후 턴 변수에 1을 더해 번갈아
선 플레이어를 정하도록 함

같은 그림 찾기



같은 그림 찾기

```
619 Queue* queue = malloc(sizeof(Queue)); //큐 구조체 변수 선언
620 queue->front = NULL; //큐에서 front를 초기화
621 queue->rear = NULL; //큐에서 rear를 초기화
```

```
622
623 int number[4][4];
624 char alphabet[4][4];
625 int i, j, select;
626 int tofront = 9, torear = 10;
627 int count = 1;
628
629 srand((unsigned)time(NULL));
```

```
630
631 for (i = 0; i < 4; i++)
632 {
633     for (j = 0; j < 4; j++)
634     {
635         number[i][j] = count;
636         count++;
637     }
638 }
```

```
639
640 count = 0;
```

```
641
642 for (i = 0; i < 4; i++)
643 {
644     for (j = 0; j < 4; j++)
645     {
646         alphabet[i][j] = count + 65;
647         count++;
648     }
649 }
```

타일 뒷면 배열과 타일 앞면 배열 선언

도착 지점과 탈락 지점까지 얼마나 남았는
지를 계산하는 tofront, torear 변수 선언

타일의 앞면을 1부터 16개의 숫자로 초기화

타일의 뒷면을 A~P의 16개의 문자로 초기화

같은 그림 찾기

타일의 뒷면 배열을 무작위로 셔플

큐(컨베이어 벨트)에 A~P까지의 문자 중 무작위로 20개를 엔큐
You라는 변수를 추가한 후, 앞에서부터 9번 움직인 위치 위치하도록 초기화

타일의 앞면 출력

```
650
651 for (int i = 0; i < 100; i++)
652 {
653     int randNum1 = rand() % 4;
654     int randNum2 = rand() % 4;
655     int randNum3 = rand() % 4;
656     int randNum4 = rand() % 4;
657
658     int temp;
659
660     temp = alphabet[randNum1][randNum2];
661
662     alphabet[randNum1][randNum2] = alphabet[randNum3][randNum4];
663
664     alphabet[randNum3][randNum4] = temp;
665 }
```

```
666
667 for(i = 0; i < 20; i++)
668     Enqueue(queue, rand() % 16 + 65);
669
670 Node* you = queue->front;
671
672 for (i = 0; i < 9; i++)
673     you = you->next;
674
```

```
675 while (1)
676 {
677     printf(" *게임 타일* \n");
678
679     for (i = 0; i < 4; i++)
680     {
681         for (j = 0; j < 4; j++)
682             printf("%3d", number[i][j]);
683         printf("\n");
684     }
```

같은 그림 찾기

각종 정보 출력

```
685 printf("\n도착 지점까지 %d칸 남았고,\n", torear);
686 printf("\n탈락 지점까지 %d칸 남았습니다\n", tofront);
687
688
689 printf("\n당신이 찾아야 할 다음 타일 : %c\n", you->data);
690
691 printf("\n열어볼 타일의 번호를 입력하세요 : ");
692 scanf_s("%d", &select);
693
694 for (i = 0; i < 4; i++)
695 {
696     for (j = 0; j < 4; j++)
697     {
698         if (select == number[i][j])
699         {
700             printf("\n%d 번 타일의 문자는 %c 입니다\n", number[i][j], alphabet[i][j]);
701
702             if (alphabet[i][j] == you->data)
703             {
704                 printf("\n맞추었습니다 한 칸 앞으로 전진합니다\n\n");
705                 you = you->next;
706                 torear--;
707             }
708
709             else
710             {
711                 printf("\n틀렸습니다 컨베이어벨트가 한 칸 후진합니다\n");
712                 Enqueue(queue, rand() % 16 + 65);
713                 Dequeue(queue);
714                 torear++;
715             }
716
717             tofront = 19 - torear;
718
719             printf("\n계속하시려면 1을 입력해주세요\n");
720
```

You의 데이터값이 뒤집은 타일의 뒷면과 일치한다면, 한칸 전진
전진 동작은 You변수를 한칸 움직임

틀렸으면 컨베이어벨트 움직임
컨베이어벨트의 rear(앞)에는 새로운 문자가 하나 추가되고, 맨 뒤(front)에 있는 문자는 디큐

Q & A

질의응답시간

감사합니다