

Fruit Tetris

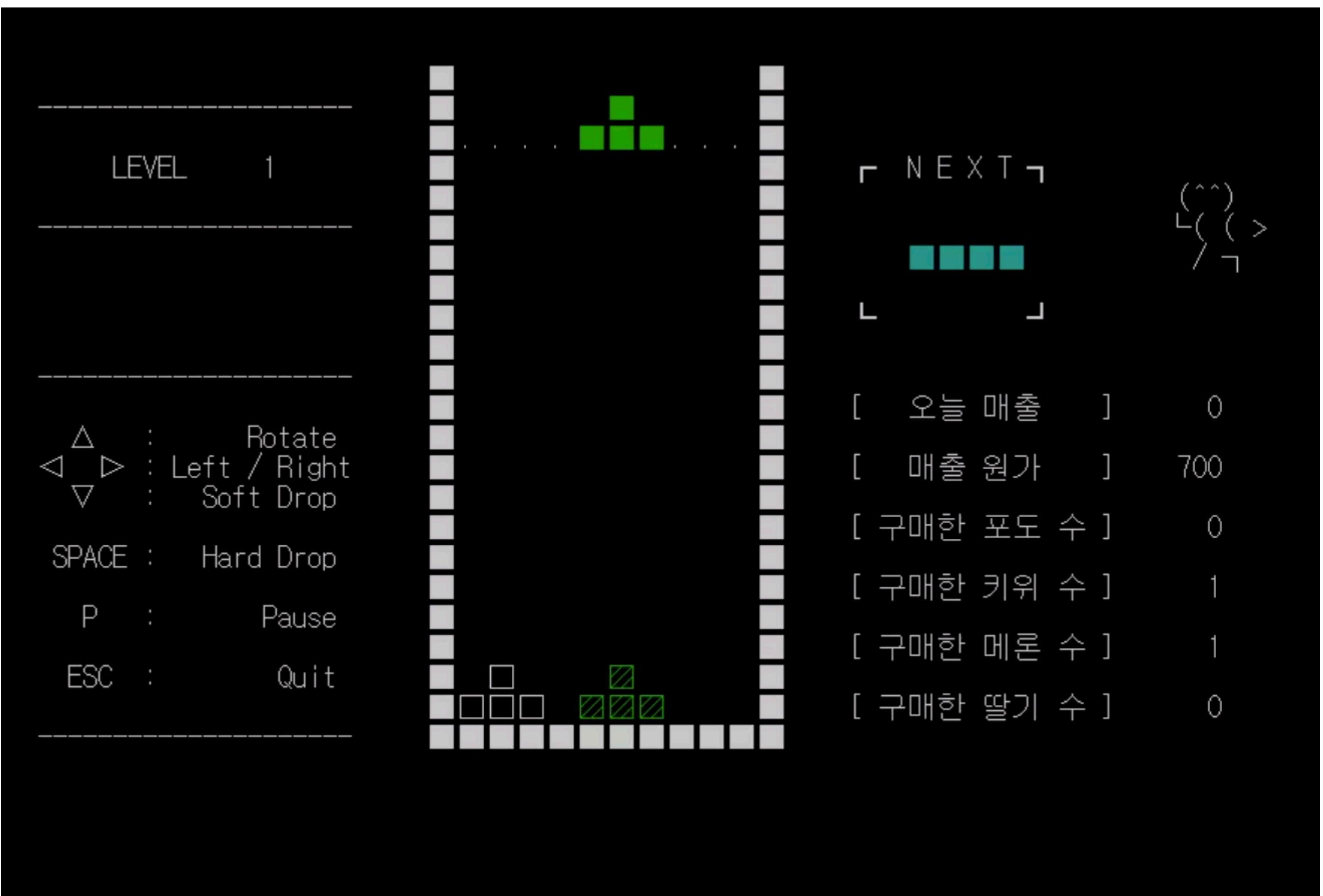
BCP Term Project

Group 5 김민서, 김혜리, 이선민, 이지원

Contents

Fruit Tetris

- Concept Description
- Demonstration
- Code Description
- Feedback



Concept Description

Fruit Tetris

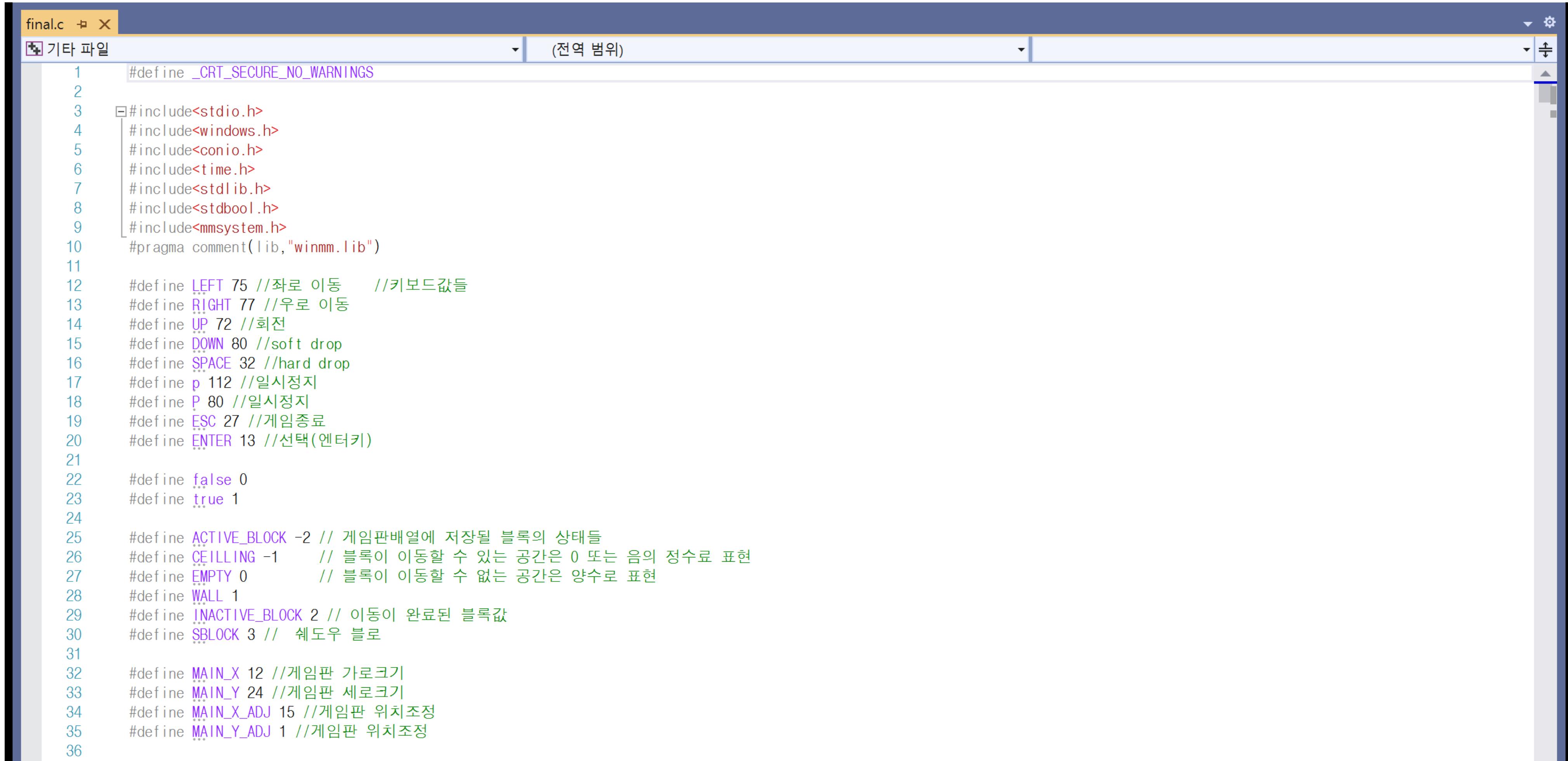
- Story Telling: the main character who lost money in stocks and bitcoin will become a fruit seller and make money
- Each block has its own color and is considered a fruit
- Generated blocks are considered a purchased fruit (cost of sales) and is considered to be sold when a line is completed (sales)

Demonstration

Introduction

Code Description

define

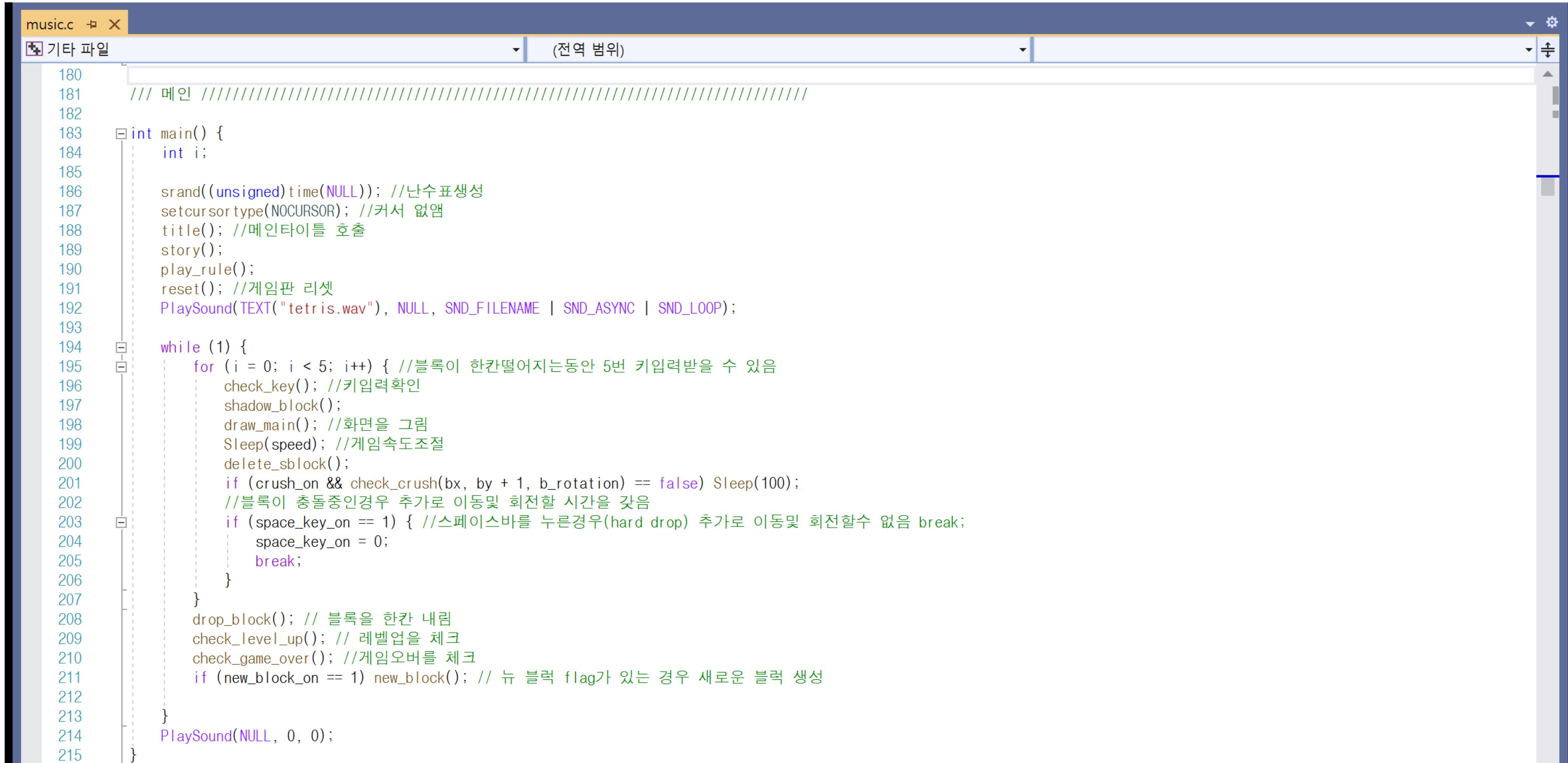


```
final.c
 기타 파일 (전역 범위)

1 #define _CRT_SECURE_NO_WARNINGS
2
3 #include<stdio.h>
4 #include<windows.h>
5 #include<conio.h>
6 #include<time.h>
7 #include<stdlib.h>
8 #include<stdbool.h>
9 #include<mmsystem.h>
10 #pragma comment(lib, "winmm.lib")
11
12 #define LEFT 75 //좌로 이동 //키보드값들
13 #define RIGHT 77 //우로 이동
14 #define UP 72 //회전
15 #define DOWN 80 //soft drop
16 #define SPACE 32 //hard drop
17 #define p 112 //일시정지
18 #define P 80 //일시정지
19 #define ESC 27 //게임종료
20 #define ENTER 13 //선택(엔터키)
21
22 #define false 0
23 #define true 1
24
25 #define ACTIVE_BLOCK -2 // 게임판배열에 저장될 블록의 상태들
26 #define CEILING -1 // 블록이 이동할 수 있는 공간은 0 또는 음의 정수로 표현
27 #define EMPTY 0 // 블록이 이동할 수 없는 공간은 양수로 표현
28 #define WALL 1
29 #define INACTIVE_BLOCK 2 // 이동이 완료된 블록값
30 #define SBLOCK 3 // 쉐도우 블록
31
32 #define MAIN_X 12 //게임판 가로크기
33 #define MAIN_Y 24 //게임판 세로크기
34 #define MAIN_X_ADJ 15 //게임판 위치조정
35 #define MAIN_Y_ADJ 1 //게임판 위치조정
36
```

Code Description

main



The screenshot shows a code editor window with the file "music.c" open. The code is written in C and describes the main loop of a game. The code is annotated with Korean comments explaining the logic of each section.

```
music.c  X
기타 파일 (전역 범위)

180
181     /// 메인 /////////////////////////////////
182
183 int main() {
184     int i;
185
186     srand((unsigned)time(NULL)); //난수표생성
187     setcursor type(NOCURSOR); //커서 없앰
188     title(); //메인타이틀 호출
189     story();
190     play_rule();
191     reset(); //게임판 리셋
192     PlaySound(TEXT("tetris.wav"), NULL, SND_FILENAME | SND_ASYNC | SND_LOOP);
193
194     while (1) {
195         for (i = 0; i < 5; i++) { //블록이 한칸떨어지는동안 5번 키입력받을 수 있음
196             check_key(); //키입력확인
197             shadow_block();
198             draw_main(); //화면을 그림
199             Sleep(speed); //게임속도조절
200             delete_sblock();
201             if (crush_on && check_crush(bx, by + 1, b_rotation) == false) Sleep(100);
202             //블록이 충돌중인경우 추가로 이동및 회전할 시간을 갖음
203             if (space_key_on == 1) { //스페이스바를 누른경우(hard drop) 추가로 이동및 회전할수 없음 break;
204                 space_key_on = 0;
205                 break;
206             }
207         }
208         drop_block(); // 블록을 한칸 내림
209         check_level_up(); // 레벨업을 체크
210         check_game_over(); //게임오버를 체크
211         if (new_block_on == 1) new_block(); // 뉴 블럭 flag가 있는 경우 새로운 블럭 생성
212     }
213 }
214
215 }
```

The code starts with a series of initialization and setup calls. It then enters a main loop where it repeatedly checks for user input (using `check_key()`), updates the game state (using `shadow_block()`, `draw_main()`, `Sleep()`, and `delete_sblock()`), and handles specific events like hard drops (using `space_key_on`). After each iteration of the inner loop, it performs a full block drop (`drop_block()`), checks for level up or game over conditions (`check_level_up()` and `check_game_over()`), and creates new blocks if necessary (`new_block()`). The loop concludes with a final call to `PlaySound()`.

Code Description

title

```
final.c  X
▣ 기타 파일 (전역 범위)
217     /// 타이틀 ///////////////////////////////////////////////////////////////////
218
219     void title(void) {
220         int x = 5; //타이틀화면이 표시되는 x좌표
221         int y = 4; //타이틀화면이 표시되는 y좌표
222         int cnt; //타이틀 프레임을 세는 변수
223
224         SetConsoleTextAttribute(GetStdHandle(STD_INPUT_HANDLE), ENABLE_MOUSE_INPUT);
225         system("mode con cols=110 lines=35 | title 5조)Fruit Tetris");
226
227         int i, j;
228
229         char * Menu[10][6] = {
230             { "███████", "███", "███████", "███", "███████", "███" },
231             { "███████", "███", "███████", "███", "███████", "███" },
232             { "    ███", "███", "    ███", "███", "    ███", "███" },
233             { "    ███", "███", "    ███", "███", "    ███", "███" },
234             { "    ███", "███", "    ███", "███", "    ███", "███" },
235             { "    ███", "███", "    ███", "███", "    ███", "███" },
236             { "    ███", "███", "    ███", "███", "    ███", "███" },
237             { "    ███", "███", "    ███", "███", "    ███", "███" },
238             { "    ███", "███", "    ███", "███", "    ███", "███" },
239             { "    ███", "███", "    ███", "███", "    ███", "███" },
240         };
241
242         gotoxy(x + 12, y + 14); printf("▶ Press any key to start ◀");
243
244         for (i = 0; i < 10; i++) {
245             gotoxy(x, y + i);
246             for (j = 0; j < 6; j++) {
247                 switch (j) {
248                     case 0:
249                         RED
250                         break;
251                     case 1:
252                         ORANGE
253                         break;
254                     case 2:
255                         GREEN
256                         break;
257                     case 3:
258                         BLUE
259                         break;
260                     case 4:
261                         PURPLE
262                         break;
263                     case 5:
264                         CYAN
265                         break;
266                 }
267             }
268         }
269
270         system("cls");
271         title();
272     }
273 }
```

Code Description

story

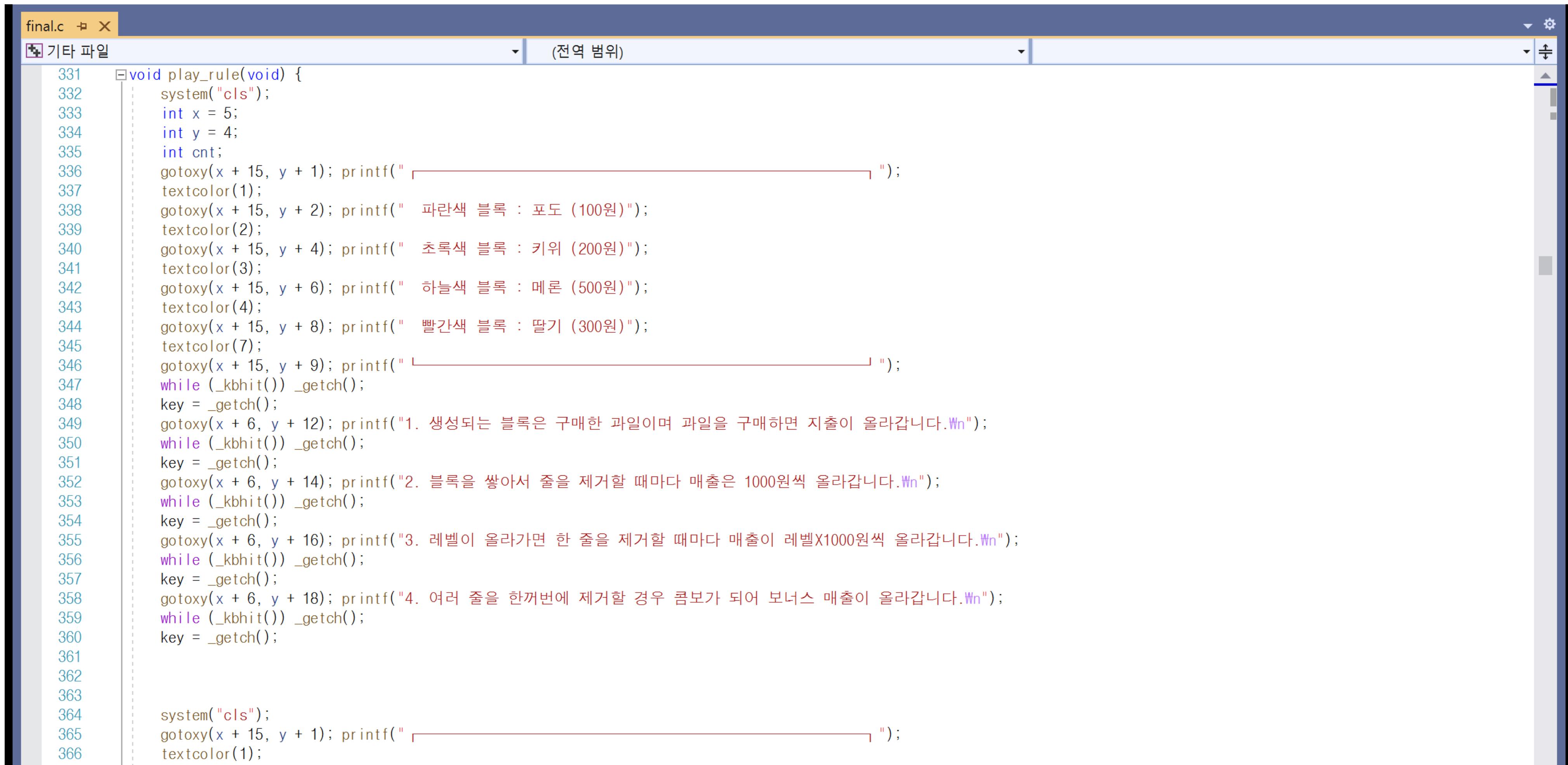
The screenshot shows a code editor window with the file 'final.c' open. The code is written in C and contains Korean comments and text. The code is organized into three main sections: a story section, a game instruction section, and a character and option section.

```
final.c  X
기타 파일 (전역 범위)

280 // 스토리 및 방법 /////////////////////////////////
281
282 void story(void) {
283     system("cls");
284     int x = 5;
285     int y = 4;
286
287     textcolor(7); //스토리
288     gotoxy(x + 1, y + 2); printf("202X년 X월 X일…");
289     while (_kbhit()) _getch();
290     key = _getch();
291     gotoxy(x + 1, y + 4); printf("일어나보니 나의 주식과 비트코인은 전부 휴지 조각이 되어 있었다…");
292     while (_kbhit()) _getch();
293     key = _getch();
294     gotoxy(x + 1, y + 6); printf("이렇게 된 이상 파일장수로 새 삶을 시작해보려 한다…!");
295     while (_kbhit()) _getch();
296     key = _getch();
297
298     textcolor(3); //게임 방법
299     gotoxy(x + 1, y + 12); printf("-----");
300     gotoxy(x + 8, y + 14); printf("< How to play >");
301     gotoxy(x + 1, y + 16); printf(" △ : Rotate");
302     gotoxy(x + 1, y + 17); printf(" ◇ ◇ : Left / Right");
303     gotoxy(x + 1, y + 18); printf(" ▽ : Soft Drop");
304     gotoxy(x + 12, y + 16); printf(" SPACE : Hard Drop");
305     gotoxy(x + 12, y + 17); printf(" P : Pause");
306     gotoxy(x + 12, y + 18); printf(" ESC : Quit");
307     gotoxy(x + 1, y + 20); printf("-----");
308
309     textcolor(7); //캐릭터 및 말풍선
310     gotoxy(x + 29, y + 12); printf("┌————┐");
311     gotoxy(x + 29, y + 13); printf(" |      |");
312     gotoxy(x + 29, y + 14); printf(" |      |");
313     gotoxy(x + 29, y + 15); printf(" |      |");
314     gotoxy(x + 29, y + 16); printf(" |      |");
```

Code Description

play_rule



The screenshot shows a code editor window with the file "final.c" open. The code is a C program containing a function named "play_rule". The function performs several tasks, including clearing the screen, setting initial coordinates (x=5, y=4), and printing four fruit items with their colors and prices. It then enters a loop where it prints a message about level-up rewards, waits for keyboard input, and updates the key variable. The code ends by clearing the screen again.

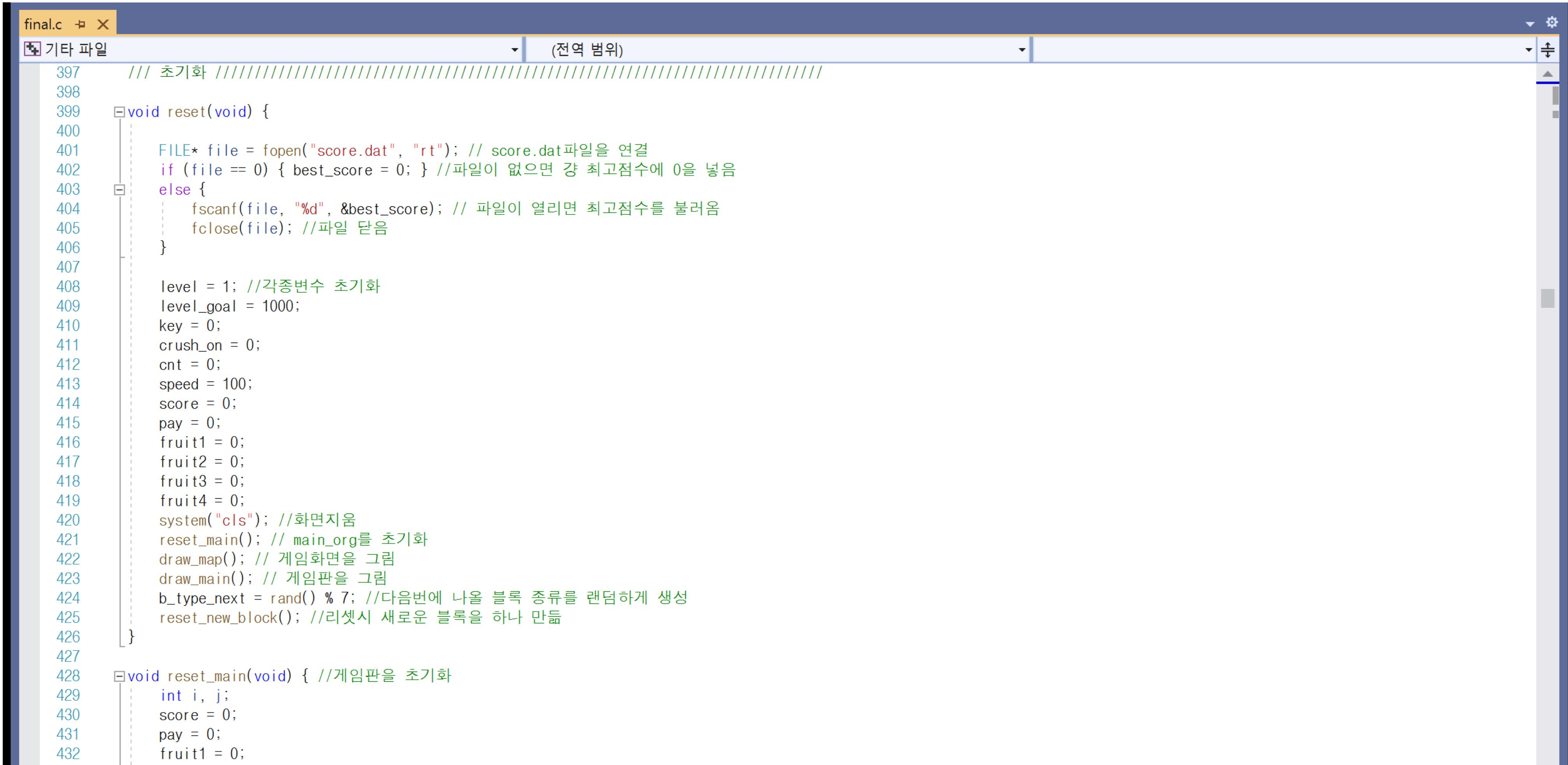
```
final.c  X
기타 파일 (전역 범위)

331 void play_rule(void) {
332     system("cls");
333     int x = 5;
334     int y = 4;
335     int cnt;
336     gotoxy(x + 15, y + 1); printf(" _____ ");
337     textcolor(1);
338     gotoxy(x + 15, y + 2); printf(" 파란색 블록 : 포도 (100원)");
339     textcolor(2);
340     gotoxy(x + 15, y + 4); printf(" 초록색 블록 : 키위 (200원)");
341     textcolor(3);
342     gotoxy(x + 15, y + 6); printf(" 하늘색 블록 : 메론 (500원)");
343     textcolor(4);
344     gotoxy(x + 15, y + 8); printf(" 빨간색 블록 : 딸기 (300원)");
345     textcolor(7);
346     gotoxy(x + 15, y + 9); printf(" _____ ");
347     while (_kbhit()) _getch();
348     key = _getch();
349     gotoxy(x + 6, y + 12); printf("1. 생성되는 블록은 구매한 파일이며 파일을 구매하면 지출이 올라갑니다.\n");
350     while (_kbhit()) _getch();
351     key = _getch();
352     gotoxy(x + 6, y + 14); printf("2. 블록을 쌓아서 줄을 제거할 때마다 매출은 1000원씩 올라갑니다.\n");
353     while (_kbhit()) _getch();
354     key = _getch();
355     gotoxy(x + 6, y + 16); printf("3. 레벨이 올라가면 한 줄을 제거할 때마다 매출이 레벨X1000원씩 올라갑니다.\n");
356     while (_kbhit()) _getch();
357     key = _getch();
358     gotoxy(x + 6, y + 18); printf("4. 여러 줄을 한꺼번에 제거할 경우 콤보가 되어 보너스 매출이 올라갑니다.\n");
359     while (_kbhit()) _getch();
360     key = _getch();

361
362
363
364     system("cls");
365     gotoxy(x + 15, y + 1); printf(" _____ ");
366     textcolor(1);
```

Code Description

reset



The screenshot shows a code editor window with the file 'final.c' open. The 'reset' function is highlighted with a dashed blue rectangle. The code is written in C and includes comments in Korean explaining its purpose.

```
final.c
397 // 초기화
398
399 void reset(void) {
400
401     FILE* file = fopen("score.dat", "rt"); // score.dat 파일을 연결
402     if (file == 0) { best_score = 0; } // 파일이 없으면 강 최고점수에 0을 넣음
403     else {
404         fscanf(file, "%d", &best_score); // 파일이 열리면 최고점수를 불러옴
405         fclose(file); // 파일 닫음
406     }
407
408     level = 1; // 각종 변수 초기화
409     level_goal = 1000;
410     key = 0;
411     crush_on = 0;
412     cnt = 0;
413     speed = 100;
414     score = 0;
415     pay = 0;
416     fruit1 = 0;
417     fruit2 = 0;
418     fruit3 = 0;
419     fruit4 = 0;
420     system("cls"); // 화면지움
421     reset_main(); // main_org를 초기화
422     draw_map(); // 게임화면을 그림
423     draw_main(); // 게임판을 그림
424     b_type_next = rand() % 7; // 다음번에 나올 블록 종류를 랜덤하게 생성
425     reset_new_block(); // 리셋시 새로운 블록을 하나 만듦
426 }
427
428 void reset_main(void) { // 게임판을 초기화
429     int i, j;
430     score = 0;
431     pay = 0;
432     fruit1 = 0;
```

Code Description

draw_map

The screenshot shows a code editor window with the file 'final.c' open. The code is a C program, specifically the 'draw_map' function, which is responsible for displaying game status information on the screen. The code uses the 'gotoxy' function to print text at specific coordinates. The text includes the level number, next four pieces, current score, pay amount, and various control keys. It also includes a section for level 1 with additional symbols. The code ends with a call to 'buy_fruit()'.

```
final.c
472
473     void draw_map(void) { //게임 상태 표시를 나타내는 함수
474         int y = 3;           // level, goal, score만 게임중에 값이 바뀔수 도 있음 그 y값을 따로 저장해둠
475         // 그래서 혹시 게임 상태 표시 위치가 바뀌어도 그 함수에서 안바꿔도 되게..
476         gotoxy(STATUS_X_ADJ, y); printf("-----");
477         gotoxy(STATUS_X_ADJ, STATUS_Y_LEVEL = y + 2); printf("    LEVEL %5d", level);
478         gotoxy(STATUS_X_ADJ, y + 4); printf("-----");
479         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 2); printf("  ↗ N E X T ↘ ");
480         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 3); printf("      ");
481         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 4); printf("      ");
482         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 5); printf("      ");
483         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 6); printf("      ");
484         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 7); printf(" ↖ ↙ ");
485         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_SCORE = y + 10); printf("[ 오늘 매출 ] %6d", score);
486         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 12); printf("[ 매출 원가 ] %6d", pay); //점수 표시
487         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT = y + 14); printf("[ 구매한 포도 수 ] %6d", fruit1); //포도 가격 100원
488         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 2); printf("[ 구매한 키위 수 ] %6d", fruit2); // 키위 가격 200원
489         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 4); printf("[ 구매한 메론 수 ] %6d", fruit3); //메론 가격 500원
490         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 6); printf("[ 구매한 딸기 수 ] %6d", fruit4); // 딸기 가격 300원
491         gotoxy(STATUS_X_ADJ, y + 9); printf("-----");
492         gotoxy(STATUS_X_ADJ, y + 11); printf(" △ : Rotate");
493         gotoxy(STATUS_X_ADJ, y + 12); printf(" ◄ ► : Left / Right");
494         gotoxy(STATUS_X_ADJ, y + 13); printf(" ▽ : Soft Drop");
495         gotoxy(STATUS_X_ADJ, y + 15); printf(" SPACE : Hard Drop");
496         gotoxy(STATUS_X_ADJ, y + 17); printf(" P : Pause");
497         gotoxy(STATUS_X_ADJ, y + 19); printf(" ESC : Quit");
498         gotoxy(STATUS_X_ADJ, y + 21); printf("-----");
499     if (level = 1) {
500         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ + 10, y + 3); printf(" ( ^ ) ");
501         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ + 10, y + 4); printf(" ↖ ( ( > ) ");
502         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ + 10, y + 5); printf(" / ↗ ");
503     }
504
505
506
507
buy_fruit();
```

Code Description

draw_main

The screenshot shows a code editor window with the file 'Final.c' open. The code is part of a Tetris game implementation. The function 'draw_main' is highlighted. The code uses nested loops to iterate through the game board. It first iterates over columns (j) from 1 to MAIN_X - 1, checking if the current cell is empty. If it is, it sets it to 'CEILLING'. Then, it iterates over rows (i) from 0 to MAIN_Y, and for each row, it iterates over columns (j) from 0 to MAIN_X. It compares the value at main_cpy[i][j] with main_org[i][j]. If they are different, it prints the character corresponding to the new value. The code includes comments in Korean explaining the logic for clearing the board and handling changes in the game state.

```
509
510 void draw_main(void) { //게임판 그리는 함수
511     int i, j;
512
513     for (j = 1; j < MAIN_X - 1; j++) { //천장은 계속 새로운 블럭이 지나가서 지워지면 새로 그려줌
514         if (main_org[3][j] == EMPTY) main_org[3][j] = CEILLING;
515     }
516     for (i = 0; i < MAIN_Y; i++) {
517         for (j = 0; j < MAIN_X; j++) {
518             if (main_cpy[i][j] != main_org[i][j]) { //cpy랑 비교해서 값이 달라진 부분만 새로 그려줌.
519                 //이게 없으면 게임판전체를 계속 그려서 느려지고 반짝거림
520                 gotoxy(MAIN_X_ADJ + j, MAIN_Y_ADJ + i);
521                 switch (main_org[i][j]) {
522                     case EMPTY: //빈칸모양
523                         printf(" ");
524                         break;
525                     case CEILLING: //천장모양
526                         printf(".");
527                         break;
528                     case WALL: //벽모양
529                         printf("■");
530                         break;
531                     case INACTIVE_BLOCK: //굳은 블럭 모양
532                         printf("□");
533                         break;
534                     case ACTIVE_BLOCK: //움직이고있는 블럭 모양
535                         textcolor(randcolor);
536                         printf("■");
```

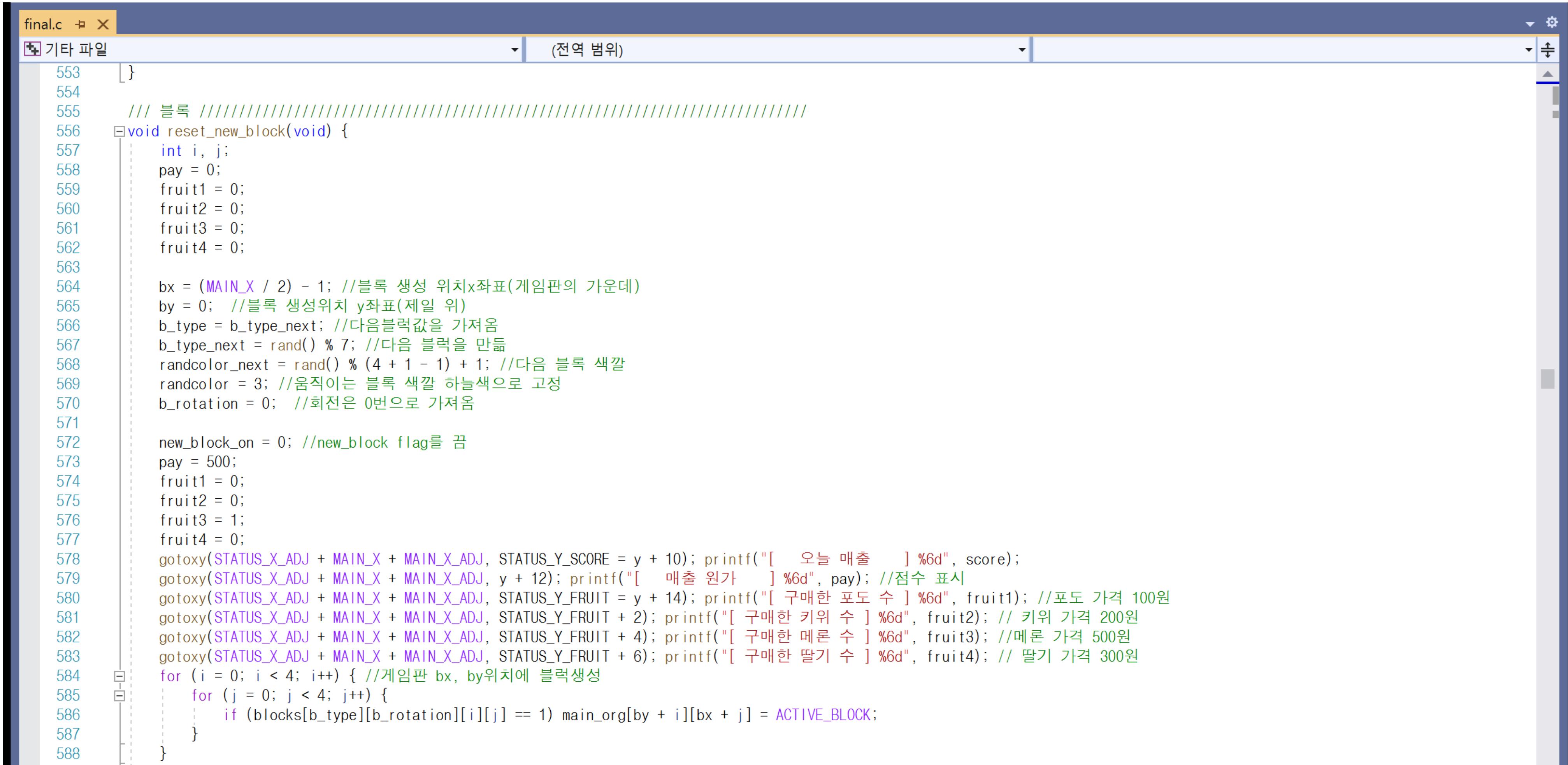
Code Description

`new_block`

```
Final.c  ✘ X
Tetris_final (전역 범위)
605     }
606
607     void new_block(void) { //새로운 블록 생성
608         int i, j;
609
610         bx = (MAIN_X / 2) - 1; //블록 생성 위치x좌표(게임판의 가운데)
611         by = 0; //블록 생성위치 y좌표(제일 위)
612         b_type = b_type_next; //다음블럭값을 가져옴
613         b_type_next = rand() % 7; //다음 블럭을 만듦
614         inactive_color = randcolor; //굳은 블록 색깔
615         randcolor = randcolor_next; //움직이는 블록 색깔
616         randcolor_next = rand() % (4 + 1 - 1) + 1; //다음 블록 색깔
617         b_rotation = 0; //회전은 0번으로 가져옴
618
619         new_block_on = 0; //new_block flag를 끔
620         buy_fruit();
621         for (i = 0; i < 4; i++) { //게임판 bx, by위치에 블럭생성
622             for (j = 0; j < 4; j++) {
623                 if (blocks[b_type][b_rotation][i][j] == 1) main_org[by + i][bx + j] = ACTIVE_BLOCK;
624             }
625         }
626
627         for (i = 1; i < 3; i++) { //게임상태표시에 다음에 나올블럭을 그림
628             for (j = 0; j < 4; j++) {
629                 if (blocks[b_type_next][0][i][j] == 1) {
630                     gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ + 2 + j, i + 6);
631                     textcolor(randcolor_next);
632                 }
633             }
634         }
635     }
636 }
```

Code Description

reset_new_block

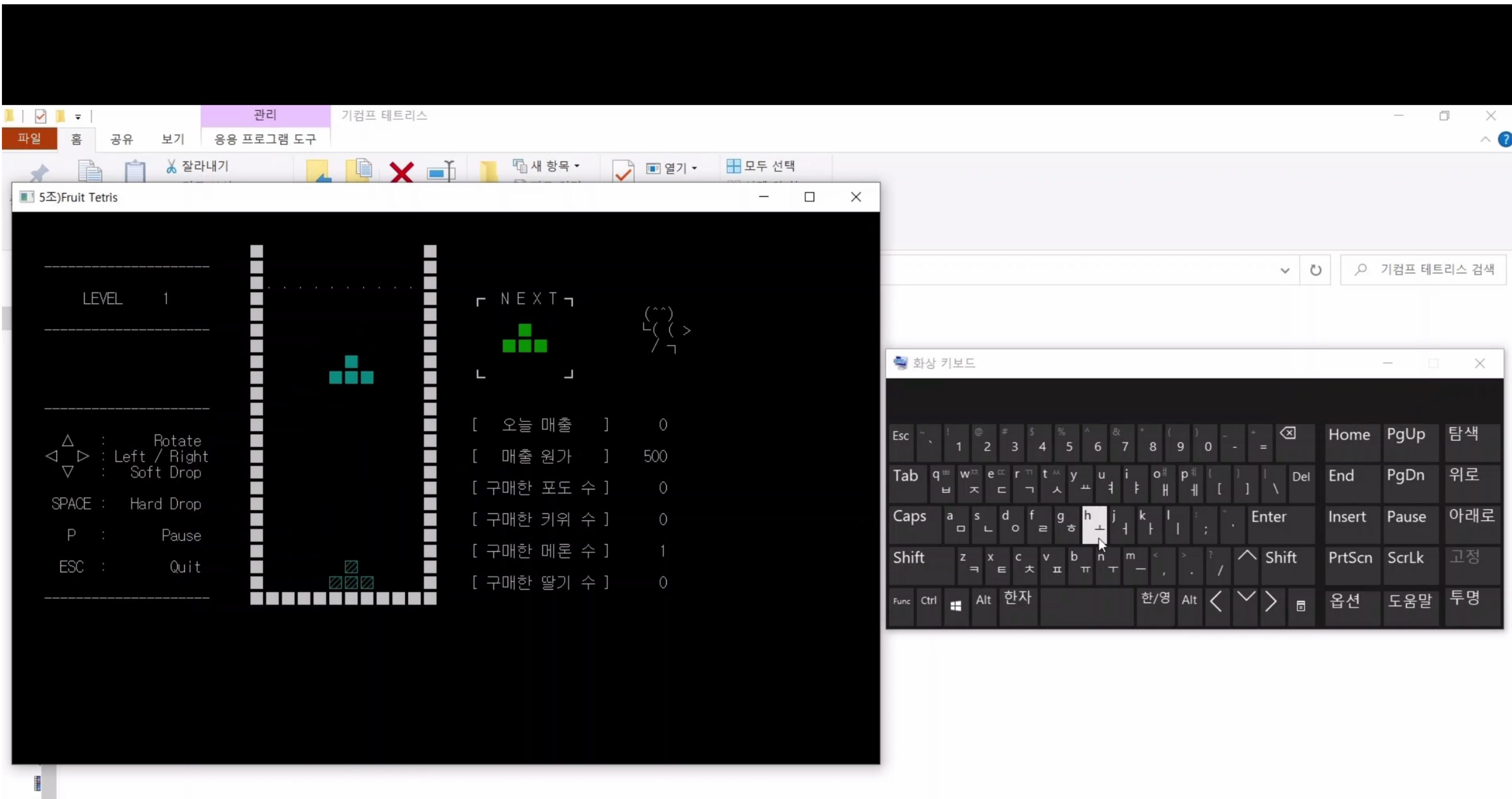


```
final.c  X
기타 파일  (전역 범위)

553     []
554
555     /// 블록 /////////////////////////////////
556     void reset_new_block(void) {
557         int i, j;
558         pay = 0;
559         fruit1 = 0;
560         fruit2 = 0;
561         fruit3 = 0;
562         fruit4 = 0;
563
564         bx = (MAIN_X / 2) - 1; //블록 생성 위치x좌표(게임판의 가운데)
565         by = 0; //블록 생성위치 y좌표(제일 위)
566         b_type = b_type_next; //다음블럭값을 가져옴
567         b_type_next = rand() % 7; //다음 블럭을 만듦
568         randcolor_next = rand() % (4 + 1 - 1) + 1; //다음 블록 색깔
569         randcolor = 3; //움직이는 블록 색깔 하늘색으로 고정
570         b_rotation = 0; //회전은 0번으로 가져옴
571
572         new_block_on = 0; //new_block flag를 끔
573         pay = 500;
574         fruit1 = 0;
575         fruit2 = 0;
576         fruit3 = 1;
577         fruit4 = 0;
578         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_SCORE = y + 10); printf("[ 오늘 매출 ] %6d", score);
579         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 12); printf("[ 매출 원가 ] %6d", pay); //점수 표시
580         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT = y + 14); printf("[ 구매한 포도 수 ] %6d", fruit1); //포도 가격 100원
581         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 2); printf("[ 구매한 키위 수 ] %6d", fruit2); // 키위 가격 200원
582         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 4); printf("[ 구매한 메론 수 ] %6d", fruit3); //메론 가격 500원
583         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 6); printf("[ 구매한 딸기 수 ] %6d", fruit4); // 딸기 가격 300원
584         for (i = 0; i < 4; i++) { //게임판 bx, by위치에 블럭생성
585             for (j = 0; j < 4; j++) {
586                 if (blocks[b_type][b_rotation][i][j] == 1) main_org[by + i][bx + j] = ACTIVE_BLOCK;
587             }
588         }
}
```

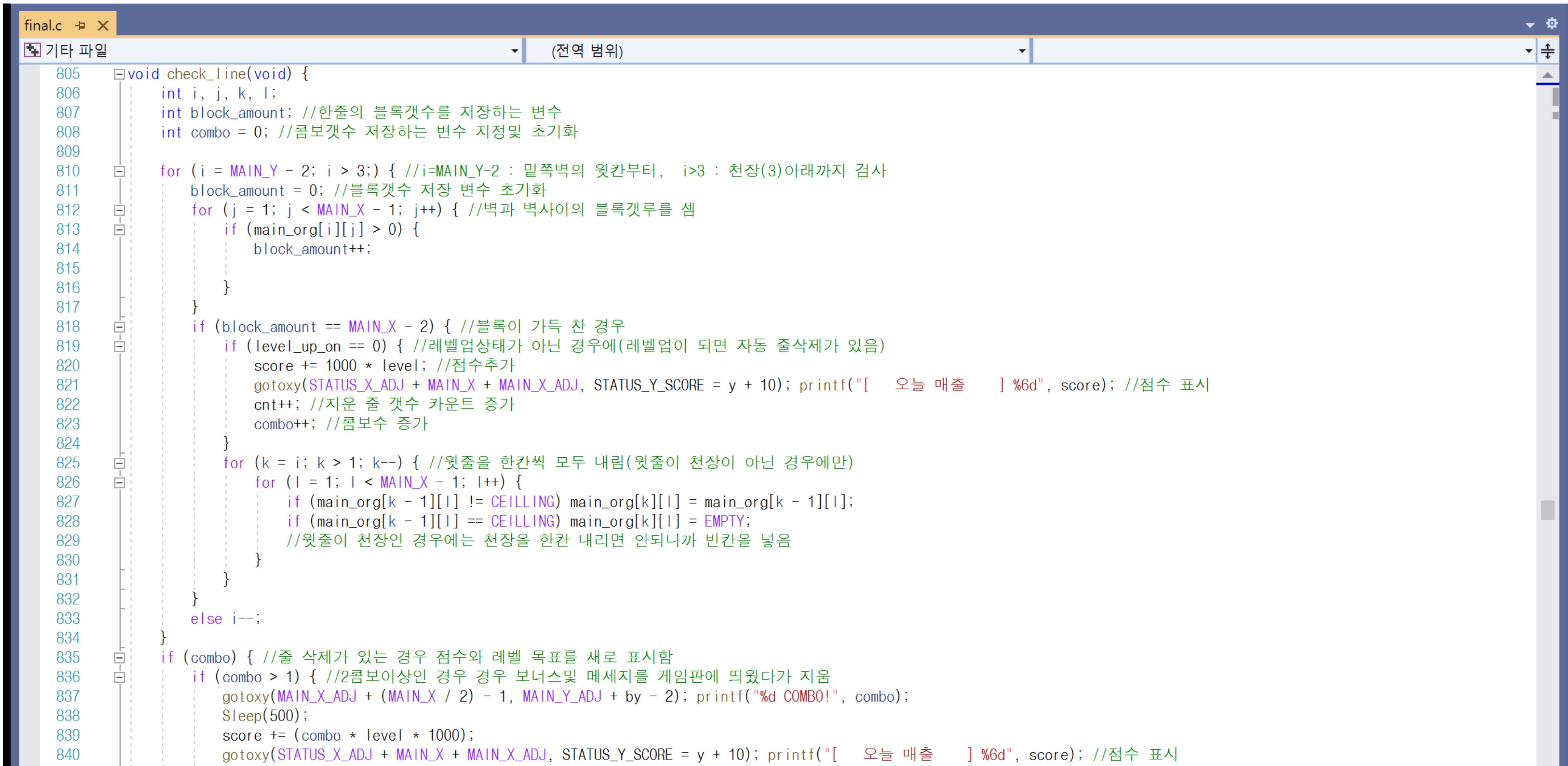
Code Description

reset_new_block



Code Description

check_line



```
final.c  X
파일 목록 (전역 범위)
805 void check_line(void) {
806     int i, j, k, l;
807     int block_amount; //한줄의 블록갯수를 저장하는 변수
808     int combo = 0; //콤보갯수 저장하는 변수 지정및 초기화
809
810     for (i = MAIN_Y - 2; i > 3;) { //i=MAIN_Y-2 : 밑쪽벽의 윗칸부터, i>3 : 천장(3)아래까지 검사
811         block_amount = 0; //블록갯수 저장 변수 초기화
812         for (j = 1; j < MAIN_X - 1; j++) { //벽과 벽사이의 블록갯수를 셈
813             if (main_org[i][j] > 0) {
814                 block_amount++;
815             }
816         }
817         if (block_amount == MAIN_X - 2) { //블록이 가득 찬 경우
818             if (level_up_on == 0) { //레벨업상태가 아닌 경우에(레벨업이 되면 자동 줄삭제가 있음)
819                 score += 1000 * level; //점수추가
820                 gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_SCORE = y + 10); printf("[ 오늘 매출 ] %6d", score); //점수 표시
821                 cnt++; //지운 줄 갯수 카운트 증가
822                 combo++; //콤보수 증가
823             }
824             for (k = i; k > 1; k--) { //윗줄을 한칸씩 모두 내림(윗줄이 천장이 아닌 경우에만)
825                 for (l = 1; l < MAIN_X - 1; l++) {
826                     if (main_org[k - 1][l] != CEILING) main_org[k][l] = main_org[k - 1][l];
827                     if (main_org[k - 1][l] == CEILING) main_org[k][l] = EMPTY;
828                     //윗줄이 천장인 경우에는 천장을 한칸 내리면 안되니까 빈칸을 넣음
829                 }
830             }
831         }
832     }
833     else i--;
834 }
835 if (combo) { //줄 삭제가 있는 경우 점수와 레벨 목표를 새로 표시함
836     if (combo > 1) { //2콤보이상인 경우 경우 보너스및 메세지를 게임판에 띄웠다가 지움
837         gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 1, MAIN_Y_ADJ + by - 2); printf("%d COMBO!", combo);
838         Sleep(500);
839         score += (combo * level * 1000);
840         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_SCORE = y + 10); printf("[ 오늘 매출 ] %6d", score); //점수 표시
841 }
```

Code Description

check_level_up

```
Final.c  ✘ (전역 범위)
Tetris_final  ✘ (전역 범위)

void check_level_up(void) {
    int i, j;
    if (cnt >= 10 && level <= 4) { //레벨별로 10줄씩 없애야함. 10줄이상 없앤 경우
        draw_main();
        level_up_on = 1; //레벨업 flag를 띄움
        level += 1; //레벨을 1 올림
        cnt = 0; //자운 줄수 초기화

        for (i = 0; i < 4; i++) {
            gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 3, MAIN_Y_ADJ + 4);
            printf("      ");
            gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 3, MAIN_Y_ADJ + 6);
            printf("      ");
            Sleep(200);

            gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 3, MAIN_Y_ADJ + 4);
            printf("☆LEVEL UP!☆");
            gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 3, MAIN_Y_ADJ + 6);
            printf("☆SPEED UP!☆");
            Sleep(200);
        }
        reset_main_cpy(); //텍스트를 지우기 위해 main_cpy를 초기화.
        // (main_cpy와 main_org가 전부 다르므로 다음번 draw() 호출시 게임판 전체를 새로 그리게 됨)

        for (i = MAIN_Y - 2; i > MAIN_Y - 2 - (level - 1); i--) { //레벨업보상으로 각 레벨-1의 수만큼 아래쪽 줄을 지워줌
            for (j = 1; j < MAIN_X - 1; j++) {
                main_org[i][j] = INACTIVE_BLOCK; // 줄을 블록으로 모두 채우고
                gotoxy(MAIN_X_ADJ + j, MAIN_Y_ADJ + i); // 별을 찍어줌.. 이뻐보이게
                printf("★");
                Sleep(20);
            }
            Sleep(100); //별찍은거 보여주기 위해 delay
            check_line(); //블록으로 모두 채운것 지우기
        }
        // .check_line() 함수 내부에서 level up flag가 켜져있는 경우 점수는 없음.
    }
}

Final.c  ✘ (전역 범위)
Tetris_final  ✘ (전역 범위)

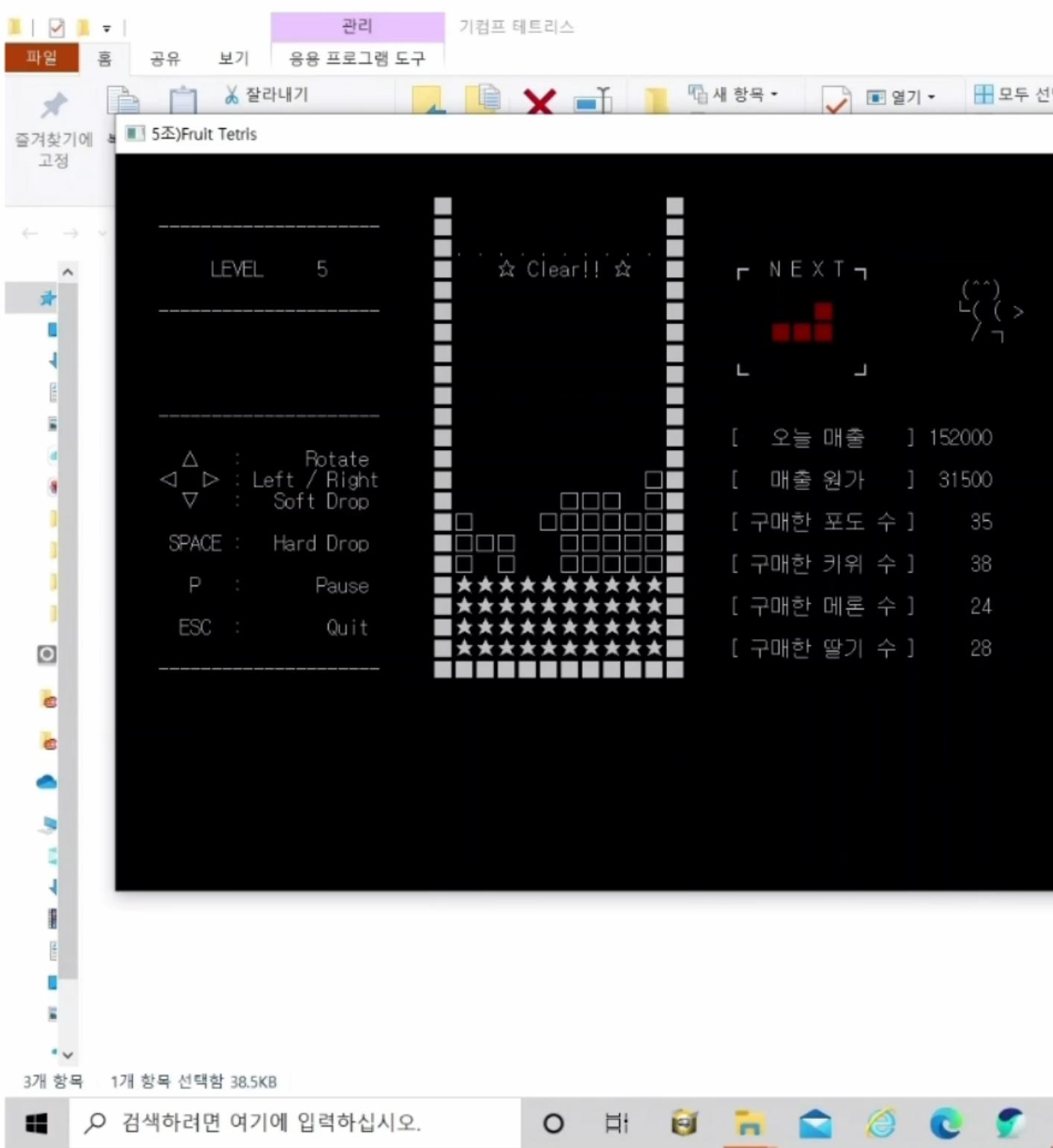
899     gotoxy(STATUS_X_ADJ, STATUS_Y_LEVEL); printf("      LEVEL %5d", level); //레벨표시
900     gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_SCORE = y + 10); printf("[      ");
901
902     //gotoxy(STATUS_X_ADJ, STATUS_Y_GOAL); printf(" GOAL : %5d", 10 - cnt); // 레벨목표
903
904 }
905 if (level > 4) {
906     for (i = 0; i < 4; i++) {
907         gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 3, MAIN_Y_ADJ + 4);
908         printf("      ");
909         gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 3, MAIN_Y_ADJ + 6);
910         printf("      ");
911         Sleep(400);

912         gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 3, MAIN_Y_ADJ + 4);
913         printf("☆ Clear!! ☆");
914         gotoxy(MAIN_X_ADJ + (MAIN_X / 2) - 3, MAIN_Y_ADJ + 6);
915         printf("      ");
916         Sleep(400);
917     }
918
919     reset_main_cpy(); //텍스트를 지우기 위해 main_cpy를 초기화.
920     // (main_cpy와 main_org가 전부 다르므로 다음번 draw() 호출시 게임판 전체를 새로 그리게 됨)
921     for (int j = 1; j < MAIN_Y - 1; j++) {
922     {
923         gotoxy(MAIN_X_ADJ + 1, j + MAIN_Y_ADJ); printf("★★★★★★★★★★★");
924         Sleep(30);
925     }
926 }
```

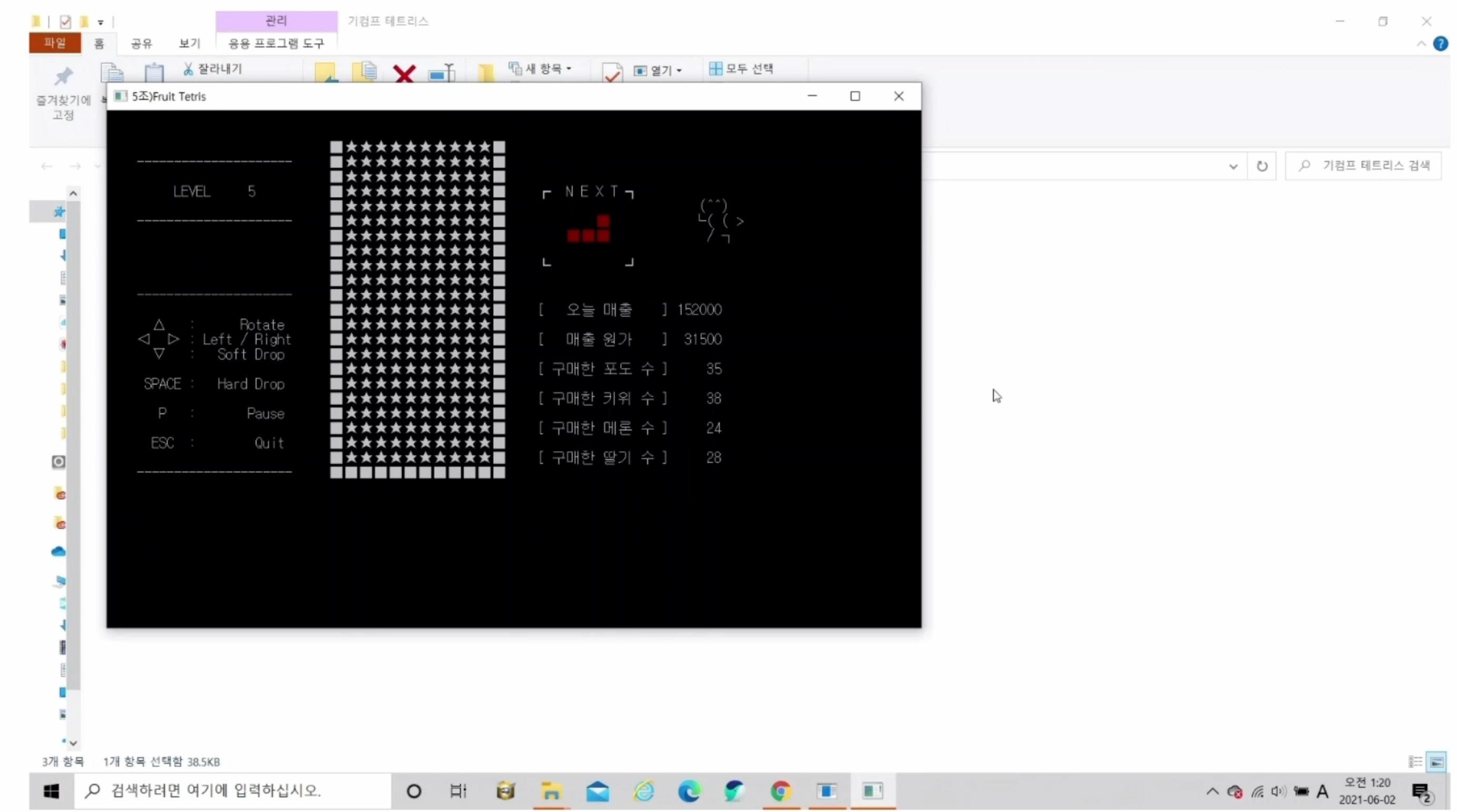
Code Description

check_level_up

Demonstration

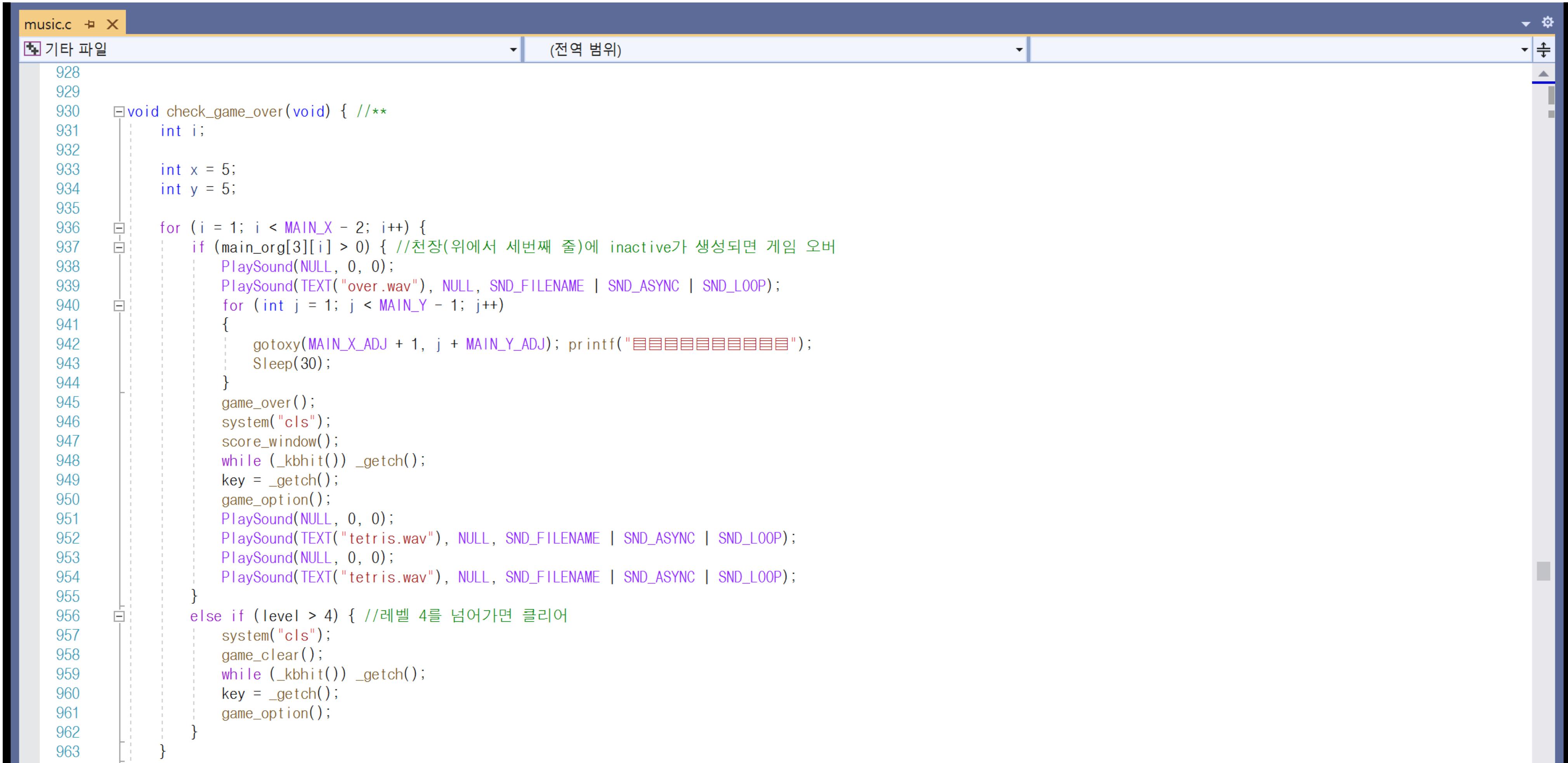


Demonstration



Code Description

check_game_over

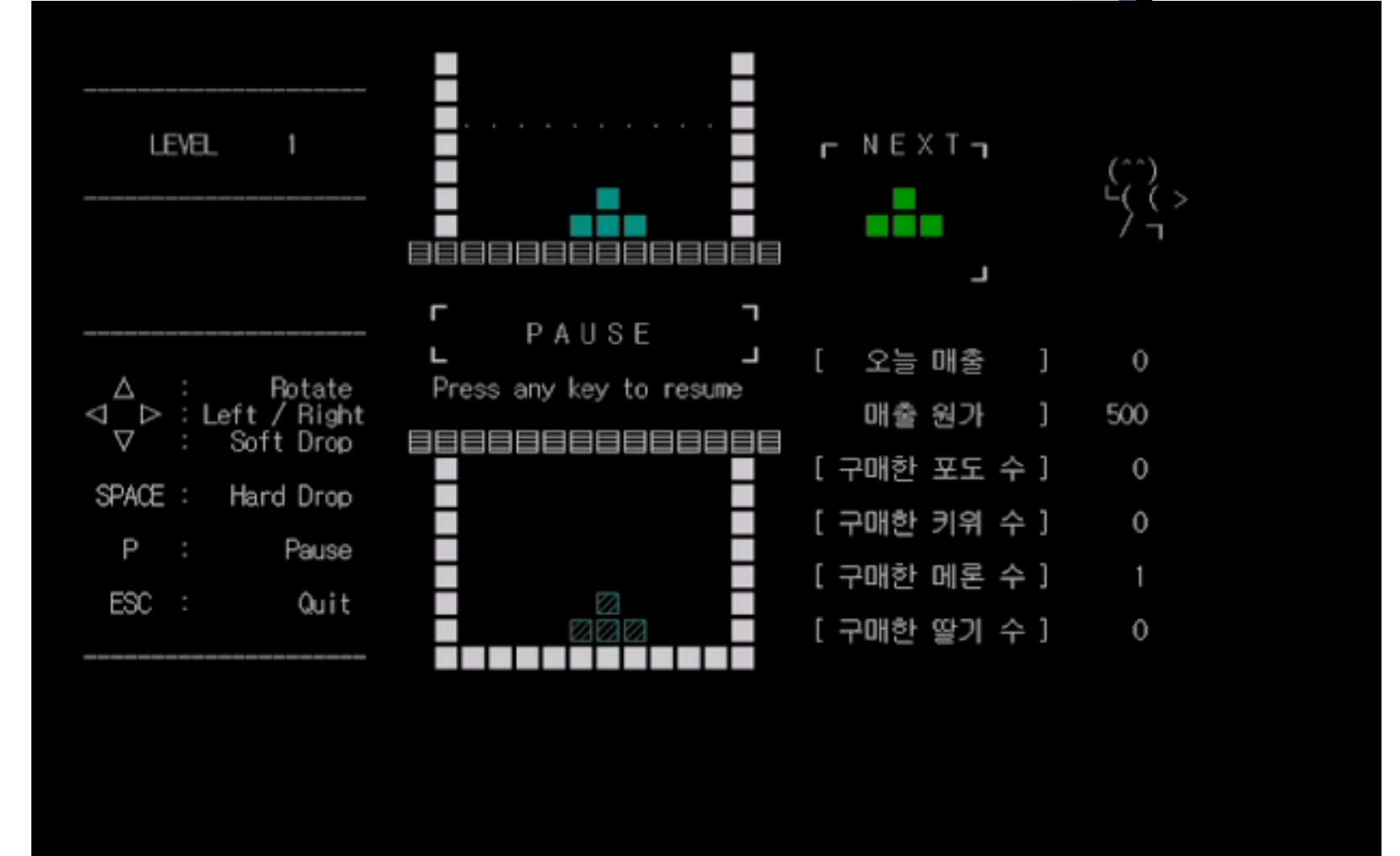


```
music.c  X
▣ 기타 파일  (전역 범위)

928
929
930 void check_game_over(void) { /**
931     int i;
932
933     int x = 5;
934     int y = 5;
935
936     for (i = 1; i < MAIN_X - 2; i++) {
937         if (main_org[3][i] > 0) { //천장(위에서 세번째 줄)에 inactive가 생성되면 게임 오버
938             PlaySound(NULL, 0, 0);
939             PlaySound(TEXT("over.wav"), NULL, SND_FILENAME | SND_ASYNC | SND_LOOP);
940             for (int j = 1; j < MAIN_Y - 1; j++) {
941                 gotoxy(MAIN_X_ADJ + 1, j + MAIN_Y_ADJ); printf("::::::::::::::::::");
942                 Sleep(30);
943             }
944             game_over();
945             system("cls");
946             score_window();
947             while (_kbhit()) _getch();
948             key = _getch();
949             game_option();
950             PlaySound(NULL, 0, 0);
951             PlaySound(TEXT("tetris.wav"), NULL, SND_FILENAME | SND_ASYNC | SND_LOOP);
952             PlaySound(NULL, 0, 0);
953             PlaySound(TEXT("tetris.wav"), NULL, SND_FILENAME | SND_ASYNC | SND_LOOP);
954         }
955     }
956     else if (level > 4) { //레벨 4를 넘어가면 클리어
957         system("cls");
958         game_clear();
959         while (_kbhit()) _getch();
960         key = _getch();
961         game_option();
962     }
963 }
```

Code Description

pause



The image shows a screenshot of a Tetris game. At the top, there is a status bar with "LEVEL 1" and a score of "0". Below the status bar, the game board displays a 10x20 grid of blocks. A green L-shaped tetromino is currently falling. On the right side of the screen, there is a control panel with the following information:

NEXT	(^)
PAUSE	[오늘 매출] 0
< △ > : Left / Right	[매출 원가] 500
▽ : Soft Drop	[구매한 포도 수] 0
SPACE : Hard Drop	[구매한 키위 수] 0
P : Pause	[구매한 메론 수] 1
ESC : Quit	[구매한 딸기 수] 0

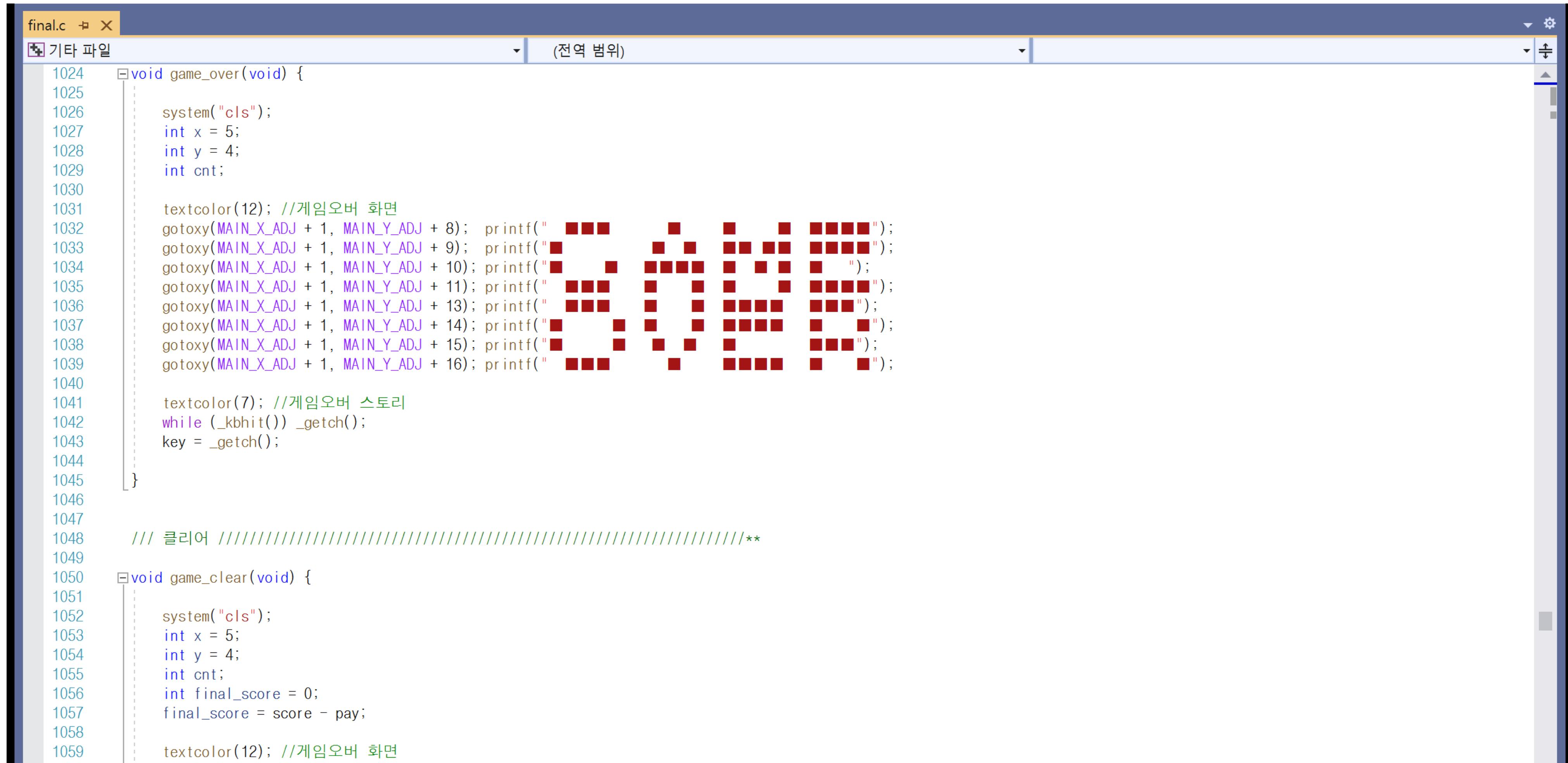
Below the control panel, there is a message: "Press any key to resume".

Code Snippet (final.c):

```
964     }
965
966     void pause(void) { //게임 일시정지 함수
967         int i, j;
968
969         int x = 14;
970         int y = 9;
971         PlaySound(NULL, 0, 0);
972         for (i = 1; i < MAIN_X - 2; i++) { //게임 일시정지 메세지
973             gotoxy(x, y + 0); printf("■■■■■■■■■■■■■■■■■■");
974             gotoxy(x, y + 1); printf("      ");
975             gotoxy(x, y + 2); printf("  P A U S E  ");
976             gotoxy(x, y + 3); printf("      P A U S E      ");
977             gotoxy(x, y + 4); printf("  L      J      ");
978             gotoxy(x, y + 5); printf(" Press any key to resume      ");
979             gotoxy(x, y + 6); printf("      ");
980             gotoxy(x, y + 7); printf("■■■■■■■■■■■■■■■■■■");
981         }
982         _getch(); //키입력까지 대기
983
984         system("cls"); //화면 지우고 새로 그림
985         //PlaySound(TEXT("tetris.wav"), NULL, SND_FILENAME | SND_ASYNC | SND_LOOP);
986         reset_main_cpy();
987         if (randcolor == 1) {
988             fruit1--;
989             pay -= 100;
990         }
991         else if (randcolor == 2) {
992             fruit2--;
993             pay -= 200;
994         }
995         else if (randcolor == 3) {
996             fruit3--;
997             pay -= 500;
998         }
999     }
```

Code Description

game_over



The screenshot shows a code editor window with the file "final.c" open. The code is written in C and contains two functions: `game_over` and `game_clear`.

```
final.c  X
기타 파일  (전역 범위)

1024 void game_over(void) {
1025     system("cls");
1026     int x = 5;
1027     int y = 4;
1028     int cnt;
1029
1030     textcolor(12); //게임오버 화면
1031     gotoxy(MAIN_X_ADJ + 1, MAIN_Y_ADJ + 8); printf(" ■■■ ");
1032     gotoxy(MAIN_X_ADJ + 1, MAIN_Y_ADJ + 9); printf(" ■■■ ");
1033     gotoxy(MAIN_X_ADJ + 1, MAIN_Y_ADJ + 10); printf(" ■■■ ");
1034     gotoxy(MAIN_X_ADJ + 1, MAIN_Y_ADJ + 11); printf(" ■■■ ");
1035     gotoxy(MAIN_X_ADJ + 1, MAIN_Y_ADJ + 13); printf(" ■■■ ");
1036     gotoxy(MAIN_X_ADJ + 1, MAIN_Y_ADJ + 14); printf(" ■■■ ");
1037     gotoxy(MAIN_X_ADJ + 1, MAIN_Y_ADJ + 15); printf(" ■■■ ");
1038     gotoxy(MAIN_X_ADJ + 1, MAIN_Y_ADJ + 16); printf(" ■■■ ");
1039
1040     textcolor(7); //게임오버 스토리
1041     while (_kbhit()) _getch();
1042     key = _getch();
1043
1044 }
1045
1046
1047
1048 // 클리어 //////////////////////////////**/
1049
1050 void game_clear(void) {
1051     system("cls");
1052     int x = 5;
1053     int y = 4;
1054     int cnt;
1055     int final_score = 0;
1056     final_score = score - pay;
1057
1058     textcolor(12); //게임오버 화면
1059 }
```

The `game_over` function prints a series of red squares (■) on the screen at coordinates (MAIN_X_ADJ + 1, MAIN_Y_ADJ + 8) through (MAIN_X_ADJ + 1, MAIN_Y_ADJ + 16). It then waits for a key press using `_kbhit()` and `_getch()`. The `game_clear` function clears the screen using `system("cls")` and initializes variables for a new game.

Code Description

game_clear

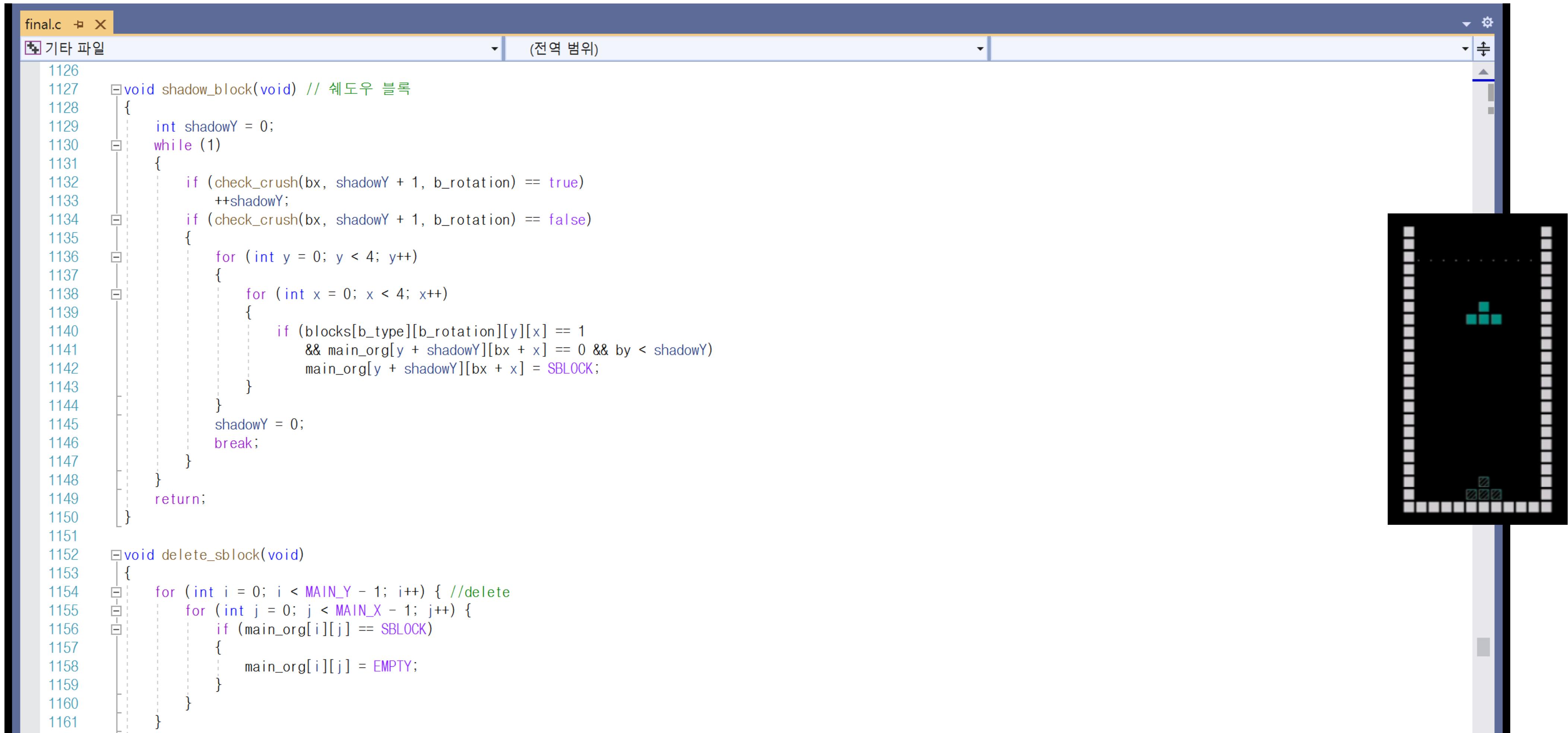


```
final.c  X
기타 파일 (전역 범위)

1051
1052     system("cls");
1053     int x = 5;
1054     int y = 4;
1055     int cnt;
1056     int final_score = 0;
1057     final_score = score - pay;
1058
1059     textcolor(12); //게임오버 화면
1060     gotoxy(x , y + 1); printf(" ■ ■ ■ ■ ■ ");
1061     gotoxy(x , y + 2); printf(" ■ ■ ■ ■ ■ ");
1062     gotoxy(x , y + 3); printf(" ■ ■ ■ ■ ■ ");
1063     gotoxy(x , y + 4); printf(" ■ ■ ■ ■ ■ ");
1064     gotoxy(x , y + 5); printf(" ■ ■ ■ ■ ■ ");
1065
1066     textcolor(11); //건물
1067     for ( int j = 20; j >= 5; j--)
1068     {
1069         gotoxy(x + 25, y + j); printf(" | ■ ■ ■ ■ ■ | ");
1070         Sleep(50);
1071     }
1072     textcolor(12); gotoxy(x + 30, y + 2); printf("▶"); textcolor(11);
1073     gotoxy(x + 30, y + 3); printf(" | ");
1074     gotoxy(x + 25, y + 4); printf(" _____ ");
1075     gotoxy(x + 26, y + 6); printf("    중앙백화점    ");
1076     textcolor(7);
1077     gotoxy(x + 2, y + 8); printf(" _____ ");
1078     gotoxy(x + 2, y + 10); printf(" [ 오늘 번 돈 ] %6d원 ", score);
1079     gotoxy(x + 2, y + 11); printf(" [ 매출 원가 ] %6d원 ", pay);
1080     gotoxy(x + 2, y + 12); printf(" [ 순수익 ] %6d원 ", final_score);
1081     gotoxy(x + 2, y + 14); printf(" _____ ");
1082
1083
1084     for (cnt = 0;; cnt++) //cnt를 1씩 증가시키면서 계속 반복
1085     {
1086         if (_kbhit()) break; //키입력이 있으면 무한루프 종료
```

Code Description

shadow_block

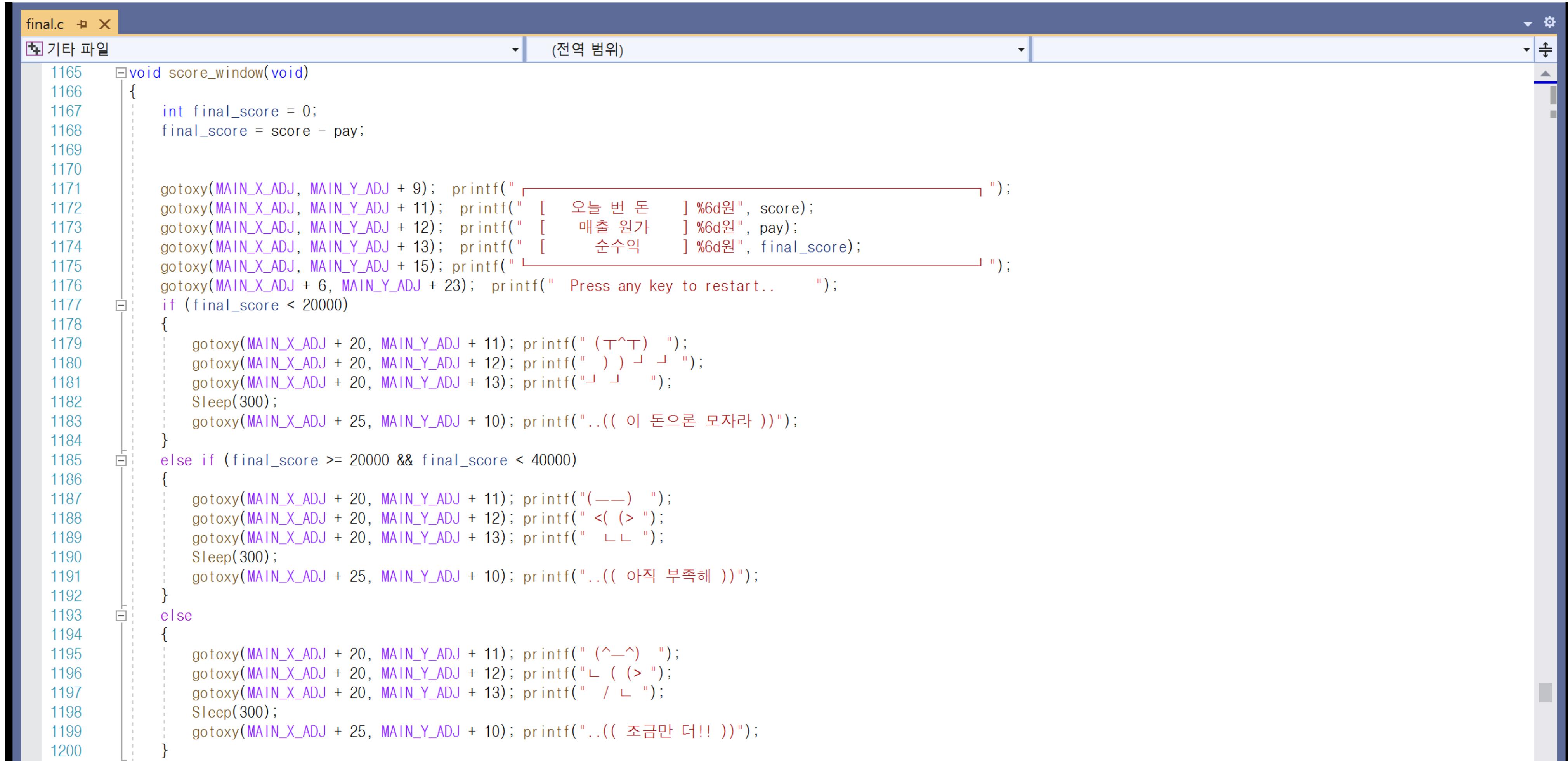


The image shows a code editor window with the file 'final.c' open. The 'shadow_block' function is highlighted with syntax coloring. To the right of the editor is a 10x10 grid visualization of a game board state, showing a blue L-shaped tetromino falling towards a wall of grey blocks.

```
final.c
1126 void shadow_block(void) // 쉐도우 블록
1127 {
1128     int shadowY = 0;
1129     while (1)
1130     {
1131         if (check_crush(bx, shadowY + 1, b_rotation) == true)
1132             ++shadowY;
1133         if (check_crush(bx, shadowY + 1, b_rotation) == false)
1134         {
1135             for (int y = 0; y < 4; y++)
1136             {
1137                 for (int x = 0; x < 4; x++)
1138                 {
1139                     if (blocks[b_type][b_rotation][y][x] == 1
1140                         && main_org[y + shadowY][bx + x] == 0 && by < shadowY)
1141                         main_org[y + shadowY][bx + x] = SBLOCK;
1142                 }
1143             }
1144             shadowY = 0;
1145             break;
1146         }
1147     }
1148     return;
1149 }
1150
1151
1152 void delete_sbblock(void)
1153 {
1154     for (int i = 0; i < MAIN_Y - 1; i++) { //delete
1155         for (int j = 0; j < MAIN_X - 1; j++) {
1156             if (main_org[i][j] == SBLOCK)
1157             {
1158                 main_org[i][j] = EMPTY;
1159             }
1160         }
1161     }
}
```

Code Description

score_window



The screenshot shows a code editor window with the file "final.c" open. The code is a C function named "score_window". The function calculates a final score by subtracting pay from score. It then prints various messages and ASCII art based on the final score. The code uses gotoxy() for output and includes Sleep(300) calls between some output operations.

```
final.c  X
기타 파일 (전역 범위)

1165 void score_window(void)
1166 {
1167     int final_score = 0;
1168     final_score = score - pay;
1169
1170
1171     gotoxy(MAIN_X_ADJ, MAIN_Y_ADJ + 9); printf(" _____");
1172     gotoxy(MAIN_X_ADJ, MAIN_Y_ADJ + 11); printf(" [ 오늘 번 돈 ] %6d원", score);
1173     gotoxy(MAIN_X_ADJ, MAIN_Y_ADJ + 12); printf(" [ 매출 원가 ] %6d원", pay);
1174     gotoxy(MAIN_X_ADJ, MAIN_Y_ADJ + 13); printf(" [ 순수익 ] %6d원", final_score);
1175     gotoxy(MAIN_X_ADJ, MAIN_Y_ADJ + 15); printf(" _____");
1176     gotoxy(MAIN_X_ADJ + 6, MAIN_Y_ADJ + 23); printf(" Press any key to restart.. ");
1177     if (final_score < 20000)
1178     {
1179         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 11); printf(" (T^T) ");
1180         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 12); printf(" ) ) ㄴ ㄴ ");
1181         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 13); printf(" ㄴ ㄴ ");
1182         Sleep(300);
1183         gotoxy(MAIN_X_ADJ + 25, MAIN_Y_ADJ + 10); printf("..(( 이 돈으론 모자라 ))");
1184     }
1185     else if (final_score >= 20000 && final_score < 40000)
1186     {
1187         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 11); printf("(-_-) ");
1188         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 12); printf(" < ( > );
1189         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 13); printf(" ㄴ ㄴ ");
1190         Sleep(300);
1191         gotoxy(MAIN_X_ADJ + 25, MAIN_Y_ADJ + 10); printf("..(( 아직 부족해 ))");
1192     }
1193     else
1194     {
1195         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 11); printf(" (^_^) ");
1196         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 12); printf(" ㄴ ( (> ";
1197         gotoxy(MAIN_X_ADJ + 20, MAIN_Y_ADJ + 13); printf(" / ㄴ ");
1198         Sleep(300);
1199         gotoxy(MAIN_X_ADJ + 25, MAIN_Y_ADJ + 10); printf("..(( 조금만 더!! ))");
1200     }
}
```

Code Description

buy_fruit

The screenshot shows a code editor window with the file 'final.c' open. The code is written in C and contains a function named 'buy_fruit'. The function takes no arguments and returns nothing. It uses a switch statement based on a variable 'randcolor' (which has values 1, 2, 3, or 4) to determine which type of fruit to buy and how much it costs. The code also includes a call to 'game_option' at the end.

```
final.c  X
기타 파일 (전역 범위)

1201 []
1202
1203 void buy_fruit(void)
1204 {
1205     if (randcolor == 1) {
1206         fruit1++;
1207         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT); printf("[ 구매한 포도 수 ] %6d", fruit1);
1208         pay += 100;
1209         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 12); printf("[ 매출 원가 ] %6d", pay);
1210     }
1211     else if (randcolor == 2) {
1212         fruit2++;
1213         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 2); printf("[ 구매한 키위 수 ] %6d", fruit2);
1214         pay += 200;
1215         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 12); printf("[ 매출 원가 ] %6d", pay);
1216     }
1217     else if (randcolor == 3) {
1218         fruit3++;
1219         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 4); printf("[ 구매한 메론 수 ] %6d", fruit3);
1220         pay += 500;
1221         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 12); printf("[ 매출 원가 ] %6d", pay);
1222     }
1223     else {
1224         fruit4++;
1225         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, STATUS_Y_FRUIT + 6); printf("[ 구매한 딸기 수 ] %6d", fruit4);
1226         pay += 300;
1227         gotoxy(STATUS_X_ADJ + MAIN_X + MAIN_X_ADJ, y + 12); printf("[ 매출 원가 ] %6d", pay);
1228     }
1229 }
1230
1231 void game_option(void) //***
1232 {
1233     system("cls");
1234     int x = 15;
1235     int y = 13;
```

On the right side of the editor, there is a sidebar displaying the following statistics:

[오늘 매출]	28000
[매출 원가]	18900
[구매한 포도 수]	16
[구매한 키위 수]	21
[구매한 메론 수]	16
[구매한 딸기 수]	17

Code Description

game_option

The screenshot shows a code editor window with the file 'final.c' open. The code implements a game option function. The function initializes variables, prints a menu, and then enters a loop to handle keyboard input for movement.

```
final.c  X
기타 파일  (전역 범위)

1231
1232     void game_option(void) //***
1233 {
1234     system("cls");
1235     int x = 15;
1236     int y = 13;
1237     int cnt = 1;
1238
1239     textcolor(7);
1240     gotoxy(x, y - 3); printf("■■■■■■■■■■■■■■■■");
1241
1242     gotoxy(x + 5, y); printf("> 재시작");
1243
1244     gotoxy(x + 6, y + 2); printf("종료");
1245
1246     gotoxy(x, y + 5); printf("■■■■■■■■■■■■■■■■");
1247
1248     while (cnt) //cnt를 1씩 증가시키면서 계속 반복
1249     {
1250         if (_kbhit()) { //키입력이 있는 경우
1251             key = _getch(); //키값을 받음
1252             if (key == 224) { //방향키인경우
1253                 do { key = _getch(); } while (key == 224); //방향키지시값을 버림
1254                 switch (key) {
1255                     case DOWN: //아래쪽 방향키 눌렀을때
1256                         if (y == 13) {
1257                             gotoxy(x + 5, y); printf(" ");
1258                             y = 15;
1259                             gotoxy(x + 5, y); printf(">");
1260                         }
1261                         break;
1262                     case UP: //위쪽 방향키 눌렀을때
1263                         if (y == 15) {
1264                             gotoxy(x + 5, y); printf(" ");
1265                             y = 13;
1266                             gotoxy(x + 5, y); printf(">");
```

The right side of the editor shows the rendered output of the printed text. It displays a menu with three items: a series of vertical bars, the text "> 재시작", another series of vertical bars, the text "종료", and finally another series of vertical bars.

Feedback

Fruit Tetris

- A phenomenon in which the shadow block is not represented in its normal position
- Lagging occurs due to reduced code efficiency
- A phenomenon in which the color of the first block is fixed to black (Fixed to light blue in our code)

Thank you

Fruit Tetris

Group 5